Data Min Knowl Disc (2011) 23:447–478
DOI 10.1007/s10618-010-0210-x

# Leveraging social media networks for classification

**Lei Tang · Huan Liu**

Received: 10 August 2009 / Accepted: 15 December 2010 / Published online: 14 January 2011


**Abstract** 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 Facebook, also brings about many data mining opportunities and novel challenges. In particular, we focus on classification tasks with user interaction information in a social network. Networks in social media are heterogeneous, consisting of various relations. Since the relation-type information may not be available in social media, most existing approaches treat these inhomogeneous connections homogeneously, leading to an unsatisfactory classification performance. In order to handle the network heterogeneity, we propose the concept of social dimension to represent actors' latent affiliations, and develop a classification framework based on that. The proposed framework, SocioDim, first extracts social dimensions based on the network structure to accurately capture prominent interaction patterns between actors, then learns a discriminative classifier to select relevant social dimensions. SocioDim, by differentiating different types of network connections, outperforms existing representative methods of classification in social media, and offers a simple yet effective approach to integrating two types of seemingly orthogonal information: the network of actors and their attributes.

Responsible editor: Johannes Gehrke.

L. Tang (✉)
Advertising Sciences, Yahoo! Labs, Santa Clara, CA 95054, USA
e-mail: ltang@yahoo-inc.com

H. Liu
Computer Science and Engineering, Arizona State University, Tempe, AZ 85287, USA
e-mail: Huan.Liu@asu.edu

**Keywords** Social media · Social network analysis · Relational learning · Within-network classification · Collective inference

## 1 Introduction

Social media, such as Facebook, MySpace, Twitter, BlogSpot, Digg, YouTube, and Flickr, has streamlined ways for people to express their thoughts, voice their opinions, and connect to each other anytime and anywhere. For instance, popular content-sharing sites like Del.icio.us, Flickr, and YouTube allow users to upload, tag and comment on different types of content (bookmarks, photos, or videos). Users registered at these sites can also become friends, fans, or followers of others. The prolific and expanded use of social media has turned online interactions into a vital part of the human experience. The election of Barack Obama as the President of United States was partially attributed to his smart Internet strategy and access to millions of younger voters through a novel use of social media. As reported in the New York Times, in response to the disputed Iranian presidential election in 2009, Twitter has emerged as a global communication tool during the protest.

The increasing online traffic in social media brings about many data mining opportunities for user profiling, targeting, recommendation, and crowd-opinion analysis. Take social networking advertising as an example. Advertising in social media has encountered many challenges.[1] A common approach to targeting is to build a model that maps from user profiles (e.g., the geography location, education level, gender) to ad categories. Since social media often comes with a friendship network between users and many daily interactions, how can we exploit this rich user interaction information to infer the ads that might attract a user? This can be considered as a classification problem. The key task boils down to classifying users into relevant ad categories. For the classification problem, some labeled data can be collected. Online activities of users such as clicking on an ad, purchasing a product, or hobbies on their profiles reflect the users' potential interests. In reality, however, the label information is limited. Given a social network of users with their interaction information, we investigate *how to leverage this rich interaction information for classification tasks in social media.*

The analysis of social structures based on network topology has been studied in social sciences [49]. A traditional social science study involves the circulation of questionnaires, asking respondents to detail their interaction with others. Then a network can be constructed based on the response, with nodes representing individuals, and edges the interactions between them. This type of data collection confines most traditional social network analysis to a limited scale, typically hundreds of actors at most in one study. Various relations are present in one network. The relations between actors are often explicitly known, e.g., actor $a$ is the mother of actor $b$; actors $b$ and $c$ are colleagues. This type of relation information enables the anatomy of group process, group norm and social role analysis [49].

Diverse relations also make up the connections in large-scale networks present in social media. For example, one user might connect to her friends, relatives, college

[1] http://www.nytimes.com/2008/12/14/business/media/14digi.html.

classmates, colleagues, or online buddies with similar hobbies. The connections in a social network are inherently *heterogeneous*, representing various kinds of relationships between users. However, when a network is collected from social media, most of the time, no explicit information is available as to why these users connect to each other and what their relationships are. Existing methods that address classification problems with networked data seldom consider this heterogeneity. In other words, most existing methods treat inhomogeneous connections homogeneously. This can lead to undesirable results for a heterogeneous network. Therefore, we propose to differentiate various types of connections in building a discriminative classifier.

We present SocioDim, a novel classification framework based on latent social dimensions. Each dimension can be considered as the description of a plausible affiliation of actors that accounts for the interactions between them. With these social dimensions, we can take advantage of the power of discriminative learning, such as support vector machines or logistic regression, to automatically select relevant social dimensions for classification. The proposed framework is sufficiently flexible to allow the plug-in of different modules, and outperforms alternative representative methods on social media data. SocioDim also offers a simple yet effective approach to integrating network information with other features associated with actors, such as social content or profile information.

The paper is organized as follows. We formally state the classification problem with networked data in Sect. 2, and review existing work to handle the problem and discuss its limitations in Sect. 3. We propose an alternative framework to address the classification problem in networked data in Sect. 4. Some empirical results on real-world data are presented and discussed in Sects. 5 and 6. Further analysis of the framework is presented in Sect. 7. We present the related work in Sect. 8 and conclude in Sect. 9 by pointing out promising directions for future research using the proposed framework.

## 2 Problem statement

In this work, we study classification with networked data. For instance, in an advertising campaign, advertisers attempt to deliver ads to those users who are interested in their products, or similar categories. The outcome of user interests can be represented using + or −, with + denoting a user is interested in the product and − otherwise. We assume that the interests of some users are already known. This can be extracted from user profiles or their response to a displayed ad. The task is to infer the preference of the remaining users within the same social network.

In social media, some individuals are highly idiosyncratic. Their interests cannot be captured by merely one class. It is normal to have multiple interests in a user profile. Rather than concentrating on univariate cases of classification in networked data [29] (with each node having only one class label), we examine a more challenging task in which each node in a network can have multiple labels. In this setup, the univariate classification is just a special case. The multi-label classification problem with networked data can be formally described below:

*Given:*

– $K$ categories $\mathcal{Y} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_K\}$;

- a network $\mathcal{A} = (V, E, Y)$ representing the interactions between nodes, where $V$ is the vertex set, $E$ is the edge set, and each node $v_i$ is associated with class labels $\mathbf{y}_i$ whose value can be unknown;
- the known labels $Y^L$ for a subset of nodes $V^L$ in the network, where $V^L \subseteq V$ and $y_{ij} \in \{+, -\}$ denotes the class label of the vertex $v_i$ with respect to category $\mathcal{Y}_j$.

*Find:*

- the unknown labels $Y^U$ for the remaining vertices $V^U = V - V^L$.

Here, each vertex in the network represents one actor, or one user, in social media. Thereafter, data instances, actors, vertices, nodes, entities, and objects are used interchangeably in the context of a network.

The problem above is referred to as within-network classification [29]. It can be considered as a special case of relational learning [14], which is concerned with the modeling of domains that exhibit relational structures between objects. Here, we focus on the classification of objects when they are connected within only one network.

The problem we present deals with classification being based on network information alone. In reality, there might be more features associated with each node (actor). For example, in the blogosphere, the content of blog posts are available along with a blogger network. While it is an essential task to piece together these heterogeneous types of information, it is not the main focus here. In this work, we examine different approaches to classification based on network information alone and then discuss their extensions to handle additional actor features such as content information.

## 3 Collective inference and its limitations

In this part, we briefly review the commonly-used method to handle classification with networked data and discuss its limitations when applied directly to networks in social media.

### 3.1 Collective inference

When data instances are connected in a network, they are not independent and identically distributed (i.i.d.) as in conventional data mining. It is empirically demonstrated that linked entities have a tendency to belong to the same class [29]. This correlation in the class variable of connected objects can be explained by the concept of *homophily* in social science [30]. Homophily suggests that a connection between similar people occurs at a higher rate than among dissimilar ones. It is one of the first characteristics studied by early social network researchers [2,3,50], and holds for a wide variety of relationships [30]. Homophily is also observed in social media [10,45].

Based on the empirical observation that labels of neighboring entities (nodes) are correlated, the prediction of one node cannot be made independently, but also depends on its neighbors. To handle the interdependency, collective inference is widely used to address classification problems in networked data [20,29,37]. A common Markov assumption is that, the label of one node depends on that of its neighbors (plus other attributes if applicable). In particular,

$$P(y_i|\mathcal{A}) = P(y_i|\mathcal{N}_i). \quad (1)$$

Here $\mathcal{A}$ is the network, $y_i$ the label of node $v_i$, and $\mathcal{N}_i$ a set of its "neighbors". The neighbors are typically defined as nodes that are 1-hop or 2-hop away from $v_i$ in the network [20,12]. For training, a relational classifier based on the labels of neighbors (plus other available node attributes) is learned via the labeled nodes $V^L$. For prediction, collective inference [20] is applied to find an equilibrium status such that the inconsistency between neighboring nodes in the network is minimized. Relaxation labeling [5], iterative classification [26], and Gibbs sampling [13] are the commonly-used techniques. All the collective inference variants share the same basic principle: it initializes the labels of unlabeled nodes $V^U$, and then applies the constructed relational classifier to assign class labels (or update class membership) for each node while fixing the labels (or class membership) of its neighboring nodes. This process is repeated until convergence.

This collective inference procedure shares the same spirit as the harmonic function [55] in semi-supervised learning. It is shown in [29] that a simple weighted-vote relational neighborhood classifier [28] based on collective inference yields almost identical performance as the Gaussian field for semi-supervised learning [55].

### 3.2 Limitation of collective inference when applied directly to social media

A social network is often a composite of various relations. People communicate with their friends online. They may also communicate with their parents or random acquaintances. The diversity of connections indicates that two connected users do not necessarily share certain class labels. When relation type information is not available, directly applying collective inference to such a network cannot differentiate connections between nodes, and thus fails to predict the class membership of actors in the network. To give a palpable understanding, let us look at a toy example in Fig. 1. Actor 1 connects to Actor 2 because they work in the same IT company, and to Actor 3 because they often meet each other in the same sports club. Given the label information that Actor 1 is interested in both Biking and IT Gadgets, can we infer Actors 2 and 3's labels? Treating these two connections homogeneously, we guess that both Actors 2 and 3 are also interested in biking and IT gadgets. But if we know how Actor 1 connects to other actors, it is more reasonable to conjecture that Actor 2 is more interested in IT gadgets, and Actor 3 likes biking.

The example above assumes the cause of connections is explicitly known. But this kind of information is rarely explicit in real-world applications, though some social networking sites like Facebook and LinkedIn do ask connected users about the reason why they know each other. Most of the time, only network connections (as in Fig. 2) are available. If we can somehow differentiate the connections into different affiliations

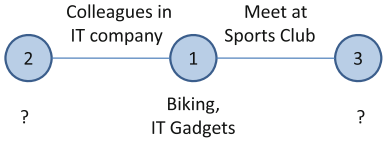



**Fig. 1** A toy example

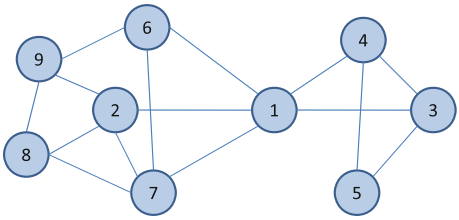



**Fig. 2** Node 1's local network

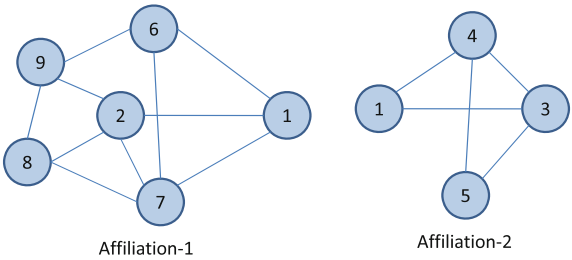



**Fig. 3** Different affiliations

(as shown in Fig. 3) and find out which affiliation is more correlated with the targeted class label, we can infer the class membership of each actor more precisely. Notice that an actor can be present in multiple affiliations. For instance, actor 1 belongs to both Affiliation-1 and Affiliation-2 in the example.

Given a network, differentiating its connections into distinct affiliations is not an easy task, as the same actor is involved in multiple affiliations. Moreover, one connection can be associated with more than one affiliation. For instance, one can connect to another as they are colleagues as well as going to the same sports club frequently. Rather than capturing affiliations among actors via connection differentiation, we resort to latent social dimensions, with each dimension representing a plausible affiliation of actors. Next, we introduce the concept of social dimensions and describe a classification framework based on that.

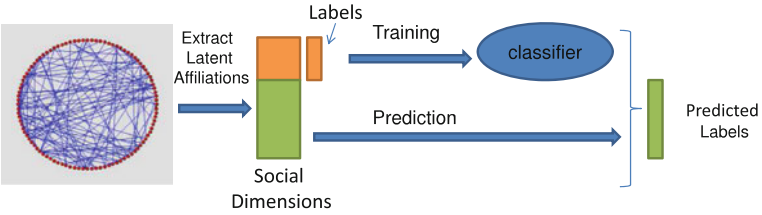
## 4 SocioDim: a framework to handle network heterogeneity

To handle the network heterogeneity as we have mentioned in the previous section, we propose to extract *social dimensions*, thereby capturing the latent affiliations of actors. Based on that, a classification framework is presented. Below, we introduce the concept of social dimensions and a fundamental assumption for our framework.

Social dimensions are the vector-format representation of actors' involvement in different affiliations. Given the extracted affiliations in Fig. 3, we can represent them as social dimensions in Table 1. If an actor is involved in one affiliation, then the entry of social dimensions corresponding to the actor and the affiliation is non-zero. Note that one actor can participate in multiple affiliations. E.g., actor 1 is associated with both affiliations. Different actors participate in disparate affiliations in varying extent.

**Table 1** Social dimensions corresponding to affiliations in Fig. 3

| Actor | Affiliation-1 | Affiliation-2 |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 0 |
| 3 | 0 | 1 |
| ... | ... | ... |




**Input:** Social network $\mathcal{A}$,
the labels of some nodes in the network $Y^L$,
the number of social dimensions to extract $k$;
**Output:** the labels of unlabeled nodes $Y^U$.

1. given $\mathcal{A}$ and $k$, extract social dimensions $S \in R^{n \times k}$ via soft clustering;
2. based on $S^L$ and $Y^L$, construct a discriminative classifier $C$;
3. based on $S^U$ and $C$, output $Y^U$ for unlabeled nodes.

**Fig. 4** SocioDim: a classification framework based on social dimensions

Numerical values, instead of boolean values, can also be used to represent affiliation membership.

We assume that the actor's label depends on his latent social dimensions. In particular, we assume

$$P(y_i|\mathcal{A}) = P(y_i|S_i) \tag{2}$$

where $S_i \in \mathbb{R}^k$ denotes the social dimensions (latent affiliations) of node $v_i$. This is fundamentally different from the Markov assumption in Eq. 1 used in collective inference, which assumes the label of one node relies on that of its neighbors. The Markov assumption does not capture the weak dependency between nodes that are not directly connected. In our approach, we assume the labels are determined by latent social dimensions. The nodes within the same affiliation tend to have similar labels even though they are not directly connected. Based on the assumption in Eq. 2, we propose a learning framework called SocioDim to handle the network heterogeneity for classification. The overview of the framework is shown in Fig. 4. The training is composed of two phases. We first extract the latent social dimensions $S_i$ for each node and then build a classifier based on the extracted dimensions to learn $P(y_i|S_i)$.

### 4.1 Phase I: extraction of social dimensions

For the first phase, we require the following:

- $A \in \mathbb{R}^{n \times n}$: an undirected network represented as a sparse matrix,
- $k$: the number of social dimensions to extract.

The output should be the social dimensions ($S \in \mathbb{R}^{n \times k}$) of all nodes in the network. It is desirable that the extracted social dimensions satisfy the following properties:

- Informative. The social dimensions should be indicative of latent affiliations of actors.
- Plural. The same actor can engage in multiple affiliations, thus having non-zero entries in different social dimensions.
- Continuous. The actors might have different degrees of association to one affiliation. Hence, a continuous value, rather than discrete {0, 1}, is more favorable.

One key observation is that when actors belong to the same affiliation, they tend to connect to each other. For example, people of the same department interact with each other more frequently than any two random people in a network. In order 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 community detection problem in networks. Most existing community detection methods partition nodes of a network into disjoint sets. But in reality, each actor is likely to subscribe to more than one affiliation. So a soft clustering scheme is preferred to extract social dimensions.

Many approaches developed for clustering on graphs serve the purpose of social dimension extraction, including modularity maximization [34], latent space models [18,36], block models [1,35] and spectral clustering [27]. Spectral clustering is originally proposed to address the partition of nodes in a graph. Spectral clustering has been shown to work reasonably well in various domains, including graphs, text, images, and microarray data. It is also proved [52] to be equivalent to a soft version of the classical *k-means* algorithm for clustering. We choose spectral clustering to extract social dimensions due to its effectiveness in various domains and the availability of a huge number of existing linear algebra packages to help solve the problem. We want to emphasize that this is not the only method of choice. Alternatives can be plugged in for this phase. This is also one nice feature of our framework, as it allows for convenient plug-ins of existing soft clustering packages. Later in the experiment part in Sect. 7.2, we will compare different strategies to extract social dimensions.

Below, we briefly review the principle of spectral clustering for presentation convenience. Spectral clustering was originally proposed to address the problem of partitioning a graph into disjoint sets. Here, the edges of a graph can have weights, denoting the similarity between nodes. Intuitively, we want to find a partition of the graph, so that the edges between groups have a small weight and the edges within a group have a large weight. This is closely related to the minimum-cut problem. For two disjoint vertex sets $B, C \subset V$, the cut between $B$ and $C$ is defined as

$$Cut(B, C) = \sum_{v_i \in B,\ v_j \in C} A_{ij}.$$

A k-way partition $(C_1, C_2, \ldots, C_k)$ should satisfy $\cup_{i=1}^{k} C_i = V$ and $C_i \cap C_j = \phi, \forall i \neq j$. The problem of finding a good k-way partition can be formulated as

$$\min cut(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} cut(C_i, V/C_i)$$

In practice, this might return trivial partitions like a group consisting of only one vertex separated from the remaining network. To find a somehow "balanced" partition, some alternative objectives are proposed to take into account the group size. One commonly used objective is *normalized cut* [38], which is defined below:

$$Ncut(C_1, \ldots, C_k) = \frac{1}{k} \sum_{i=1}^{k} \frac{cut(C_i, V/C_i)}{vol(C_i)} \tag{3}$$

where $vol(C_i) = \sum_{v_j \in C_i} d_j$, and $d_j$ represents the degree of node $v_j$. The coefficient $1/k$ is added to normalize the score between 0 and 1. If we define a community indicator matrix $H$ as

$$H_{ij} = \begin{cases} 1/\sqrt{vol(C_j)} & \text{if vertex i belongs to community } C_j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

It can be verified that,

$$Ncut(C_1, C_2, \ldots, C_k) = \frac{1}{k} Tr(H^T L H)$$

where $L$ is the graph Laplacian. In particular,

$$L = D - A$$

where $D = diag(d_1, d_2, \ldots, d_n)$ and $A$ is the network. Observe that $H^T D H = I$, so the minimization of Ncut can be written as

$$\begin{aligned} \min_{C_1,\ldots,C_k} \quad & Tr(H^T L H) \\ s.t. \quad & H^T D H = I \\ & H \text{ in form of Eq. 4} \end{aligned}$$

Due to the discreteness of partition, the problem is NP-hard to solve. Alternatively, we solve a relaxation of the problem as follows:

$$\begin{aligned} \min_{H} \quad & Tr(H^T L H) \\ s.t. \quad & H^T D H = I \end{aligned}$$

Let

$$S = D^{1/2} H, \tag{5}$$

**Table 2** Social dimensions extracted according to spectral clustering

| Node | Ideal | Case | Spectral clustering |
|---|---|---|---|
| 1 | 1 | 1 | −0.0893 |
| 2 | 1 | 0 | 0.2748 |
| 3 | 0 | 1 | −0.4552 |
| 4 | 0 | 1 | −0.4552 |
| 5 | 0 | 1 | −0.4643 |
| 6 | 1 | 0 | 0.1864 |
| 7 | 1 | 0 | 0.2415 |
| 8 | 1 | 0 | 0.3144 |
| 9 | 1 | 0 | 0.3077 |

the problem above can be reformulated as

$$\min_{S} \quad Tr(S^T \tilde{L} S) \tag{6}$$

$$s.t. \quad S^T S = I \tag{7}$$

where $\tilde{L}$ is the normalized Laplacian defined below

$$\widetilde{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}. \tag{8}$$

The optimal solution of $S$ corresponds to the first $k$ eigenvectors of the normalized graph Laplacian $\widetilde{L}$ with the smallest eigenvalues. Typically in spectral clustering, a post-processing step like k-means clustering is applied to $S$ or $H$ to find out a disjoint partition [27]. For our problem, however, disjoint partition is discouraged. So, we take the first few eigenvectors of $\widetilde{L}$ as the social dimensions extracted from the network $A$.

For example, if we apply spectral clustering to the toy network in Fig. 2, we obtain the social dimension in the last column in Table 2. In the table, we also show the ideal representation of the two affiliations in the network. Nodes 3, 4, and 5 belong to the same affiliation, thus sharing similar negative values in the extracted social dimension. Nodes 2, 6, 7, 8, and 9, associated with Affiliation-1, have similar positive values. Node 1, which bridges these two affiliations, has a value in between. The social dimension extracted based on spectral clustering does capture actor affiliations in certain degrees.

In summary, given a network $A$, we construct the normalized graph Laplacian $\widetilde{L}$ as in Eq. 8, and then compute its first $k$ smallest eigenvectors as the social dimensions. Note that $\widetilde{L}$ is sparse. So the power method or Lanczos method [15] can be used to calculate the top eigenvectors if $k$ is not too large. Many existing numerical optimization packages can be employed.

### 4.2 Phase II: classification learning based on social dimensions

This phase constructs a classifier with the following inputs:

– $\mathbf{Y}^L$: the labels of some nodes in the network $\mathcal{A}$,
– $S^L$: the social dimensions of the labeled nodes.

The social dimensions extracted in the first phase are deemed as features of data instances (nodes). We conduct conventional supervised learning based on the social dimensions and the label information. A discriminative classifier like support vector machine (SVM) or logistic regression can be used. Other features associated with the nodes, if available, can also be included during the discriminative learning. This phase is critical, as the classifier will determine which dimensions are relevant to the class label. A linear SVM is exploited in this work due to its simplicity and scalability [42].

One concern with spectral clustering is that the obtained dimensions are not unique. Let $S$ be the extracted dimensions based on Eq. 7, and $P$ be an orthonormal matrix such that $P \in R^{k\times k}$, $P^T P = PP^T = I_k$. It can be verified that $S' = SP$ is a solution with the same objective:

$$Tr((S')^T \widetilde{L}(S')) = Tr((SP)^T \widetilde{L}(SP)) = Tr(S^T \widetilde{L} SPP^T) = Tr(S^T \widetilde{L} S)$$

Essentially, the solutions are equivalent under an orthogonal transformation. But this non-uniqueness does not affect the discriminative learning if a linear SVM is employed. The linear SVM with social dimensions $S$ can be considered as a kernel machine with a linear kernel $\mathcal{K} = SS^T$. With an orthogonal transformation $P$, the new kernel $\mathcal{K}'$ does not change:

$$\mathcal{K}' = S'S'^T = (SP)(SP)^T = SPP^T S^T = SS^T = \mathcal{K}. \tag{9}$$

It follows that the classifier and thus its predictions remain the same. Therefore, the overall classification is not affected by the non-uniqueness of social dimensions in the previous phase.

### 4.3 Phase III: prediction

The prediction phase requires:

– $C$: the constructed classifier based on training,
– $S^U$: the social dimensions of those unlabeled nodes in the network.

Prediction is straightforward once the classifier is ready, because social dimensions have been calculated in Phase I for all the nodes, including the unlabeled ones. We treat social dimensions of the unlabeled nodes as features and apply the constructed classifier to make predictions. Different from existing within-network classification methods, collective inference becomes unnecessary. Though the distribution of actors does not follow the conventional i.i.d. assumption, the extracted social dimensions in Phase I already encode correlations between actors along with the network. Each node can be predicted independently without collective inference. Hence, this framework is efficient in terms of prediction.

We emphasize that this proposed framework, SocioDim, is flexible. We choose spectral clustering to extract social dimensions and SVM to build the classifier.

This does not restrict us from using alternative choices. Any soft clustering scheme can be used to extract social dimensions in the first phase. The classification learning phase can also be replaced with any classifier other than SVM. This flexibility enables the convenient use of many existing software packages developed for clustering or classification. One minor issue is that, depending on the adopted classifier, the final decision function might not be unique because of the non-uniqueness of social dimensions as discussed before. It is beyond the scope of this work to study how the final classification performance fluctuates with respect to social dimensions under different transformations.

## 5 Experiment setup

In the experiment below, we will compare our proposed SocioDim framework with representative collective-inference methods when heterogeneity is present in a network. Before we proceed to the details, we describe the data collected for experiments and the baseline methods for comparison.

### 5.1 Data sets

In this work, we focus on classification tasks in social media. We shall examine how different approaches behave in real-world social networks. Two data sets are collected: one from BlogCatalog[2] and the other from the popular photo sharing site Flickr:[3]

- *BlogCatalog* A blog in BlogCatalog is associated with various information pieces such as the categories the blog is listed under, blog level tags, snippets of 5 most recent blog posts, and blog post level tags. Bloggers submit their blogs to Blog Catalog and specify the metadata mentioned above for improved access to their blogs. This way the blog sites are organized under pre-specified categories. A blogger also specifies his connections with other bloggers. A blogger's interests could be gauged by the categories he publishes his blogs in. Each blogger could list his blog under more than one category. Note that we only crawl a small portion of the whole network. Some categories occur rarely, and they demonstrate no positive correlation between neighboring nodes in the network. Thus, we pick 39 categories with a reasonably large sample pool for evaluation purposes. On average, each blogger lists their blog under 1.6 categories.
- *Flickr* is a popular website to host photos uploaded by users, and it has become an active community platform. Users in Flickr can tag photos and add contacts. Users can also subscribe to different interest groups ranging from "black and white photos"[4] to a specific subject (say "bacon"[5]). Among the huge network and numerous groups on Flickr, we randomly chose around 200 interest groups as the

[2] http://www.blogcatalog.com/.

[3] http://www.flickr.com/.

[4] http://www.flickr.com/groups/blackandwhite/.

[5] http://www.flickr.com/groups/everythingsbetterwithbacon/.

**Table 3** Statistics of social network data

| Data | BlogCatalog | Flickr |
|---|---|---|
| Categories (K) | 39 | 195 |
| Actors (n) | 10, 312 | 80, 513 |
| Links (m) | 333, 983 | 5, 899, 882 |
| Network density | $6.3 \times 10^{-3}$ | $1.8 \times 10^{-3}$ |
| Maximum degree | 3, 992 | 5,706 |
| Average degree | 65 | 146 |
| Average labels | 1.4 | 1.3 |
| Average category NCut | 0.48 | 0.46 |

class labels and crawled the contact network among the users subscribed to these groups for our experiment. The users with only one single connection are then removed from the data set.

Table 3 lists some statistics of the network data. As seen in the table, the connections among social actors are extremely sparse. The degree distribution is highly imbalanced, a typical phenomenon in scale-free networks. We also compute the normalized cut score (Ncut) for each category and report the average Ncut. Note that if a category is nearly isolated from the rest of a network, the Ncut score should be close to 0. Clearly, for both data sets, the categories are not well-separated from the rest of the network, demonstrating the difficulty of the classification tasks. Both data sets are publicly available from authors' homepages.

### 5.2 Baseline methods

We apply SocioDim to both data sets. Spectral clustering is employed to extract social dimensions. The number of latent dimensions is set to 500 and one-vs-rest linear SVM is used for discriminative learning. We also compare SocioDim to two representative relational learning methods based on collective inference (Weighted-Vote Relational Neighbor Classifier [28] and Link-Based Classifier [26]), and two baseline methods without learning (a Majority Model and a Random Model):

– Weighted-Vote Relational Neighbor Classifier (wvRN). wvRN [28] works like a lazy learner. No learning is conducted during training. In prediction, the relational classifier estimates the class membership $p(\mathbf{y}_i|\mathcal{N}_i)$ as the weighted mean of its neighbors.

$$p(\mathbf{y}_i|\mathcal{N}_i) = \frac{1}{\sum_j w_{ij}} \sum_{v_j \in \mathcal{N}_i} w_{ij} \cdot p(\mathbf{y}_j|\mathcal{N}_j) \tag{10}$$

$$= \frac{1}{|\mathcal{N}_i|} \sum_{\{j:(v_i,v_j)\in E\}} p(\mathbf{y}_j|\mathcal{N}_j) \tag{11}$$

where $w_{ij}$ in (10) are the weights associated with the edge between node $v_i$ and $v_j$. Eq. 11 is derived, because the networks studied here use {0, 1} to represent connections between actors, and we only consider the first order Markov assumption that the label of one actor depends on his connected friends). Collective inference is exploited for prediction. We iterate over each node of the network to predict its class membership until the membership change is small enough. WVrn has been shown to work reasonably well for classification in the univariate case and is recommended as a baseline method for comparison [29].

– Network Only Link-Based Classifier (LBC) [26]. This classifier creates relational features of one node by aggregating the label information of its neighbors. Then, a relational classifier can be constructed based on labeled data. In particular, we use average class membership (as in Eq. 11) of each class as relational features, and employ SVM to build the relational classifier. For prediction, relaxation labeling [5] is utilized as the collective inference scheme.
– Majority Model (MAJORITY). This baseline method uses the label information only. It does not leverage any network information for learning or inference. It simply predicts the class membership as the proportion of positive instances in the labeled data. All nodes are assigned the same class membership. This model is inclined to predict categories of larger size.
– Random Model (RANDOM). As indicated by the name, this model predicts the class membership for each node randomly. Neither network nor label information is used. This model is included for the relative comparison of various methods.

In our experiments, actors often have more than one label. We apply the methods above to each category independently and report the average performance. Since most methods yield a ranking of labels rather than an exact assignment, a thresholding process is normally required. It has been shown that different thresholding strategies lead to quite different performances [9,42]. To avoid the effect of thresholding, we assume the number of labels on the test data is already known and check how the top-ranking predictions match with the true labels. Two commonly used measures Micro-F1 and Macro-F1 [42] are adopted to evaluate the classification performance.

## 6 Experiment results

In this section, we experimentally examine the following questions: How is the classification performance of our proposed framework compared to that of collective inference? Does differentiating heterogeneous connections presented in a network yield a better performance?

### 6.1 Performance on BlogCatalog data

We gradually increase the number of labeled nodes from 10% to 90%. For each setting, we randomly sample a portion of nodes as labeled. This process is repeated 10 times, and the average performance is recorded. The performances of different methods and the standard deviation are plotted in Fig. 5. Clearly, our proposed SocioDim

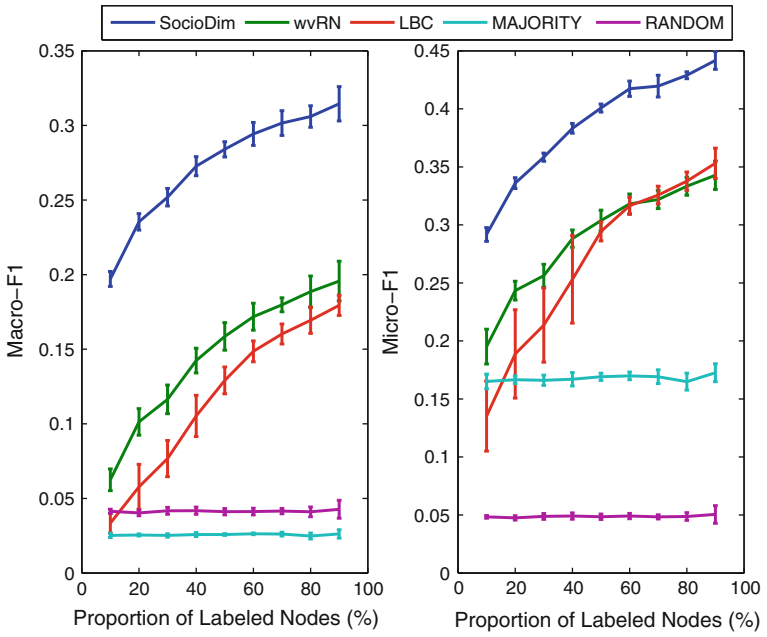



**Fig. 5** Performance on BlogCatalog with 10, 312 nodes (better viewed in color)

outperforms all the other methods. wvRN, as shown in the figure, is the runner-up most of the time. MAJORITY performs even worse than RANDOM in terms of Macro-F1, as it always picks the majority class for prediction.

The superiority of SocioDim over other relational learning methods with collective inference is evident. As shown in the figure, the link-based classifier (LBC) performs poorly with few labeled data. This is because LBC requires a relational classifier before the inference. When samples are few, the learned classifier is not robust enough. This is indicated by the large deviation of LBC in the figure when labeled samples are less than 50%. We notice that LBC in this case takes many iterations to converge. wvRN is more stable, but its performance is not comparable to SocioDim. Even with 90% of nodes being labeled, a substantial difference between wvRN and SocioDim is still observed. Comparing all three methods (SocioDim, wvRN and LBC), SocioDim is the most stable and achieves the best performance.

### 6.2 Performance on Flickr data

Compared to BlogCatalog, the Flickr network is larger, with around 100,000 nodes. In practice, the label information in large-scale networks is often very limited. Here we examine a similar case. We change the proportion of labeled nodes from 1% to 10%. Roughly, the number of labeled actors increases from around 1, 000 to 10, 000. The performances are reported in Fig. 6.

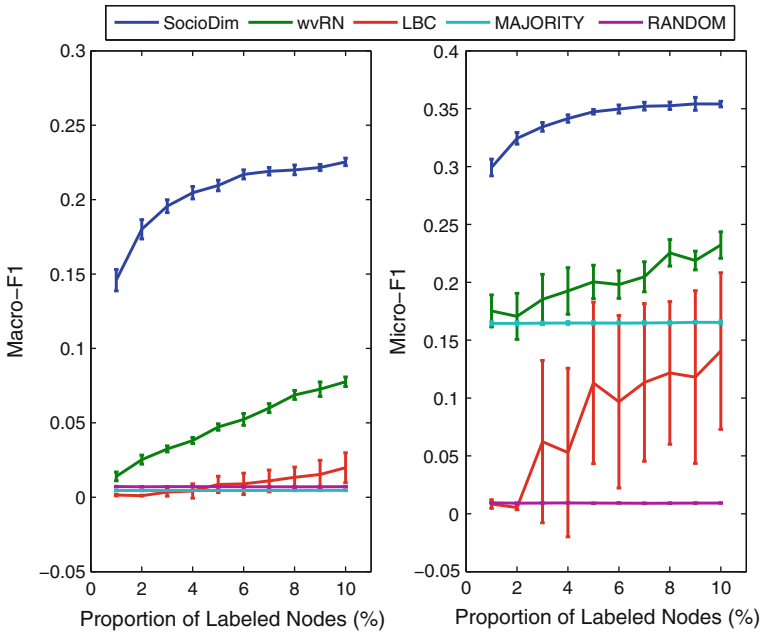



**Fig. 6** Performance on Flickr with 80, 513 nodes (Better viewed in color)

The methods based on collective inference, such as wvRN and LBC, perform poorly. The LBC fails most of the time (almost like random) and is highly unstable. This can be verified by the fluctuation of its Micro-F1. LBC tries to learn a classifier based on the features aggregated from a node's neighbors. The classifier can be problematic when the labeled data are extremely sparse and the network is noisy as presented here. While alternative collective inference methods fail, SocioDim performs consistently better than other methods by differentiating heterogeneous connections within the network.

It is noticed the prediction performance on both data sets is around 20–30% for F1-measure, suggesting that social media networks are very noisy. As shown in later experiments, the performance can be improved when other actor features are also included for learning.

## 6.3 Understanding the SocioDim framework

In the previous subsections, we show that SocioDim outperforms representative methods based on collective inference. *Why does SocioDim demonstrate better performance than collective inference?* We will explore different hypotheses to better understand the SocioDim framework.

### *6.3.1 $H_1$: Does SocioDim win because a given network is too sparse?*

One might suspect that the poor performance of collective inference is due to the sparsity of a given network. As shown in Table 3, the density of the BlogCatalog network

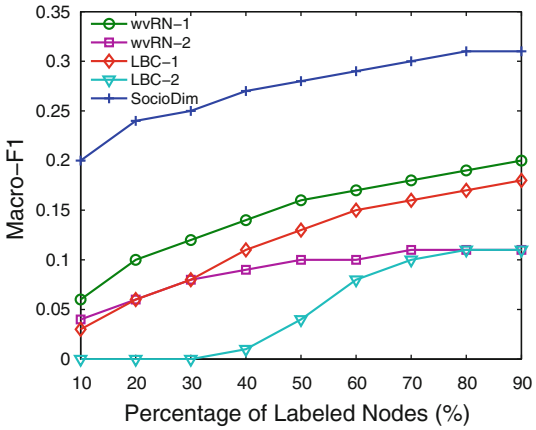



**Fig. 7** Performances of collective inference by expanding the neighborhood

is only $6.3 \times 10^{-3}$. Flickr is even sparser. When a network is too sparse, Gallagher et al. [12] suggest expanding the neighborhood of one node by adding "ghost edges" to connect those nodes that are 2-hop away. Following their idea, we construct a network by linking all nodes that are within 2 hops. After the expansion for BlogCatalog, the network density leaps to $6.16 \times 10^{-1}$. We cannot expand any more as the network becomes almost a complete graph when nodes within 3 hops are connected. Flickr becomes quite dense after a 2-hop expansion, causing computational problems. Therefore, we report the performance of collective inference of the expanded network on BlogCatalog only in Fig. 7, where $i$ denotes the number of hops to consider for defining the neighborhood. For both wvRN and LBC, the performance deteriorates after the neighborhood is expanded. This is because of the increase of connections. Though it alleviates the sparsity problem, it also introduces more heterogeneity. Collective inference, no matter how we define the neighborhood, is not comparable to SocioDim.

### 6.3.2 *$H_2$: Does SocioDim win because nodes within a category are well isolated from the rest of a network?*

Another hypothesis is related to the community effect present in a network. Since SocioDim relies on soft community detection to extract social dimensions, and the networks we studied are extremely sparse, one might suspect SocioDim wins because there are very few inter-category edges. Intuitively, when a category is well-isolated from the whole network, the clustering in the first phase of SocioDim captures this structure, thus it defeats collective inference. Surprisingly, *this intuition is not correct*. Based on our empirical observation, when a category is well-isolated, there is not much difference between wvRN and SocioDim. SocioDim's superiority is more observable when the nodes of one category are blended into the whole network.

In order to calibrate whether the nodes of one category are isolated from the rest of a given network, we compute the *category Ncut* following Eq. 3. Given a category, we split a network into two sets: one set containing all the nodes of the category, the other

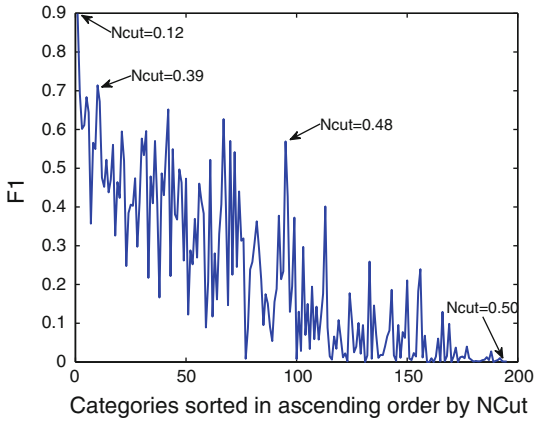



**Fig. 8** Performance of SocioDim in individual categories

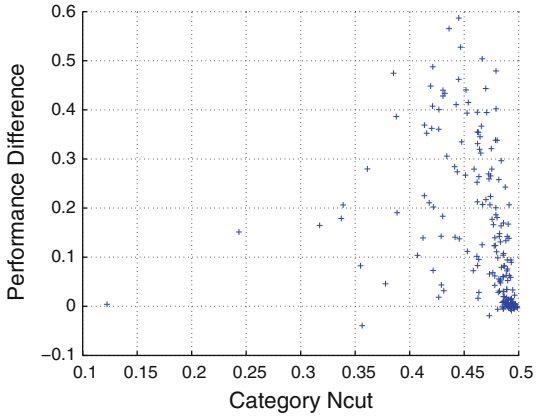



**Fig. 9** Performance improvement of SocioDim over wvRN wrt. Category Ncut

set all the nodes not belonging to that category. The normalized cut can be computed. If a category is well-isolated from a network, the Ncut should be close to 0. The larger the NCut is, the less the category is isolated from the remaining network. The average category Ncut scores on BlogCatalog and Flickr are 0.48 and 0.46, respectively, as reported in Table 3. This implies that most categories are actually well connected to the remaining network, rather than being an isolated group as one might suppose.

Figure 8 shows the performance of SocioDim on the 195 individual categories of the Flickr data when 90% of nodes are labeled. As expected, SocioDim performance tends to decrease when the Ncut increases. An interesting pattern emerges when we plot the improvement of SocioDim over wvRN with respect to category Ncut in Fig. 9. The largest improvements happen for those categories that are not well-isolated. Notice the plus at the bottom left, which corresponds to the case when $Ncut = 0.12$. For this category, SocioDim achieves 90% F1 as shown in Fig. 8. But the improvement over wvRN is almost 0 as in Fig. 9. That is, $wvRN$ is comparable. Most of SocioDim's improvements occur when $Ncut > 0.35$. Essentially, when a category is not well-isolated from the remaining network, SocioDim tends to outperform wvRN substantially.

**Table 4** Statistics of *imdb* data

| Data | Size | Density | Ave. degree | Base acc. | Category Ncut |
|---|---|---|---|---|---|
| $imdb_{\text{prodco}}$ | 1126 | 0.035 | 38.8 | 0.501 | 0.24 |
| $imdb_{\text{all}}$ | 1371 | 0.049 | 67.3 | 0.562 | 0.36 |

### 6.3.3 $H_3$: Does SocioDim win because it addresses network heterogeneity?

In the previous subsection, we notice that SocioDim performs considerably better than wvRN when a category is not so well-separated. We attribute the gain to taking into account connection heterogeneity. Hence, we adopt a benchmark relational data (*imdb* used in [29]) with varying heterogeneity. The *imdb* network is collected from the Internet Movie Data base, with nodes representing 1377 movies released between 1996 and 2001. The goal is to estimate whether the opening weekend box-office receipts of a movie "will" exceed 2 million dollars. Two versions of network data are constructed: $imdb_{prodco}$ and $imdb_{all}$. In $imdb_{prodco}$, two movies are connected if they share a production company. While in $imdb_{all}$, they are connected if they share actors, directors, producers or production companies. Clearly, the connections in $imdb_{all}$ are more heterogeneous. Both network data sets have one giant connected component each, with others being singletons or trivial-size components. Here, we report the performance on the largest components.

We notice that these two data sets demonstrate different characteristics from the previously-studied social media data. (1) the connections are denser. For instance, the density of $imdb_{all}$ is 0.049 (7 times denser than BlogCatalog and 27 times than Flickr). (2) the classification task is also much easier. It is a binary classification task. The class distribution is balanced, different from the imbalanced distribution present in the social media data. Hence, we report classification performance in terms of accuracy as in [29]. Finally, (3) classes are well separated as suggested by the low category Ncut in Table 4.

Figure 10 plots the performance of SocioDim and wvRN on $imdb_{prodco}$ and $imdb_{all}$. When connections are relatively homogeneous (e.g., $imdb_{prodco}$ data), SocioDim and wvRN demonstrate comparable classification performance. When connections become heterogeneous, the category NCut increases from 0.24 to 0.36. The performances of both methods decrease as shown in Fig. 10b. For instance, with 50% of nodes labeled, SocioDim's accuracy decreases from 80% to about 77%, but wvRN's performance drops severely, from 80% to around 66% with introduced heterogeneity.

We notice that the performance decrease of wvRN is most observable when labeled data are few. With increasing labeled data, wvRN's performance climbs up. The comparison on the two networks of distinctive degree of heterogeneity confirms our original hypothesis: *SocioDim, by differentiating heterogeneous connections, performs better than collective inference*. This effect is more observable when a network presents heterogeneity and the labeled data are few.

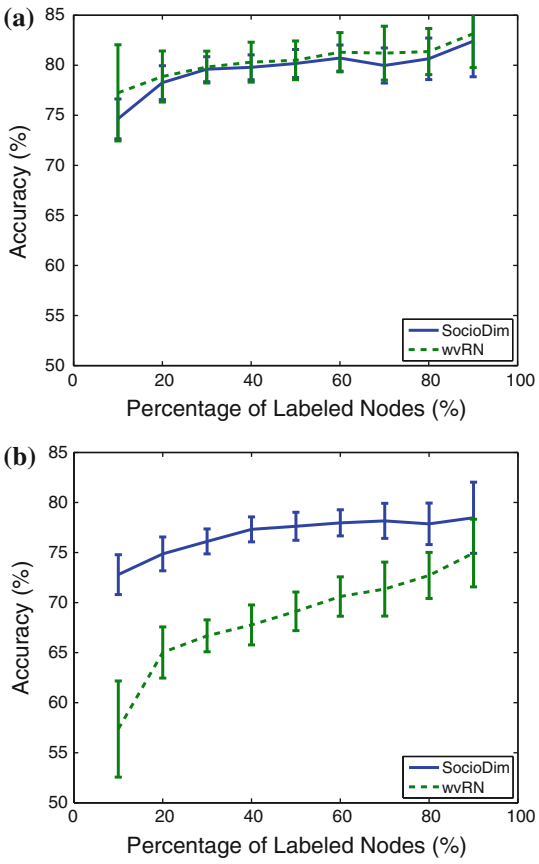



**Fig. 10** Classification performance on *imdb* network. **a** $imdb_{prodco}$. **b** $imdb_{all}$

## 6.4 Visualization of extracted social dimensions

In order to get some tangible idea of extracted social dimensions, we examine tags associated with each dimension. It is impractical to show tag clouds of all the extracted dimensions (500 social dimensions for both data sets). Thus, given a category, we investigate its dimension with the maximum SVM weight and check whether it is truly informative about the category.

Since we use soft clustering to extract social dimensions, each dimension is represented by continuous values (as in the last column in Table 2). For simplicity, we pick the top 20 nodes with the maximum positive values as representatives of one dimension. For example, nodes 8, 9, 7 and 6 are the top 4 nodes for the social dimension extracted following spectral clustering in Table 2. Tags of those representative nodes in one dimension are aggregated as the tags of that dimension. For the sake of clear visualization, only those tags that occur more than once are kept in a tag cloud with font size denoting their relative frequency.

Due to the space limit, we showcase only two examples from BlogCatalog. Figure 11 lists the tag clouds of selected social dimensions for categories *Health* and *Animal*,

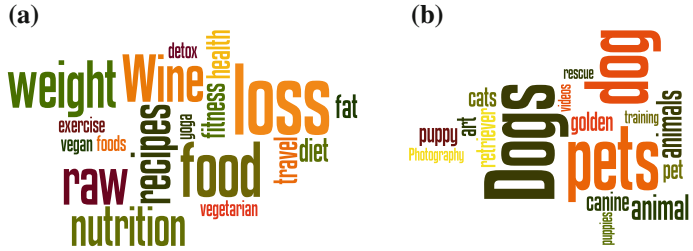



**Fig. 11** Social dimensions selected by *health* and *animal*, respectively. **a** Health. **b** Animal

respectively. Evidently, both are quite relevant to their target categories. Based on the tag cloud in Fig. 11a, it is not difficult to figure out that the social dimension is about food and weight loss, which is highly relevant to *Health*. Similarly, the social dimension in Fig. 11b is about dogs and pets, thus relevant to *Animal*. These examples suggest that our extracted social dimensions are sensible, and that relevant dimensions can be selected accordingly by the classification learning phase of the SocioDim framework.

## 7 Further analysis of SocioDim

In the previous section, we have shown that SocioDim outperforms methods based on collective inference. In this section, we analyze properties of SocioDim from different aspects. The following questions will be explored:

- Social media provides more than mere actor network information. How can we include different types of information in the classification framework?
- Are there any other strategies to extract social dimensions? How do they perform compared with spectral clustering?
- The current instantiation of SocioDim requires users to provide a parameter to set the number of social dimensions. How sensitive is the classification performance to this parameter?

### 7.1 Integration of actor network and actor features

In social media, various kinds of user information besides social networks can be collected. For instance, in blogosphere, people post blogs, write comments and upload tags. Some users also provide profile information. It is desirable to utilize all the information available to achieve more accurate classification. However, the actor features (e.g., user profiles, social content or tag information) and the networks are presented in disparate formats, hence some efforts are required for the integration.

One nice property of SocioDim is that it converts a network into features. So, if actor features are available, it is straightforward to couple the network features with actor features: simply combine the extracted social dimensions with actor features, and the discriminative learning procedure determines which features are more informative

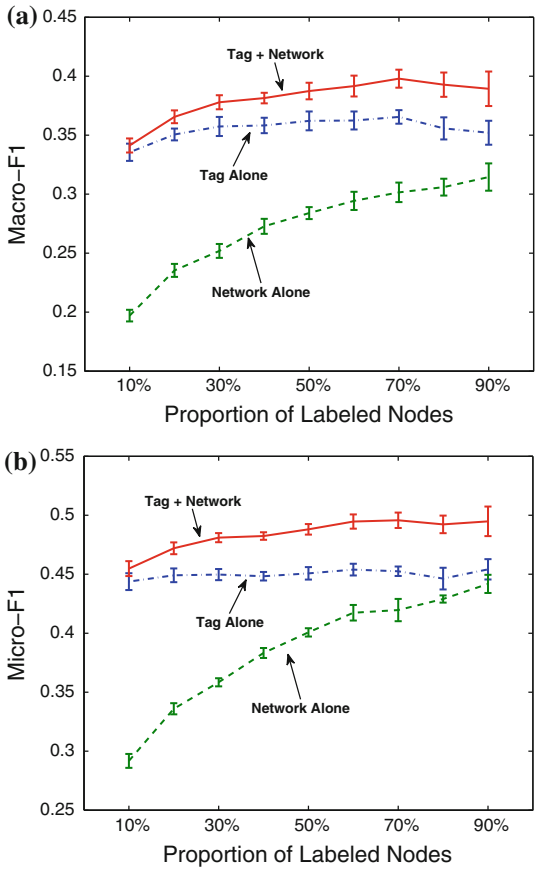



**Fig. 12** Performance of network with actor features on BlogCatalog. **a** Macro-F1. **b** Micro-F1

of a class label. This simple combination of network information and actor features allows for integration of data in disparate format and can lead to more accurate classification in general. Here we take BlogCatalog as an example to show the effect. In BlogCatalog, the blogger can upload some tags descriptive of his blog site. We use tag information as actor features for the bloggers. The performance of using tag or network alone, or the combination of the two, are plotted in Fig. 12.

Tags are normally quite descriptive of a blogger while networks tend to be noisy. It should not be surprising that the performance based on tags alone is better than the performance based on networks. It is noticed that increasing the labeled samples does not help much for performance based on tags, partly because some users do not provide tags. But if we combine the social dimensions extracted from a network with the tag features, the performance is increased by 3–6%. Networks in social media are noisy. They provide complementary, though maybe weak, information of user interests. There are other relational models to capture the dependency of connected nodes and additional attributes (e.g.,[44,43]), but they normally require a lot of effort.

Our proposed SocioDim provides a *simple* yet *effective* approach to integrating network information and actor features for accurate classification.

### 7.2 Alternative strategies to extract social dimensions

Each social dimension represents one latent affiliation. One actor is allowed to participate in multiple different affiliations, hence a soft clustering scheme is preferred to extract social dimensions. In previous parts, we adopted spectral clustering, which is proved to equal to a soft version of k-means partition [52], to extract social dimensions. One basic question is: *how different is the performance of a hard partition from that of a soft clustering?* Spectral clustering involves the calculation of the top eigenvectors of a normalized graph Laplacian whereas a k-means partition algorithm is normally faster and more scalable. Should these two methods be comparable in terms of classification performance, then a k-means partition should be preferred.

For completeness, we also explore other alternative soft clustering schemes. Recently, modularity [33] is proposed to calibrate the strength of community structure in scale-free networks. Modularity is like a statistical test where the null model is a uniform random graph in which one actor connects to others with uniform probability while keeping the same degree as a given network. Consider dividing the interaction matrix $A$ of $n$ vertices and $m$ edges into $k$ non-overlapping communities. Let $s_i$ denote the community membership of vertex $v_i$, and $d_i$ represents the degree of vertex $i$. For two nodes with degree $d_i$ and $d_j$ respectively, the expected number of edges between the two in a uniform random graph model is $d_i d_j/2m$. Modularity measures how far the interaction deviates from a uniform random graph with the same degree distribution. It is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[ A_{ij} - \frac{d_i d_j}{2m} \right] \delta(s_i, s_j) \tag{12}$$

where $\delta(s_i, s_j) = 1$ if $s_i = s_j$, and 0 otherwise. A larger modularity indicates denser within-group interaction. Note that $Q$ could be negative if the vertices are split into bad clusters. $Q > 0$ indicates the clustering captures some degree of community structure. In general, one aims to find a community structure such that $Q$ is maximized. Modularity maximization can be relaxed in a similar way to spectral clustering [33]. The community indicators are the top largest eigenvectors of a modularity matrix defined below:

$$B = A - \frac{\mathbf{d}\mathbf{d}^T}{2m} \tag{13}$$

Modularity maximization is adopted to extract social dimensions in our preliminary work [39] and has been shown to outperform methods based on collective inference.

Three representative clustering methods are compared: k-means partition, modularity maximization and spectral clustering. Following the SocioDim framework, we apply these methods to extract social dimensions, respectively. Then SVM is employed

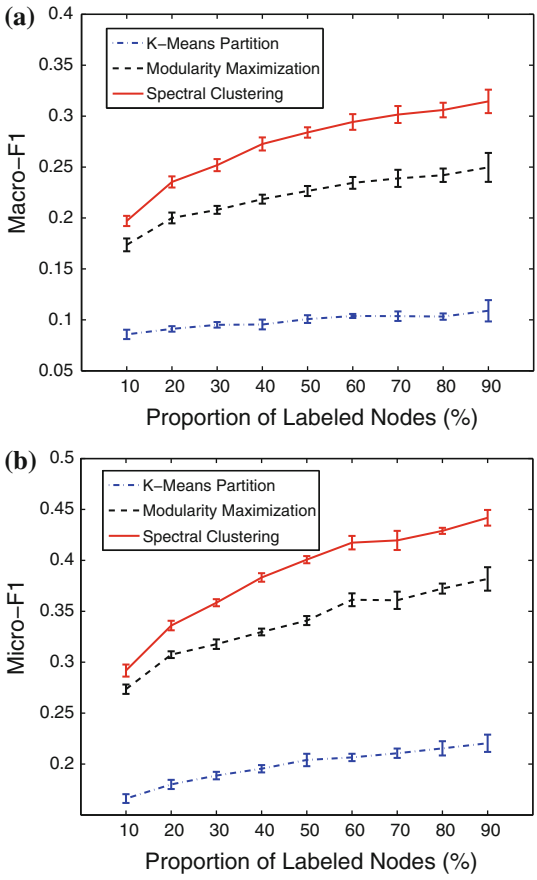



**Fig. 13** Different strategies of social dimension extraction on BlogCatalog. **a** Macro-F1. **b** Micro-F1

for the subsequent discriminative learning. To be fair, we fix the dimensionality to 500 for all the methods. For k-means, we adopt a similar strategy as in [19] by considering the connections of each user as features and using k-means with cosine similarity for clustering. For modularity maximization, we compute the top eigenvectors of the modularity matrix defined in Eq. 13. The performances on BlogCatalog and Flickr are plotted in Figs. 13 and 14, respectively.

Clearly, different methods yield quite different performances. This indicates that the extraction of social dimensions can be crucial to our SocioDim framework. The difference of soft clustering and hard partition is evident on BlogCatalog. Both spectral clustering and modularity maximization outperform k-means partition. When the network scales to a larger size as in Flickr, modularity maximization does not show a strong superiority over hard partition. Indeed, the performance of modularity maximization and that of k-means partition are comparable on Flickr. Spectral clustering, on the contrary, excels in all cases. Spectral clustering seems to capture the latent affiliations more accurately for within-network classification. A related study shows that maximizing modularity tends to find communities composed of small clusters [11].

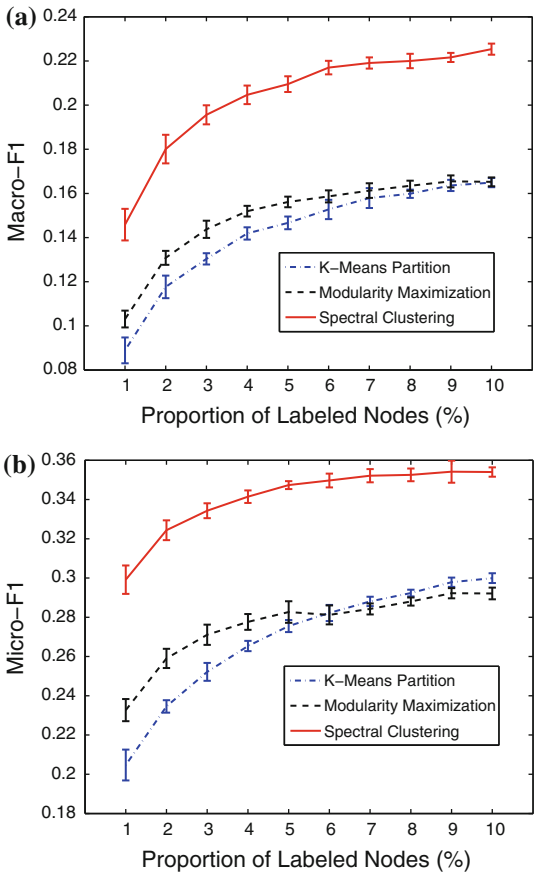



**Fig. 14** Different strategies of social dimension extraction on Flickr. **a** Macro-F1. **b** Micro-F1

This might explain why modularity maximization is inferior to spectral clustering in our experiments.

In summary, soft clustering, consistent with the social dimension concept, outperforms hard partition for social dimension extraction. Based on our empirical experience, spectral clustering is a good candidate to extract social dimensions. Of course, other strategies can also be explored for more accurate classification.

### 7.3 Sensitivity to social dimensionality

In the experiments above, the social dimensionality is fixed to 500 for SocioDim. In this subsection, we examine how the performance fluctuates with a varying number of social dimensions. On both data sets, we vary the dimensionality from 100 to 1, 000 and observe its performance variation. The respective performance changes on BlogCatalog and Flickr are plotted in Fig. 15. To make the figure legible, we only plot the cases when 10, 50 or 90% of nodes in the network are labeled on BlogCatalog, and 1, 5 or 9% on Flickr.

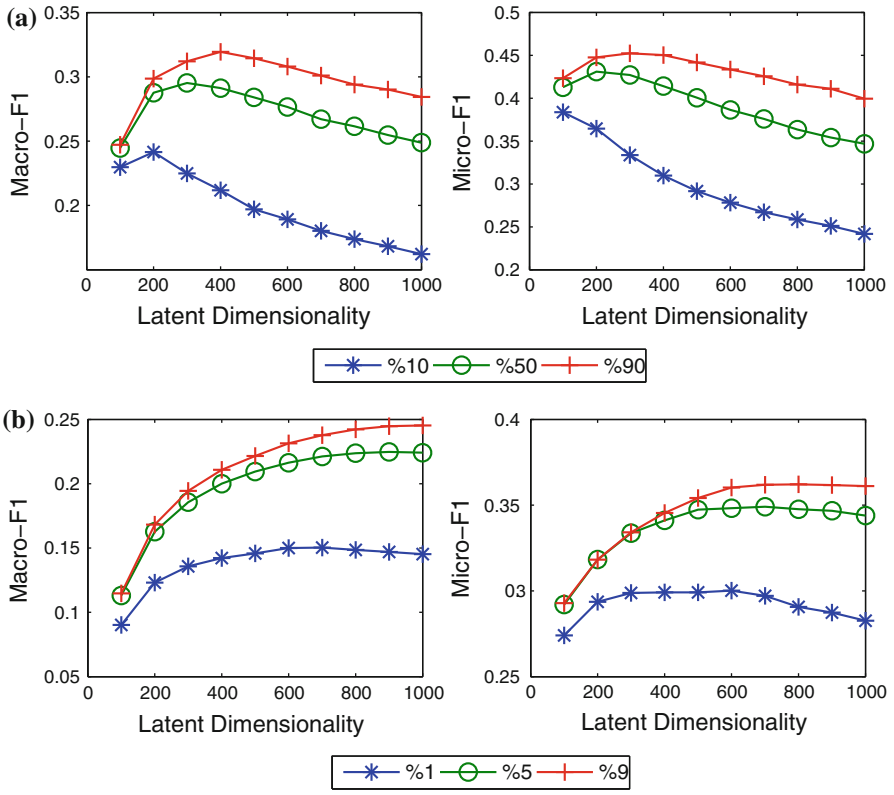



**Fig. 15** SocioDim sensitivity to latent dimensionality. **a** BlogCatalog. **b** Flickr

As seen in the figure, the performance on BlogCatalog peaks at around 200–400 dimensions and decreases with more social dimensions. For Flickr, the performance stabilizes after 500–600 dimensions. If fewer (<200) latent social dimensions are selected, then some discriminative dimensions might be missed and thus the performance deteriorates. Flickr requires much more social dimensions than BlogCatalog. This agrees with our intuition about social dimensions. As each dimension represents one latent affiliation, more affiliations are presented in a larger network, generally.

Another phenomenon is that the optimal dimensionality increases with the number of labeled samples. For instance, for macro-F1 on BlogCatalog, the best performance is achieved when only 200 dimensions are selected with 10% of labeled nodes. This number increases to 300 when 50% of nodes are labeled, and 400 with 90% of nodes labeled. On Flickr data, the desired number of social dimensions to reach the best performance also correlates positively with the dimensionality (300, 500 and 600 respectively when 1, 5 and 9% of nodes are labeled for micro-F1 on Flickr as in Fig. 15b). Essentially, when more nodes are labeled, we have to zoom into affiliations of finer granularity to capture user interests.

In practice, the optimal dimensionality depends on the network size and the number of labeled nodes. Generally, it correlates positively with these statistics mentioned

above. This provides a high-level guideline to set the parameter, which can save some time if extensive cross validation is required.

## 8 Related work

As our SocioDim framework addresses within-network classification, we review literature about relational learning and semi-supervised learning. On the other hand, community detection is a key component in our framework to extract social dimensions. It is included for discussion as well.

### 8.1 Relational learning

Relational learning [14] refers to the classification when objects or entities are presented in multiple relations or network format. In this work, we study a special case: within-network classification [29] when the objects are connected in one network. The data instances in the network are not independently identically distributed (i.i.d.) as in conventional data mining. In order to capture the correlation between labels of neighboring data objects, a Markov dependency assumption is widely adopted. That is, the label of one node depends on the labels (and attributes) of its neighbors. Based on the assumption, collective inference [20] is proposed for prediction. Normally, a relational classifier is constructed based on the relational features of labeled data, and then an iterative process is required to determine class labels for unlabeled data. It is shown that a simple weighted vote relational neighborhood classifier [28] works reasonably well on some benchmark relational data and is recommended as a baseline for comparison [29].

In our implementation of collective inference, we define the neighborhood to be the nodes that are only 1-hop away. Gallagher et al. [12] propose to add "ghost edges" before relational learning when the network is too sparse. The ghost edges essentially connect nodes that are 2 hops away. After the expansion of the neighborhood of one node for collective inference, a better classification performance is observed. However, this strategy cannot be applied to networks in social media. In social networks, the small-world effect [46] is often observed [4]. That is, any pair of nodes in a large-scale social network are only several hops away, relating to the well-known "six degree of separation". For instance, in our Flickr data, the average degree of one node is 146. Roughly, the nodes that are two hops away from one node can be as high as $146 \times 146 = 21,316$. Of course, this number is not precise as the friends of friends may overlap. This huge number of neighbors brings in much more noise and heterogeneity in connections, which can worsen the performance of collective inference. This is empirically verified in a smaller BlogCatalog network in Sect. 6.3.1. Often, a network becomes very dense after neighborhood expansion. As a result, the scalability can be a concern as well.

There are many more complicated relational models to model the dependence between connected entities. For instance, probabilistic relational model (PRM) as introduced in [44,43]. Please refer to [14] for a comprehensive treatment. No doubt such models are quite powerful to model various dependencies amongst entities,

though the subsequent inference always requires certain approximation. Their complexity and scalability are often a barrier for practical use. Indeed, Macskassy and Provost compared wvRN with PRM, and found that wvRN outperforms PRM on several relational data sets [29]. Given the extreme simplicity of wvRN and its outstanding performance, wvRN is adopted as a baseline in our experiments.

Many relational classifiers only capture the local dependency based on the Markov assumption. To capture the long-distance correlation, the latent group model [32] and the nonparametric infinite hidden relational model [51] are proposed. Both present generative models such that the links (and actor attributes) are generated based on actors' latent cluster membership. They share a similar spirit as SocioDim. But the model intricacy and high computational cost for inference hinders their direct application to huge networks. So Neville and Jensen in [32] propose to use a clustering algorithm to find the hard cluster membership of each actor first, and then fix the latent group variables for later inference. In social media, a network is often very noisy. Some nodes do not show a strong community membership and hard clustering might assign them randomly [19]. The resultant community structure can change drastically even with the removal of one single edge in the network. Our social dimensions are represented as continuous values. Each node is allowed to be involved in different dimensions in a flexible manner. It is also empirically verified that hard partition is not comparable to soft clustering in our experiment in Sect. 7.2. Another difference is that both the latent group model and nonparametric infinite hidden relational model are generative, while SocioDim allows the plug-in of any discriminative classifier. In conjunction with the discriminative power of SVM, SocioDim yields more accurate and stable performances.

### 8.2 Semi-supervised learning

Another related field is semi-supervised learning [54], originally proposed to address the label shortage problem by exploiting unlabeled data. One branch of semi-supervised learning is the graph-based approach [55,53]. Indeed, they share quite a similar assumption as collective inference. The performances of wvRN and Zhu's method [55] are nearly identical as reported in [29]. Considering that Zhu's method involves the computation of the inverse of a matrix of the same size as a given network, wvRN is used as the baseline in our experiments.

Some work [25,7] attempts to address semi-supervised learning with multiple labels by utilizing the relationship between different labels. The relationship can be obtained either from external experts or computed based on the labeled data. But its computational cost is prohibitive. We tried the method presented in [7], which constructs a graph between different labels and then calculates a label assignment so that it is smooth on both the instance graph and the label graph. It is required to solve a Sylvester equation [15] and direct implementation takes an extremely long time to find a solution, preventing us from reporting any comparative results.

On the other hand, some papers try to construct kernels based on graphs for SVM. Diffusion kernel [21] is a commonly used one. However, it requires full SVD of the graph Laplacian, which is not applicable for large-scale networks. Empirically, the

classification performance is sensitive to the diffusion parameter. Cross validation or some variant of kernel learning is required to select a proper diffusion kernel [48].

### 8.3 Community detection

Extracting latent social dimensions is related to community detection [41]. That has been an active field in social network analysis, and various methods have been proposed including stochastic block models [35,1], the latent space model [18,17], spectral clustering [27] and modularity maximization [33]. A comprehensive treatment is presented in [14]. In this work, spectral clustering is employed for SocioDim, but any other soft clustering methods should also serve the purpose.

Recently, Kumar et al. [22] found that real-world networks consist of a giant connected component with others being singletons and small-size connected components. Leskovec [23] studied the statistical properties of communities on the giant connected component and found a similar pattern. The optimal spectral cut always returns a community of 100 to 200 nodes, loosely connected (say, one or two edges) to the remaining network. A further comprehensive comparison of various community detection algorithms is reported in [24]. In these papers, most community detection methods focus on discrete binary cases, i.e., extracting one community from a network based on certain criterion. Whereas SocioDim employs soft clustering to extract social dimensions, and typically many more dimensions instead of just one or two are extracted. We believe a comprehensive comparison of different soft clustering approaches for the extraction of social dimensions and their scalability is an interesting line of future work.

## 9 Conclusions and future work

Social media provides a virtual social networking environment. The presence of partial label information and networking information allows us to build better classifiers. This work proposes a novel approach to dealing with heterogeneous connections prevalent in social media. To differentiate heterogeneous connections, we propose to extract latent social dimensions via soft clustering such as modularity maximization and spectral clustering. Based on the extracted social dimensions, a discriminative classifier like SVM can be constructed to determine which dimensions are informative for classification. Extensive experiments on social media data demonstrated that our proposed social dimension approach outperforms alternative relational learning methods based on collective inference, especially when the labeled data are few. It is noticed that some relational models perform poorly in social media data. This is due to the heterogeneity of connections and high irregularity of human interactions in social media. Our approach, by differentiating disparate types of connections among social actors and converting network information into conventional features, achieves effective learning for classification.

Many interesting directions can be explored along with the SocioDim framework.

– The SocioDim framework converts networks into features, thus enabling convenient integration of data in disparate formats. How is it compared with other

relational learning approaches that model network dependency and actor attributes? Is there any more effective method other than simple juxtaposition to integrate social dimensions and actor features?

– In this work, spectral clustering is employed to extract social dimensions. The resultant dimensions are dense, causing computational problems. On the contrary, a hard partition delivers sparse social dimensions, as each actor is associated with only one affiliation. But this constraint also limits its corresponding classification performance. It is imperative to marry the advantages of both soft clustering and hard partition. That is, each actor is allowed to participate in more than one affiliation, yet the corresponding social dimensions remain sparse. Some preliminary results aiming at extracting *sparse* social dimensions have been presented in [40]. On the other hand, Menon and Elkan [31] show that supervised and unsupervised extraction of social dimensions yield comparable results.
– Another line of research is parallel and distributed computing to handle evolving, large-scale networks. Luckily, our SocioDim consists of two well studied steps: spectral clustering and SVM learning. Both have been extended to distributed cases [16,6,8]. Thus SocioDim can be deployed to harness the power of parallel computing. In social media, networks are highly dynamic. Each day, new members join a social network, and new connections are established among existing members. It remains a challenge to achieve efficient updates in a parallel setting.
– In the current model,we do not employ the relationship between different labels. In order to handle this multi-label classification [47], a commonly-used one-vs-rest scheme [42] is used. In certain scenarios, the labels can present certain structures like a hierarchical taxonomy. It merits further research to employ both social networks and label networks for joint classification.

**Acknowledgements** This research is, in part, sponsored by the Air Force Office of Scientific Research grant FA95500810132. We thank BlogCatalog and Flickr for providing APIs. We acknowledge Xufei Wang and Munmun De Choudhury for their help with data collection. We also wish to acknowledge Subbarao Kambhampati and Pat Langley for their suggestions to improve this work. We thank the anonymous reviewers wholeheartedly for their expert opinions and constructive suggestions.

## References

1. Airodi EM, Blei D, Fienberg SE, Xing EP (2008) Mixed membership stochastic block models. J Mach Learn Res 9:1981–2014
2. Almack JC (1922) The influence of intelligence on the selection of associates. Sch Soc 16:529–530
3. Bott H (1928) Observation of play activities in a nursery school. Genet Psychol Monogr 4:44–88
4. Chakrabarti D, Faloutsos C (2006) Graph mining: laws, generators, and algorithms. ACM Comput Surv 38(1):2
5. Chakrabarti S, Dom B, Indyk P (1998) Enhanced hypertext categorization using hyperlinks. In: SIGMOD '98: proceedings of the 1998 ACM SIGMOD international conference on management of data. ACM, New York, NY, USA, pp 307–318
6. Chang E, Zhu K, Wang H, Bai H, Li J, Qiu Z, Cui H (2007) Psvm: parallelizing support vector machines on distributed computers. Adv Neural Inf Process Syst 20:1081–1088
7. Chen G, Wang F, Zhang C (2008) Semi-supervised multi-label learning by solving a sylvester equation. In: Proceedings of the SIAM international conference on data mining, Bethesda, MD, USA, pp 410–419

8. Chen W-Y, Song Y, Bai H, Lin C-J, Chang EY (2010) Parallel spectral clustering in distributed systems. IEEE Trans Pattern Anal Mach Intell 99
9. Fan R-E, Lin C-J (2007) A study on threshold selection for multi-label classication. Technical report, National Taiwan University
10. Fiore AT, Donath JS (2005) Homophily in online dating: when do you like someone like yourself?. In: CHI '05: CHI '05 extended abstracts on human factors in computing systems. ACM, New York, NY, USA, pp 1371–1374
11. Fortunato S, Barthelemy M (2007) Resolution limit in community detection. PNAS 104(1):36–41
12. Gallagher B, Tong H, Eliassi-Rad T, Faloutsos C (2008) Using ghost edges for classification in sparsely labeled networks. In: KDD '08: proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp 256–264
13. Geman S, Geman D (1990) Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, San Francisco, CA, USA, pp 452–472
14. Getoor L, Taskar B (Eds) (2007) Introduction to statistical relational learning. The MIT Press, London, England
15. Golub GH, Van Loan CF (1996) Matrix computations. 3. Johns Hopkins University Press, Baltimore
16. Graf H, Cosatto E, Bottou L, Dourdanovic I, Vapnik V (2005) Parallel support vector machines: the cascade svm. Adv Neural Inf Process Syst 17(521-528):2
17. Handcock MS, Raftery AE, Tantrum JM. (2007) Model-based clustering for social networks. J R Stat Soc A 127(2):301–354
18. Hoff PD, Raftery AE, Handcock MS (2002) Latent space approaches to social network analysis. J A Stat Assoc 97(460):1090–1098
19. Hopcroft J, Khan O, Kulis B, Selman B (2003) Natural communities in large linked networks. In: KDD '03: proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp 541–546
20. Jensen D, Neville J, Gallagher B (2004) Why collective inference improves relational classification. In: KDD '04: proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp 593–598
21. Kondor RI, Lafferty J (2002) Diffusion kernels on graphs and other discrete structures. In: ICML, New York, NY, USA
22. Kumar R, Novak J, Tomkins A (2006) Structure and evolution of online social networks. In: KDD '06: proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp 611–617
23. Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2008) Statistical properties of community structure in large social and information networks. In: WWW '08: proceeding of the 17th international conference on world wide web. ACM, New York, NY, USA, pp 695–704
24. Leskovec J, Lang KJ, Mahoney M (2010) Empirical comparison of algorithms for network community detection. In: WWW '10: proceedings of the 19th international conference on World wide web. ACM, New York, NY, USA, pp 631–640
25. Liu Y, Jin R, Yang L (2006) Semi-supervised multi-label learning by constrained non-negative matrix factorization. In: AAAI, Orlando, FL, USA
26. Lu Q, Getoor L (2003) Link-based classification. In: ICML: New York, NY, USA
27. Luxburg Uv (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416
28. Macskassy SA, Provost F (2003) A simple relational classifier. In: Proceedings of the multi-relational data mining workshop (MRDM) at the ninth ACM SIGKDD international conference on knowledge discovery and data mining, ACM Press, New York, NY, USA
29. Macskassy SA, Provost F (2007) Classification in networked data: a toolkit and a univariate case study. J Mach Learn Res 8:935–983
30. McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: homophily in social networks. Annu Rev Sociol 27:415–444
31. Menon AK, Elkan C (2010) Predicting labels for dyadic data. Data Min Knowl Discov 21(2):327–343
32. Neville J, Jensen D (2005) Leveraging relational autocorrelation with latent group models. In: MRDM '05: proceedings of the 4th international workshop on Multi-relational mining. ACM, New York, NY, USA, pp 49–55
33. Newman M (2006) Finding community structure in networks using the eigenvectors of matrices. Phys Rev E Stat Nonlin Soft Matter Phys 74(3)
34. Newman M (2006) Modularity and community structure in networks. PNAS 103(23):8577–8582

35. Nowicki K, Snijders TAB (2001) Estimation and prediction for stochastic blockstructures. J Am Stat Assoc 96(455):1077–1087
36. Sarkar P, Moore AW (2005) Dynamic social network analysis using latent space models. SIGKDD Explor Newsl 7(2):31–40
37. Sen P, Namata G, Bilgic M, Getoor L, Galligher B, Eliassi-Rad T (2008) Collective classification in network data. AI Mag 29(3):93
38. Shi J, Malik J (1997) Normalized cuts and image segmentation. In: CVPR '97: proceedings of the 1997 conference on computer vision and pattern recognition (CVPR '97). IEEE Computer Society, Washington, DC, USA, pp 731
39. Tang L, Liu H (2009a) Relational learning via latent social dimensions. In: KDD '09: proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp 817–826
40. Tang L, Liu H (2009b) Scalable learning of collective behavior based on sparse social dimensions. In: CIKM '09: proceeding of the 18th ACM conference on Information and knowledge management. ACM, New York, NY, USA, pp 1107–1116
41. Tang L, Liu H (1996) Community detection and mining in social media. Synthesis lectures on data mining and knowledge discovery. Morgan and Claypool Publishers, USA
42. Tang L, Rajan S, Narayanan VK (2009) Large scale multi-label classification via metalabeler. In: WWW '09: proceedings of the 18th international conference on world wide web. New York, NY, USA, pp 211–220
43. Taskar B, Abbeel P, Koller D (2002) Discriminative probabilistic models for relational data. In: UAI, Edmonton, Canada, pp 485–492
44. Taskar B, Segal E, Koller D (2001) Probabilistic classification and clustering in relational data. In: IJCAI'01: proceedings of the 17th international joint conference on artificial intelligence. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, pp 870–876
45. Thelwall M (2009) Homophily in myspace. J Am Soc Inf Sci Technol 60(2):219–231
46. Travers J, Milgram S (1969) An experimental study of the small world problem. Sociometry 32(4):425–443
47. Tsoumakas G, Katakis I (2007) Multi label classification: an overview. Int J Data Wareh Min 3(3):1–13
48. Tsuda K, Noble WS (2004) Learning kernels from biological networks by maximizing entropy. Bioinformatics 20:326–333
49. Wasserman S, Faust K (1994) Social network analysis: methods and applications. Cambridge University Press, Cambridge
50. Wellman B (1926) The school child's choice of companions. J Edu Res 14:126–132
51. Xu Z, Tresp V, Yu S, Yu K (2008) Nonparametric relational learning for social network analysis. In: KDD'2008 workshop on social network mining and analysis, Las Vegas, NV, USA
52. Zha H, He X, Ding CHQ, Gu M, Simon HD. (2001) Spectral relaxation for k-means clustering. In: NIPS, Vancouver, Canada, pp 1057–1064
53. Zhou D, Bousquet O, Lal T, Weston J, Scholkopf B (2004) Learning with local and global consistency. In: Advances in neural information processing systems 16: proceedings of the 2003 conference. Bradford Book, Cambridge, pp 321
54. Zhu X (2006) Semi-supervised learning literature survey. MIT Press, Cambridge, USA
55. Zhu X, Ghahramani Z, Lafferty J (2003) Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, New York, NY, USA