Integrating Social Media Data for Community Detection

Jiliang Tang, Xufei Wang and Huan Liu

Computer Science & Engineering, Arizona State University, Tempe, AZ 85281
{Jiliang.Tang, Xufei Wang, Huan.Liu}@asu.edu

Abstract. Community detection is an unsupervised learning task that discovers groups such that group members share more similarities or interact more frequently among themselves than with people outside groups. In social media, link information can reveal heterogeneous relationships of various strengths, but often can be noisy. Since different sources of data in social media can provide complementary information, e.g., bookmarking and tagging data indicates user interests, frequency of commenting suggests the strength of ties, etc., we propose to integrate social media data of multiple types for improving the performance of community detection. We present a joint optimization framework to integrate multiple data sources for community detection. Empirical evaluation on both synthetic data and real-world social media data shows significant performance improvement of the proposed approach. This work elaborates the need for and challenges of multi-source integration of heterogeneous data types, and provides a principled way of *multi-source* community detection.

Keywords: Community Detection, Multi-source Integration, Social Media Data

1 Introduction

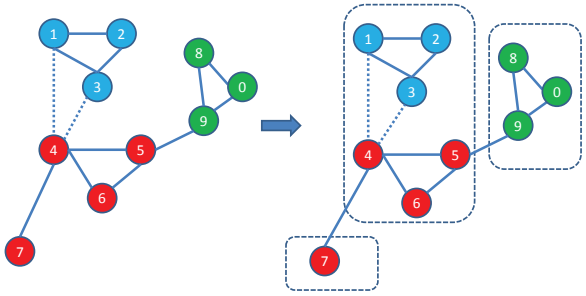
Social media is quickly becoming an integral part of our life. Facebook, one of the most popular social media websites, has more than 500 million users and more than 30 billion pieces of content shared each month[1]. YouTube attracts 2 billion video views per day[2]. Social media users can have various online social activities, e.g., forming connections, updating their status, and sharing their interested stories and movies. The pervasive use of social media offers research opportunities of group behavior. One fundamental problem is to identify groups among individuals if the group information is not explicitly available [1]. A group (or a community) can be considered as a set of users who interact more frequently or share more similarities among themselves than those outside the group. This topic has many applications such as relational learning, behavior modeling and

[1] http://www.facebook.com/press/info.php?statistics
[2] http://mashable.com/2010/05/17/youtube-2-billion-views/

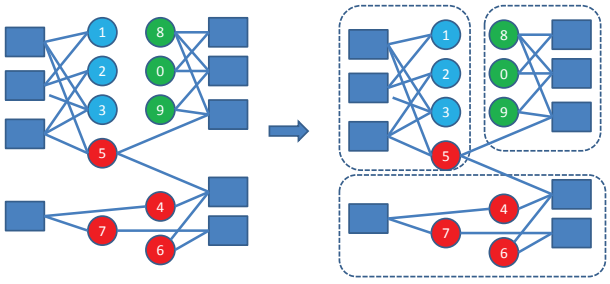
prediction [19], linked feature selection [17] [18], visualization, and group formation analysis [1].

Different from connections formed by people in the physical world, users of social media have greater freedom to connect to a greater number of users in various ways and for disparate reasons. In online social networks, the low cost of link information can lead to networks with heterogeneous relationship strengths (e.g., acquaintances and best friends mixed together) [24]. Hence, noise and casual links are prevalent in social media, posing challenges to the link-based community detection algorithms [12, 13, 4]. In addition to link information that indicates interactions, there are other sources of information that indirectly represent connections of different kinds in social media.

User profiles that describe their locations, interests, education background, etc. provide useful information differing from links. For example, Scellato et al. find that clusters of friends are often geographically close [14]. There are other activities that produce information about interactions: bookmarking data implies user interests, frequency data of commenting on their friends homepage suggests the strength of connections. These types of information can also be useful in finding a community structure in social media.



(a) Link information



(b) Tag information

Fig. 1. Community Detection based on Single Source

Figure 1 shows a toy example with two sources, i.e., link and tag information. Figure 1(a) shows the communities identified by Modularity Maximization [12] based on link information. The weak links, i.e., $(1, 4)$ and $(3, 4)$, make link-based algorithms ineffective. Figure 1(b) shows the results of k-means on the tagging information, which similarly cannot reveal the real community structures. Each source contains noisy but complementary information with other sources. For example, the links $(1, 4)$ and $(3, 4)$ are weak since they don't share any tagging information.

With these multiple and complementary sources, we ask 1) *could we improve the performance of community detection by combining multiple types of data*? And 2) *how can we integrate data of heterogeneous types effectively*? In this paper, we propose a joint optimization framework to integrate multiple data sources to discover communities in social media. Experimental results on synthetic data and real-world social media data show that the performance of community detection is significantly improved through integrating multiple sources. Our main contributions are summarized below,

- Identifying the need for integrating multiple sources for community detection in social media,
- Proposing a novel framework to integrate multiple sources for community detection and link strength prediction, and
- Presenting interesting findings such as integrating more data sources does not necessarily bring about better performance through experiment design in real-world social media datasets.

The rest of this paper is organized as follows. The related work is summarized in Section 2. The problem of multi-source integration is formally defined in Section 3. An integrating framework is introduced in Section 4, followed by empirical evaluation in Section 5 with detailed discussion. The conclusion and future work is presented in Section 6.

2 Related Work

Community detection algorithms can be divided into three generic categories based on types of data sources used: link-based, link and content-based and interaction-based algorithms. Next we review each category separately.

2.1 Community Detection based on Links

The study of link-based methods has a long history. It is closely related to graph partitioning in graph theory. For example, one approach to graph partitioning is to find disjoint subgraphs such that cuts are minimized. Since the graph partition problem is NP-hard, it is relaxed to spectral clustering for practical reasons [8]. The concept of modularity to measure the strength of a community structure is proposed in [12]. Since maximizing modularity is NP-hard, a relaxation to spectral clustering is proposed in [23].

A social media user can have multiple interactions and interests, which suggests that community structures often overlap. CFinder [13] is a local algorithm that enumerates all k-cliques and then combines any two cliques if they share $k-1$ nodes. It is computationally expensive. Evans et al. [5] propose to partition links of a line graph to uncover the overlapping community structure. A line graph can be constructed from the original graph, i.e., each vertex in the line graph corresponds to an original edge and a link in the line graph represents the adjacency between two edges in the original graph. However, this algorithm is memory inefficient, so it cannot be applied to large social networks. EdgeCluster [19] takes an edge centric view of the graph: edges are treated as instances and nodes as features, and can find highly overlapping communities. Some other ways to obtain overlapping communities include soft clustering [11] and probabilistic models [4].

2.2 Combining Link and Content Information

Generative models such as Latent Dirichlet Allocation (LDA) [2] can be used to model links and content via a shared set of community memberships. Erosheva et al. integrates abstracts and references of scientific papers under the LDA framework in document clustering applications [4]. They assume there is a fixed number of categories, each is viewed as a multinomial distribution on words or links. One problem with the generative models is that they are susceptible to irrelevant keywords. [25] proposes a probabilistic model to combine link and content information in community detection with improvement. They first build a conditional model which estimates the probability of connecting node i to node j. Then the membership of a node to a community is modeled on content information and the two models are unified via community memberships. [7] proposes the Topic-Link LDA model that co-clusters documents or blogs and authors. There are two problems with above models: 1) they are designed to model author-emails and author-scientific papers with specific assumptions; and 2) they are not designed to integrate more than two sources as needed for social media.

2.3 Utilizing Interactions beyond Links

Social media users have various types of interactions. Since interactions between users imply their closeness, information of interactions can be important in uncovering groups in social media. A co-clustering framework is proposed in [22] to leverage users' tagging behavior in community detection. It shows that more accurate community structures can be obtained by leveraging the tag information. MetaGraph Factorization (MetaFac) is presented in [6] to extract community structures from various interactions. In [20], the authors propose methods of integrating information of heterogeneous interactions for community detection. Our proposed community detection approach differs from these methods in explicitly integrating tie strength prediction.

3 Problem Statement

Before building the mathematical model, we would like to establish the notations to be used. Following the standard notations, scalars are denoted by low-case letters ($a, b, \ldots; \alpha, \beta, \ldots$), vectors are written as low-case bold letters ($\mathbf{a}, \mathbf{b}, \ldots$) and matrices correspond to bold-face upper-case letters ($\mathbf{A}, \mathbf{B}, \ldots$). $\mathbf{A}(i,j)$ is the entry at the i^{th} row and j^{th} column of the matrix $\mathbf{A}$, $\mathbf{A}(i,:)$ is the i^{th} row of $\mathbf{A}$ and $\mathbf{A}(:,j)$ is the j^{th} column of $\mathbf{A}$ etc. We use $\mathbf{0}_{i\times j}$ to represent a $i \times j$ zero matrix, $\mathbf{I}_k$ to represent a $k \times k$ unit matrix, and $\mathbf{1}_{i\times j}$ represents a $i \times j$ all one matrix. Let $\mathbf{u} = \{u_1, u_2, ..., u_n\}$ be the user set where n is the number of users, and $\mathbf{c} = \{c_1, c_2, ..., c_K\}$ where K communities are to be identified.

Definition 1. Link Matrix $\mathbf{Y}_0 \in \mathbb{R}^{n\times n}$ is the adjacency matrix whose entries represent the connectivity between two users, i.e., $\mathbf{Y}_0(i,j) = 1$ if u_j has a link to u_i, otherwise $\mathbf{Y}_0(i,j) = 0$.

In social networks, the degree distribution typically follows a power law distribution, i.e., most people have a few friends, while few people have extremely many friends. It suggests that the link matrix $\mathbf{Y}_0$ should be sparse. Actually the nonzero entities in $\mathbf{Y}_0$, i.e., the total number of edges or arcs, is normally linear, rather than squared, with respect to the number of nodes in a network. This can be verified following the properties of a power law distribution.

$$p(x) = (1-\alpha)x^{-\alpha}, x \geq x_{min} > 0 \tag{1}$$

where α is the exponent which often falls between 2 and 3 [10], x is the nodal degree. The expected number of edges is

$$E[\mu^m] = \frac{n}{2} \cdot \frac{\alpha - 1}{\alpha - 2} \cdot x_{min} \tag{2}$$

Definition 2. Affiliation Matrix is denoted by $\mathbf{H} \in \mathbb{R}^{K\times n}$. The j^{th} column of $\mathbf{H}$, $\mathbf{H}(:,j)$, represents the memberships of u_j with respect to K communities so Affiliation Matrix should be *non-negative*.

The diversity of people's interests suggests that people might belong to more than one community. Since the number of communities one belongs to can be upper bounded by his nodal degree, $\mathbf{H}$ should be *sparse*.

Definition 3. Source Matrix is denoted by $\mathbf{Y}_i \in \mathbb{R}^{m_i\times n}(1 \leq i \leq m)$, where m_i is the number of features related to the source i and m is the number of additional sources. If user u_i subscribes to a feature j (e.g., u_i uses the j^{th} tag, or comments on u_j's post), then the corresponding entry is the frequency u_i subscribes to the feature j, otherwise 0.

Source Matrix should also be sparse. For example, one person u_i usually comments on a small part of persons in $\mathbf{u}$. The entities in Source Matrix are not limited to $\{0, 1\}$ since they represent frequencies. The set of m sources is represented by $\mathcal{S} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_m\}$.

With the notations and definitions defined, our community detection problem of integrating multiple sources can be stated as follows:

Given Link Matrix $\mathbf{Y}_0$*, a set of Source Matrices* $\mathcal{S}$*, and the number of communities* K*, compute a sparse Affiliation Matrix* $\mathbf{H}$ *by leveraging different types of data in social media.*

4 A Joint Optimization Framework for Integrating Multiple Data Types

In social media, one person can have multiple activities (e.g., tagging, commenting, etc.). Links contain the static relation between users. It is about one aspect of a user and can be supplemented with additional types of information that reflect interactions of corresponding aspects. For example, tagging data implies personal interests; frequency data of commenting suggests the strength of a connection and so on. Taking into account of different data sources, we investigate how to integrate data of different types in solving the problem of community detection. In this section we begin with a formulation that integrates two data sources before generalizing it to handle multiple data sources.

4.1 Integrating Two Sources

The formation of communities in social media can be explained by the Homophily effect [9]: compared with people outside of the group, users within a group tend to share more commonalities such as forming more connections, interacting more frequently, using similar tags, having similar attitudes, etc. Thus, it is reasonable to assume that people have similar community affiliations in different sources.

Given the link matrix $\mathbf{Y}_0$ and another type of data $\mathbf{Y}_i$, integrating two data sources can be formulated as a joint optimization problem through matrix factorization techniques as follows (2JointMF),

$$\begin{aligned}
\min_{\mathbf{W}_0,\mathbf{W}_i,\mathbf{H}} & \|\mathbf{Y}_0 - \mathbf{W}_0\mathbf{H}\|_F^2 + \|\mathbf{Y}_i - \mathbf{W}_i\mathbf{H}\|_F^2 \\
& + \lambda \sum_{j=1}^{n} \|\mathbf{H}(:,j)\|_1^2 + \eta(\|\mathbf{W}_0\|_F^2 + \|\mathbf{W}_i\|_F^2), \\
\mathbf{s.t.} \quad & \mathbf{R} = \mathbf{W_0}\mathbf{H} \leq \mathbf{1}_{n\times n} \\
& \mathbf{R} = \mathbf{W_0}\mathbf{H} \geq \mathbf{0}_{n\times n} \\
& \mathbf{H} \geq \mathbf{0}_{K\times n}
\end{aligned} \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, $\mathbf{W}_0 \in \mathbb{R}^{n\times K}$ and $\mathbf{W}_i \in \mathbb{R}^{m_i\times K}$. The parameter η controls the size of the elements in $\mathbf{W}_0$ and $\mathbf{W}_i$. $\mathbf{H}$ is the Affiliation Matrix, which indicates the memberships of users w.r.t K communities. From the definition of Affiliation Matrix, $\mathbf{H}$ should be non-negative. L_1-norm regularization is widely used for the purpose of achieving sparsity of the solution [21]. In our formulation, L_1-norm regularization is applied to each

column of affiliation matrix $\mathbf{H}$ based on the observation that one user is usually involved in a small number of communities. λ balances the trade-off between the sparseness of $\mathbf{H}$ and the accuracy of approximation.

The low cost of link information can lead to networks with heterogeneous relationship strengths [24]. Weak links in online social networks might make link-based community detection algorithms ineffective, as shown in Figure 1(a), and users' multiple interactions indicate the link strengths between users [24] [16]. We use $\mathbf{R}$ to reconstruct the original link matrix and represent strengths of refined relationships between users by considering multiple sources.

Unfortunately, the formulation in Eq. (3) is not concave due to the coupling of $\mathbf{W}_0$, $\mathbf{W}_i$ and $\mathbf{H}$. Thus it is hard to find a global solution for the joint optimization problem. Actually, if we fix 2 components such as $\mathbf{W}_0$ and $\mathbf{W}_i$, the resulting optimization problem for the left 1 component, $\mathbf{H}$, is concave, therefore through computing $\mathbf{W}_0$, $\mathbf{W}_i$ and $\mathbf{H}$ alternatively, we can find a local minimal solution for Eq. (3).

For computing $\mathbf{H}$, we fix components $\mathbf{W}_0$ and $\mathbf{W}_i$ and then develop the following theorem:

Theorem 1. *When components $\mathbf{W}_0$ and $\mathbf{W}_i$ are fixed, the formulation to optimize $\mathbf{H}$ in Eq (3) is equivalent to the following constrained minimization problem:*

$$\begin{aligned} &\min_{\mathbf{H}} \|\mathbf{A} - \mathbf{B}\mathbf{H}\|_F^2 \\ &\mathbf{s.t.} \quad \mathbf{C}\mathbf{H} \leq \mathbf{D} \end{aligned} \tag{4}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are defined as follows:

$$\begin{aligned} \mathbf{A} &= (\mathbf{Y}_0^\top, \mathbf{Y}_i^\top, \mathbf{0}_{n\times 1})^\top \\ \mathbf{B} &= (\mathbf{W}_0^\top, \mathbf{W}_i^\top, \sqrt{\lambda}\mathbf{1}_{K\times 1})^\top \\ \mathbf{C} &= (-\mathbf{I}_K, \mathbf{W}_0^\top, -\mathbf{W}_0^\top)^\top \\ \mathbf{D} &= (-\mathbf{0}_{n\times K}, \mathbf{1}_{n\times n}^\top, -\mathbf{0}_{n\times n})^\top \end{aligned} \tag{5}$$

Proof. It suffices to show the objective functions and constraints in Eq (3) and Eq (4) are correspondingly equivalent by constructing matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$.

When $\mathbf{W}_0$ and $\mathbf{W}_i$ are fixed, The last regularization, $\eta(\|\mathbf{W}_0\|_F^2 + \|\mathbf{W}_i\|_F^2)$, in Eq (3) is constant. Due to the nonnegative constraint on $\mathbf{H}$, $\sum_{j=1}^{n} \|\mathbf{H}(:,j)\|_1^2 = \|\mathbf{1}_{1\times K} H\|_2^2$. Then the objective function in Eq (3) can be converted to:

$$\begin{aligned} &\|\mathbf{Y}_0 - \mathbf{W}_0\mathbf{H}\|_F^2 + \|\mathbf{Y}_i - \mathbf{W}_i\mathbf{H}\|_F^2 + \lambda\|\mathbf{e}_{1\times K} H\|_2^2 \\ &= \|(\mathbf{Y}_0^\top, \mathbf{Y}_i^\top, \mathbf{0}_{n\times 1})^\top - (\mathbf{W}_0^\top, \mathbf{W}_i^\top, \sqrt{\lambda}\mathbf{1}_{K\times 1})^\top H\|_F^2 \\ &= \|\mathbf{A} - \mathbf{B}\mathbf{H}\|_F^2 \end{aligned} \tag{6}$$

It is easy to verify that the constraints in Eq (3) can be converted into:

$$\begin{aligned}&(-\mathbf{I}_K, \mathbf{W}_0^\top, -\mathbf{W}_0^\top)^\top \mathbf{H} \leq (-\mathbf{0}_{n\times K}, \mathbf{1}_{n\times n}, -\mathbf{0}_{n\times n})^\top \\ &= \mathbf{CH} \leq \mathbf{D}\end{aligned} \tag{7}$$

which completes the proof.

For computing the component $\mathbf{W}_0$, we have the following theorem:

Theorem 2. *When components $\mathbf{W}_i$ and $\mathbf{H}$ are fixed, the formulation to optimize $\mathbf{W}_0$ in Eq (3) is equivalent to the following constrained minimization problem:*

$$\begin{aligned}&\min_{\mathbf{W}_0} \|\mathbf{A} - \mathbf{B}\mathbf{W}_0^\top\|_F^2 \\ &\mathbf{s.t.} \quad \mathbf{C}\mathbf{W}_0^\top \leq \mathbf{D}\end{aligned} \tag{8}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are defined as follows:

$$\begin{aligned}\mathbf{A} &= (\mathbf{Y}_0, \mathbf{0}_{K\times n})^\top \\ \mathbf{B} &= (\mathbf{H}, \sqrt{\eta}\mathbf{I}_K)^\top \\ \mathbf{C} &= (\mathbf{H}, -\mathbf{H})^\top \\ \mathbf{D} &= (\mathbf{1}_{n\times n}, -\mathbf{0}_{n\times n})^\top\end{aligned} \tag{9}$$

Proof. When $\mathbf{W}_i$ and $\mathbf{H}$ are fixed, $\|\mathbf{Y}_i - \mathbf{W}_i\mathbf{H}\|_F^2$, $\lambda\sum_{j=1}^n \|\mathbf{H}(:,j)\|_1^2$, and $\eta\|\mathbf{W}_i\|_F^2$ are constants. Then the objective function for $\mathbf{W}_0$ in Eq (3) is:

$$\begin{aligned}&\|\mathbf{Y_0} - \mathbf{W_0}\mathbf{H}\|_F^2 + \eta\|\mathbf{W}_0\|_F^2 \\ &= \|(\mathbf{Y}_0, \mathbf{0}_{K\times n})^\top - (\mathbf{H}, \sqrt{\eta}\mathbf{I}_k)^\top \mathbf{W}_0^\top\|_F^2 \\ &= \|\mathbf{A} - \mathbf{B}\mathbf{W}_0^\top\|_F^2\end{aligned} \tag{10}$$

The proof process for the equivalence of constraints is similar to that of Theorem 1.

From Theorem 2, we can see that given $\mathbf{H}$, the calculation of $\mathbf{W}_0$ is independent on $\mathbf{W}_i$ and $\mathbf{Y}_i$.

When $\mathbf{W}_0$ and $\mathbf{H}$ are fixed, since the three constraints in Eq (3) are independent of $\mathbf{W}_i$, the optimization problem for $\mathbf{W}_i$ is a typical least square problem:

$$\mathbf{W}_i = \mathbf{Y}_i\mathbf{H}^\top(\mathbf{H}\mathbf{H}^\top + \eta\mathbf{I}_K)^{-1} \tag{11}$$

Algorithm for Integrating Two Sources Through Theorems 1 and 2, we notice that the optimization problems for computing $\mathbf{H}$ and $\mathbf{W}_0$ are equivalent in solving the following optimization problem:

$$\begin{aligned}&\min_{\mathbf{X}} \|\mathbf{A} - \mathbf{B}\mathbf{X}\|_F^2 \\ &\mathbf{s.t.} \quad \mathbf{C}\mathbf{X} \leq \mathbf{D}\end{aligned} \tag{12}$$

this problem is indeed the collection of several linear constrained least square problems.

$$\begin{aligned} &\min_{\mathbf{X}(:,j)} \|\mathbf{A}(:,j) - \mathbf{B}\mathbf{X}(:,j)\|_F^2 \\ &\mathbf{s.t.} \quad \mathbf{C}\mathbf{X}(:,j) \leq \mathbf{D}(:,j) \end{aligned} \tag{13}$$

In our implementation, we use the active-set method to solve this linear constrained least square problem and we assume that the function to solve Eq. (13) is named *iplsqlin*, has four arguments, and outputs the optimal x, i.e., $x = iplsqlin(\mathbf{a}, \mathbf{B}, \mathbf{C}, \mathbf{d})$. Algorithm 1 shows how to update $\mathbf{H}$. To get each column of $\mathbf{H}$, we have to solve a linear constrained least square problem.

Algorithm 1 Update-$\mathbf{H}$

Input: The Link Matrix $\mathbf{Y}_0$, Source Matrix Y_i, the fixed components $\mathbf{W}_0$ and $\mathbf{W}_i$, and λ.
Output: $\mathbf{H}$.

1: Construct $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ according to Eq. (5)
2: **for** $i = 1 \rightarrow n$ **do**
3: $\quad \mathbf{H}(:,i) \leftarrow iplsqlin(\mathbf{A}(:,i), \mathbf{B}, \mathbf{C}, \mathbf{D}(:,i))$
4: **end for**

Similar as the algorithm for updating $\mathbf{H}$, the algorithm for updating $\mathbf{W}_0$ is shown in Algorithm 2. The input of Algorithm 2 is independent on $\mathbf{Y}_i$ and $\mathbf{W}_i$.

Algorithm 2 Update-$\mathbf{W}_0$

Input: The Link Matrix $\mathbf{Y}_0$, the fixed components $\mathbf{H}$ and η.
Output: $\mathbf{W}_0$

1: Construct $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ according to Eq. (9)
2: **for** $i = 1 \rightarrow n$ **do**
3: $\quad \mathbf{W}_0^\top(:,i) \leftarrow iplsqlin(\mathbf{A}(:,i), \mathbf{B}, \mathbf{C}, \mathbf{D}(:,i))$
4: **end for**

Based on Update-$\mathbf{H}$ and Update-$\mathbf{W}_0$, we have Algorithm 3 to solve the problem in Eq. (3). Note that the solution of Eq. (3) is not unique. Given a solution of $\{\mathbf{W}_0, \mathbf{W}_i, \mathbf{H}\}$, $\{\mathbf{W}_0\mathbf{D}, \mathbf{W}_i\mathbf{D}, \mathbf{D}^{-1}\mathbf{H}\}$ is also the solution for Eq. (3), where $\mathbf{D}$ is a diagonal matrix with positive elements. We seek a unique solution by

applying a normalization to each column of $\mathbf{H}$.

$$\mathbf{W}_0(j,k) = \mathbf{W}_0(j,k)\sqrt{\sum_j \mathbf{H}^2(j,k)}$$
$$\mathbf{W}_i(j,k) = \mathbf{W}_i(j,k)\sqrt{\sum_j \mathbf{H}^2(j,k)}$$
$$\mathbf{H}(j,k) = \frac{\mathbf{H}(j,k)}{\sqrt{\sum_j \mathbf{H}^2(j,k)}} \tag{14}$$

Algorithm 3 TwoSources

Input: The Link Matrix $\mathbf{Y}_0$, Source Matrix $\mathbf{Y}_i$, λ and η
Output: $\mathbf{H}$ and $\mathbf{W}_0$

1: Initialize $\mathbf{H}$, $\mathbf{W}_0$ and $\mathbf{W}_i$
2: **while** Not convergent **do**
3: Update: $\mathbf{W}_i \leftarrow \mathbf{Y}_i\mathbf{H}^\top(\mathbf{H}\mathbf{H}^\top + \eta\mathbf{I}_K)^{-1}$
4: Update: $\mathbf{H} \leftarrow$ Update-$\mathbf{H}(\mathbf{Y}_0, \mathbf{Y}_i, \mathbf{W}_0, \mathbf{W}_i, \lambda)$
5: Update: $\mathbf{W}_0 \leftarrow$ Update-$\mathbf{W}_0(\mathbf{Y}_0, \mathbf{H}, \eta)$
6: **end while**
7: Normalize $\mathbf{H}$, $\mathbf{W}_0$ and $\mathbf{W}_i$ by Eq (14)

In Algorithm 3, after some initialization, we alternatively use Eq (11), Update-$\mathbf{H}$ and Update-$\mathbf{W}_0$ to update $\mathbf{W}_i$, $\mathbf{H}$ and $\mathbf{W}_0$ by fixing two of them. This alternative process will be iterated until convergence.

Illustration based on a Toy Example To further illustrate the advantages of the proposed framework for community detection, let us consider the example shown in Figure 1. There are two sources, i.e., link information and tag information. We run our two source method, i.e., Algorithm 3, to integrate link and tag information for community detection. The Affiliation Matrix $\mathbf{H}$ is shown as follows:

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & 1 & 1 & .29 & .11 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .71 & .74 & 1 & 1 & 0 & .18 \\ 1 & 0 & 0 & 0 & 0 & .15 & 0 & 0 & 1 & .82 \end{pmatrix}$$

The first observation is that the solution is sparse and more than half of entities are exactly zeros. After normalization, $\mathbf{H}(i,j)$ is the probability of u_j belonging to c_i. We can see that the result is very consistent with the real memberships of users.

We also use $\mathbf{R}$ to reconstruct the original link matrix $\mathbf{Y}_0$. According to our framework, each column of $\mathbf{R}$, $\mathbf{R}(:,j)$, represents the strengths of relationships

between u_j and other users. We examine $\mathbf{R}$ and find that the strengths of links $(4,1)$ and $(4,3)$ are much weaker than those of links $(4,5)$, $(4,6)$ and $(4,7)$. We run Modularity Maximization on $\mathbf{R}$ and the result is shown in Figure 2, which is consistent with the ground truth, demonstrating the advantages of our proposed framework.

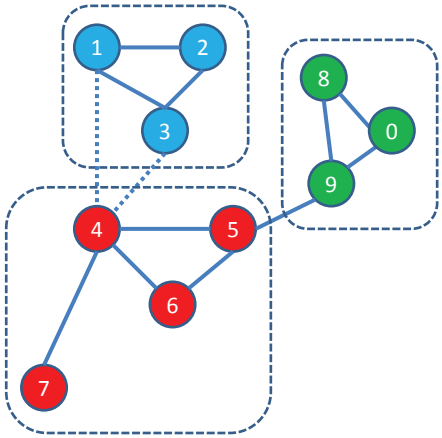


Fig. 2. Community Detection based on Two Sources

5 Integrating Multiple Sources

The development of the two-source solution paves the way for a multi-source solution. Given the link matrix $\mathbf{Y}_0$ and a set of m data sources $\mathcal{S} = \{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_m\}$, the optimization problem for integrating multiple sources can be formalized as follows (mJointMF),

$$\begin{aligned} \min_{\mathbf{W_0}, \mathbf{W}_i, \mathbf{H}} & \|\mathbf{Y}_0 - \mathbf{W}_0\mathbf{H}\|_F^2 + \sum_{i=1}^{m} \|\mathbf{Y}_i - \mathbf{W}_i\mathbf{H}\|_F^2 \\ & + \lambda \sum_{j=1}^{n} \|\mathbf{H}(:,j)\|_1^2 + \eta \sum_{k=0}^{m} \|\mathbf{W}_k\|_F^2, \\ \textbf{s.t.} \quad & \mathbf{W}_0\mathbf{H} \leq 1 \\ & \mathbf{W}_0\mathbf{H} \geq 0 \\ & \mathbf{H} \geq 0 \end{aligned} \tag{15}$$

The following theorem shows the connection between the two source integration method and the multi-source integration method for community detection.

Theorem 3. *The optimization problem for multi-source integration is equivalent to the following minimization problem,*

$$\begin{aligned}
&\min_{\mathbf{W}_0,\mathbf{W},\mathbf{H}} \|\mathbf{Y_0} - \mathbf{W}_0\mathbf{H}\|_F^2 + \|\mathbf{C} - \mathbf{W}\mathbf{H}\|_F^2 \\
&\qquad + \lambda \sum_{j=1}^{n} \|\mathbf{H}(:,j)\|_1^2 + \eta(\|\mathbf{W}_0\|_F^2 + \|\mathbf{W}\|_F^2), \\
&\mathbf{s.t.} \quad \mathbf{W}_0\mathbf{H} \leq 1 \\
&\qquad \mathbf{W}_0\mathbf{H} \geq 0 \\
&\qquad \mathbf{H} \geq 0
\end{aligned} \tag{16}$$

where $\mathbf{W}$ *and* $\mathbf{C}$ *are defined as follows:*

$$\begin{aligned}
\mathbf{C} &= (\mathbf{Y}_1^\top, \mathbf{Y}_2^\top, \ldots, \mathbf{Y}_m^\top)^\top \\
\mathbf{W} &= (\mathbf{W}_1^\top, \mathbf{W}_2^\top, \ldots, \mathbf{W}_m^\top)^\top
\end{aligned}$$

Proof. Comparing Eq (15) with Eq (16), there are two differences. Therefore, it suffices to show that they are correspondingly equivalent. The following formulation suggests that the first difference is equivalent.

$$\begin{aligned}
&\sum_{i=1}^{m} \|\mathbf{Y}_i - \mathbf{W}_i\mathbf{H}\|_F^2 \\
&= \|(\mathbf{Y}_1^\top, \mathbf{Y}_2^\top, \ldots, \mathbf{Y}_m^\top)^\top - (\mathbf{W}_1^\top, \mathbf{W}_2^\top, \ldots, \mathbf{W}_m^\top)^\top\mathbf{H}\|_F^2 \\
&= \|\mathbf{C} - \mathbf{W}\mathbf{H}\|_F^2
\end{aligned} \tag{17}$$

The equivalence of the second difference is shown below:

$$\begin{aligned}
&\eta \sum_{k=0}^{m} \|\mathbf{W}_k\|_F^2 \\
&= \eta(\|\mathbf{W}_0\|_F^2 + \|(\mathbf{W}_1^\top, \mathbf{W}_2^\top, \ldots, \mathbf{W}_m^\top)^\top\|_F^2) \\
&= \eta(\|\mathbf{W}_0\|_F^2 + \|\mathbf{W}\|_F^2)
\end{aligned} \tag{18}$$

which completes the proof.

Theorem 3 implies that the optimization problem for integrating multiple sources is equivalent to that for integrating two sources. The significance of this theorem is twofold: first, it provides a way to solve the multiple sources integration problem using two sources integration, which is shown in Algorithm 4; and second, it provides an intuitive explanation for how the data of multiple sources are integrated: *sources in* $\mathcal{S}$ *firstly stack together and then integrate with the link source.*

Algorithm 4 MultipleSource

Input: The Link Matrix $\mathbf{Y}_0$, the source set $\mathcal{S}$, λ and η.
Output: $\mathbf{H}$ and $\mathbf{W}_0$

1: Construct $\mathbf{C}$ according to Eq. (9)
2: Set $(\mathbf{H}, \mathbf{W}_0) = \text{TwoSources}(\mathbf{Y}_0, \mathbf{C}, \lambda, \eta)$
3: Normalize $\mathbf{H}$ and $\mathbf{W}_0$ by Eq (14)

6 Experimental Evaluation

To verify the effectiveness of our proposed method, we conduct experiments on both synthetic data and real-world social media data.

6.1 Synthetic Data

Since the ground truth is usually unavailable for real-world social media data, we resort to synthetic data to show if the proposed framework can achieve the design goals. The synthetic data consists of two types of information: link information and tag information. Parameters in generating the synthetic data include the number of users, n, the number of tags, t, the number of communities, K, link (tag) density within and between communities, ρ_w, ρ_b, and the ratio of noise links (tags) ρ_n. To generate the ground truth, users and tags are split evenly into each community, and according to link (tag) density within communities ρ_w, we randomly generate links between users (users and tags) in the same community. While relying on link (tag) density between communities ρ_b, we randomly create links between users (users and tags) from different communities.

To simulate noise and complementary information in sources, we design the following procedure,

- Randomly assign communities into two groups with equal size, i.e., g_1 and g_2. Let u_1 and t_1 be the set of users and tags in g_1, respectively, while u_2 and t_2 are the set of users and tags in g_2.
- Randomly add links between u_1 according to the noise ratio ρ_n, for link information. For tag information, we randomly add links for u_2.

Through the above process, with link information for u_2 being fixed, we add noisy tags to u_2, and with tag information for u_1 being fixed, noisy links are added to u_1. Therefore, link and tagging information generated above are noisy but complementary with each other.

In this experiment, we generate a set of datasets with parameters: $n = 1000$, $t = 1000$, $k = 20$, $\rho_w = 0.8$, $\rho_b = 0.1$ and varying ρ_n from 0 to 1 with step 0.1. Five baseline methods are used: LDA-Link(LL) [4], PCL-Link(PL) [25], EdgeCluster(EC) [19] and Modularity Maximization(Modu) [12] with only link information; and Tag-CoClustering(TC) [22] using only tagging information. All parameters in comparing methods are determined by cross validation. Normalized mutual information is adopted to evaluate the community quality. The average NMI performance w.r.t the noise ratio ρ_n are shown in Figure 3

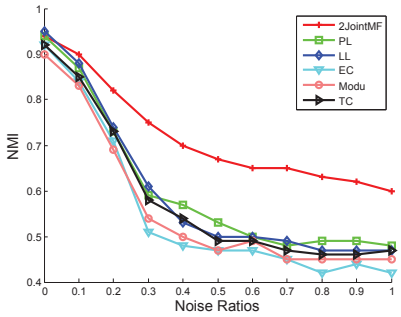


Fig. 3. NMI Performance w.r.t Noise Ratios

The first observation is that with the noise ratios increasing, all performance reduces dramatically, especially for link-based algorithms. This supports our assumption that noise links or weak links in online social networks can make link-based algorithms ineffective. By integrating two sources, our algorithm consistently outperforms algorithms with a single source.

6.2 Social Media Data

We use data from real-world social media websites, i.e., BlogCatalog[3] and Flickr[4]. The first two datasets are obtained from[19]. We crawled the third dataset to include four sources for further study. The number of communities, K, is determined by cross validation for each dataset.

BC is crawled from BlogCatalog, which is a blog directory where users can register their blogs under predefined categories. It contains 8,797 users and 7,418 tags. Two types of data are available: link and tagging information, and $K = 1,000$.

Flickr is an image sharing website in which users can specify tags for each image they upload. The dataset has 8,465 users and 7,303 tags with both link and tagging information, and $K = 500$.

BC-MS is collected from BlogCatalog with two additional sources besides link and tagging data: commenting and reading. It has 6,069 users and 5,161 tags. The four sources are S1 (linking), S2 (tagging), S3 (commenting), and S4 (reading), and $K = 500$.

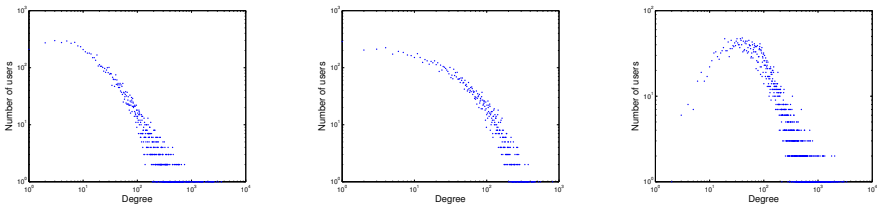
Some statistics of the datasets are shown in Table 1. We also compute the degree for each user. The distributions are shown in Figure 4, suggesting a power law distribution that is typical in social networks.

Since there is no ground truth about the online communities and the discovered communities are overlapping, we cannot compute the traditional metrics

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

Table 1. Statistics of the Datasets

	BC	Flickr	BC-MS
# of Users	8,797	8,465	6,069
# of Links	290,059	195,847	523,642
# of Sources	2	2	4
Ave Degree	66	46	173
Density	0.0075	0.0055	0.028
Clustering Coefficient	0.46	0.13	0.39



(a) BC (b) Flickr (c) BC-MS

Fig. 4. Degree Distributions

such as NMI and Modularity. Thus, we evaluate the quality of identified communities indirectly, which has been adopted by [22]. The basic assumption is that users belonging to the same communities should exhibit similar behaviors. Treating cluster memberships as features, randomly selecting a certain fraction of instances as training data and the rest as testing data, the evaluation is turned into a classification problem. We obtain the labels for each user from the social networking websites. In our work, the users' interests are treated as labels for each user. Linear SVM is adopted in our experiments since it scales well to large data sets. The training data size varies from 10% to 90% of the whole data. The experiments are repeated 10 times by shuffling data each time. Average Micro-F1 and Macro-F1 measures are reported.

Integrating Two Sources Cross validation is employed to determine the value of the regularization parameters, i.e., λ and η. We set λ to 0.05 in Flickr dataset and λ to 0.1 in both BC and BC-MS. Set η to 0.05 in all datasets. The iteration is stopped until the difference of the objective function between two consecutive steps is smaller than 1e-6. We focus on the two-source integration as in Eq. (3), which integrates link and another source (tagging, commenting, or reading) and compare it with single-source methods. PL, LL, EC and Modu work with the link matrix, and TC applies co-clustering to the user-tag data.

Tables 2 and 3 show the prediction performance on BC and Flickr, respectively. The first observation is that the prediction performance improves as when more training data is used. PL, LL, EC and TC show comparable performance for both datasets. The proposed integrative method using both links and tag-

Table 2. Performance on BC Dataset

Proportion of Labeled Nodes		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1(%)	2JointMF	**44.53**	**46.35**	**50.11**	**50.41**	**52.05**	**52.12**	**52.99**	**53.03**	**53.12**
	PL	28.94	28.85	30.85	31.20	32.25	33.10	33.11	33.42	33.60
	LL	26.61	26.24	26.57	26.73	27.74	26.63	27.50	27.38	27.99
	EC	24.85	25.55	26.27	25.18	25.28	24.80	24.11	23.94	22.22
	Modu	16.46	20.38	19.46	21.20	23.13	21.51	22.68	22.39	22.66
	TC	38.45	37.75	40.53	38.84	41.92	41.30	43.77	43.15	44.88
Macro-F1(%)	2JointMF	**29.01**	**31.12**	**34.76**	**35.54**	**36.99**	**37.59**	**38.02**	**39.11**	**39.39**
	PL	15.38	16.30	17.30	18.18	18.38	18.72	18.71	17.61	18.13
	LL	15.49	15.32	16.25	15.94	15.85	16.08	16.11	15.88	16.74
	EC	14.24	15.16	16.43	15.75	15.96	16.08	15.42	15.78	14.99
	Modu	9.32	9.34	10.61	11.39	10.53	11.01	11.01	9.69	11.66
	TC	28.85	26.83	27.68	28.52	28.18	29.69	28.60	30.16	29.96

Table 3. Performance on Flickr Dataset

Proportion of Labeled Nodes		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1(%)	2JointMF	**55.99**	**54.31**	**55.57**	**54.76**	**54.51**	**54.78**	**54.99**	**55.57**	**57.02**
	PL	42.03	44.53	44.72	45.22	46.68	47.68	47.90	48.43	49.27
	LL	40.80	41.17	42.49	42.55	43.13	44.16	45.69	45.88	46.51
	EC	39.62	39.93	40.93	41.12	41.79	41.75	42.06	42.57	43.44
	Modu	29.72	31.69	32.06	32.28	33.35	33.04	34.25	34.20	34.82
	TC	37.42	37.80	37.90	38.35	39.08	39.22	39.35	39.99	40.12
Macro-F1(%)	2JointMF	**30.62**	**30.81**	**31.13**	**31.49**	**32.04**	**32.12**	**31.99**	**32.11**	**32.42**
	PL	20.16	20.25	20.46	20.50	20.10	19.95	20.31	20.29	20.40
	LL	19.83	20.19	20.55	20.58	20.81	21.08	21.43	21.45	22.09
	EC	20.83	20.66	21.03	20.74	20.86	20.51	20.90	20.87	21.11
	Modu	15.35	13.25	13.45	13.37	13.10	13.29	13.78	13.92	14.14
	TC	20.65	20.49	21.03	20.90	20.80	20.68	21.06	21.28	21.35

ging information outperforms the single-source methods significantly. Compared to the best performance of baseline methods, on average, we achieve 17.2% and 31.5% improvement with respect to Micro-F1 in BC and Frickr, respectively. We obtain similar improvement in terms of Macro-F1. This directly supports that integrating different types of data in social media significantly improves the performance of community detection.

In addition, we report in Table 4 the performance of combining links with other data sources on BC-MS, such as S3 (commenting) and S4 (reading). Integrating an additional data source leads to much better performance. We also observe that different sources make uneven contributions to community quality: tagging information being the most, followed by commenting, and then reading. This implies that improvement might rely on the quality of sources.

Comparative Study of Integrative Methods In this section, we study performance of different data integration methods on BC-MS. We compare our

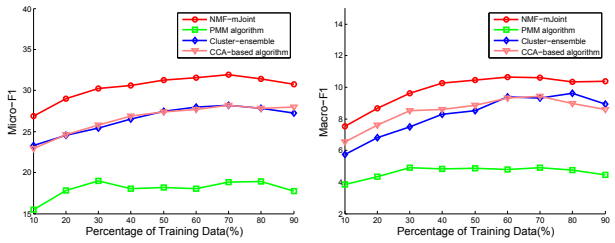
Table 4. Performance on BC-MS Dataset

Proportion of Labeled Nodes		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1(%)	2JointMF(S1+S2)	**29.95**	**32.87**	**33.47**	**34.23**	**34.53**	**34.80**	**35.04**	**34.56**	**34.91**
	2JointMF(S1+S3)	27.30	29.39	29.53	31.59	31.80	31.99	32.17	31.05	32.21
	2JointMF(S1+S4)	25.06	26.10	27.09	27.47	27.83	28.33	28.74	28.31	27.04
	PL	13.94	14.80	14.73	14.63	14.92	15.62	16.44	16.89	15.85
	LL	16.27	16.98	17.97	18.52	18.46	18.55	18.92	19.33	18.38
	EC	14.10	14.40	14.96	15.58	16.13	16.47	16.72	16.45	15.78
	Modu	9.25	12.81	12.85	14.41	12.62	13.22	13.49	14.15	13.69
	TC	14.79	14.69	15.32	15.95	15.63	15.24	16.31	16.91	16.50
Macro-F1(%)	2JointMF(S1+S2)	**9.66**	**11.14**	**12.75**	**13.01**	**13.08**	**13.11**	**13.07**	**12.75**	**13.05**
	2JointMF(S1+S3)	8.15	8.75	8.83	8.99	8.99	9.18	8.91	9.80	10.17
	2JointMF(S1+S4)	7.51	8.33	8.51	9.12	9.08	9.44	9.60	9.55	9.04
	PL	3.62	3.60	3.54	3.81	4.50	4.52	4.97	4.86	5.43
	LL	5.59	5.69	6.18	6.36	6.24	6.27	6.65	6.35	6.71
	EC	3.04	3.52	3.96	4.19	4.46	4.67	4.71	4.84	4.53
	Modu	1.92	2.57	2.81	3.22	2.71	2.71	3.13	3.10	2.95
	TC	4.06	4.11	4.58	4.85	4.91	5.04	5.12	5.00	5.25

multi-source method with three integrative baseline methods. PMM [20] first extracts the top eigenvectors of multiple data sources and combines them into a principal matrix, then obtain an overlapping clustering. Similarly, Canonical Correlation Analysis (CCA) can be used to find a transformation matrix for each source matrix such that the pairwise correlations between the projected matrices are maximized [3], and overlapping communities are then extracted. Cluster-ensemble [15] is adopted in this work to first compute the affiliation matrices for data sources, then combine them to find a consensus clustering. Note that these baseline integration methods have two stages: 1) integrating multi-source; 2) performing traditional community detection methods. In this experiment, since the input matrix can be negative, EdgeCluster is adopted as the basic community detection algorithm. However, our method performs multi-source integration and community detection simultaneously. All sources on BC-MS (linking, tagging, commenting, and reading) are integrated. The results are presented in Figures 5(a) and 5(b), respectively.

mJointMF gains 14.4% and 14.8% improvement of relative ratio compared with CCA-based method and Cluster-ensemble in terms of Micro, respectively. And it improves with relative ratios 15.7% and 20.5% compared with CCA-based method and Cluster-ensemble w.r.t Macro, respectively. In both cases, CCA-based and Cluster-ensemble have similar performance, however, PMM does not fare well.

Different Returns of Various Data Sources In this subsection, we try to investigate whether performance always improves as the number of sources increases. In earlier experiments, we observe that integrating an additional source with link data consistently improves performance over using only link infor-



(a) Micro (b) Macro

Fig. 5. Comparisons of Different Integrating Schemes

Table 5. Effects of Integrating Different Sources

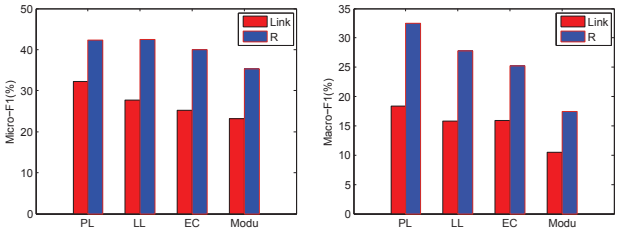
Proportion of Labeled Nodes		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1(%)	S1+S2	29.95	32.87	33.47	34.23	34.53	34.80	35.04	34.56	34.91
	S1+S3	27.30	29.39	29.53	31.59	31.80	31.99	32.17	31.05	32.21
	S1+S4	25.06	26.10	27.09	27.47	27.83	28.33	28.74	28.31	27.04
	S1+S2+S3	**32.56**	**33.61**	**35.01**	**35.67**	36.12	35.89	**36.99**	36.24	**36.90**
	S1+S2+S4	31.41	32.99	34.66	35.21	**36.54**	**36.11**	36.47	**36.41**	36.60
	S1+S3+S4	27.02	28.04	28.53	29.62	31.03	31.26	31.88	31.18	29.84
	S1+S2+S3+S4	26.90	28.98	30.21	30.61	31.27	31.53	31.91	31.40	30.76
Macro-F1(%)	S1+S2	9.66	11.14	12.75	13.01	13.08	13.11	13.07	12.75	13.05
	S1+S3	8.15	8.75	8.83	8.99	8.99	9.18	8.91	9.80	10.17
	S1+S4	7.51	8.33	8.51	9.12	9.08	9.44	9.60	9.55	9.04
	S1+S2+S3	**11.24**	11.60	**13.02**	**14.51**	**14.99**	**15.01**	**14.92**	**15.04**	**15.27**
	S1+S2+S4	10.96	**12.41**	12.99	13.19	13.48	13.99	14.38	14.66	14.84
	S1+S3+S4	8.54	8.51	8.47	8.68	9.17	8.91	9.24	9.80	8.65
	S1+S2+S3+S4	7.54	8.67	9.61	10.28	10.47	10.65	10.60	10.33	10.38

mation. We systematically examine performance by adding S2 (tagging), S3 (commenting), and S4 (reading) to S1 (linking).

As seen in Table 5, the benefit of having more data sources is not linearly associated with performance. Peak performance is achieved when integrating linking (S1), tagging (S2), and commenting (S3) in most cases. In some cases, adding another source can also worsen performance. Theorem 3 suggests that integration multiple source is divided into two phrases: 1) stacking other sources together; and 2) integrating it with link information. When more sources are added, the dimension will be increased significantly which will make the algorithm ineffective because of the curse of dimensionality; more noise may be introduced when more data sources are integrated; and redundant information may also exist in the sense that one source offers no new information due to the availability of other sources.

Validating Relationship Strengths for Community Detection In this section, we study how useful the estimated relationship strengths for link-based community detection algorithms. That is to say, we want to investigate if the link strengths estimated by our framework can help improve the performance of link-based algorithms. Four representative link-based algorithms are adopted in this experiment: PL, LL, EC, and Modu. The average Micro and Macro performance on 10 runs with 50% training dataset in BC and BC-MS datasets are shown in Figure 6 and Figure 7 respectively since similar results can be observed with other settings.

The performance of all four link-based algorithms is significantly improved. For example, on average, PL gains 34.4% and 66.7% improvement of relative ratio with respect to Micro performance in BC and BC-MS, respectively. And it improves with relative ratio 83.3% and 71.1% w.r.t Macro performance in BC and BC-MS, respectively. We have similar observations for LL, EC and Modu as well. These results indicate that the relationship strengths estimated by our framework can significantly improve the performance of link-based community detection algorithms.

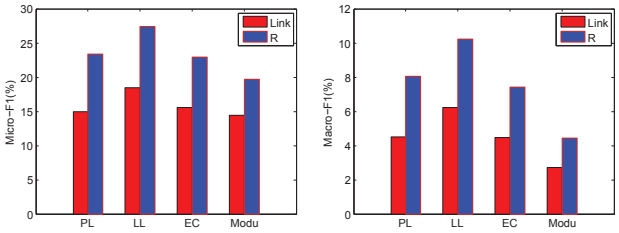


(a) Micro (b) Macro

Fig. 6. Comparisons of Performance in BC. Note that "Link" denotes performance on original link information and "R" represents performance on link information with estimated strengths

7 Conclusions

In this work, we study how to utilize social media data of different types for detecting communities. We propose an optimization framework to integrate multiple sources for community detection and estimating link strengths. Experimental results show promising findings: (1) integrating multiple data sources helps improve the performance of community detection; (2) different sources contribute unevenly to performance improvement of community detection; (3) having more data sources does not necessarily bring about better performance; and (4) the



(a) Micro (b) Macro

Fig. 7. Comparisons of Performance in BC-MS. Note that "Link" denotes performance on original link information and "R" represents performance on link information with estimated strengths

relationship strengths estimated by our framework can significantly improve the performance of link-based community detection algorithms.

This study also suggests some interesting problems for further exploration. Experimental results reveal that performance improvement might rely on the quality of sources. In order to find the relevant sources, we need efficient ways of studying the relationships between different sources as it is impractical to enumerate all sources to determine relevant sources even when the number of sources is moderately large. Exploring additional sources of social media data is another promising direction, e.g., incomplete user profiles, short and unconventional text like tweets may also be helpful.

Acknowledgments

The work is, in part, supported by ARO (#025071) and NSF (#0812551).

References

1. L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD*, pages 44–54. ACM, 2006.
2. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
3. K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, 2009.
4. E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5220, 2004.
5. T. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical Review E*, 80(1):16105, 2009.

6. Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: community discovery via relational hypergraph factorization. In *KDD*, pages 527–536. ACM, 2009.
7. Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: Joint models of topic and author community. In *ICML09*, 2009.
8. U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
9. M. McPherson, L. S. Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.
10. M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):36104, 2006.
11. M. E. Newman and E. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564, 2007.
12. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):26113, 2004.
13. G. Palla, I. Dernyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.
14. S. Scellato, C. Mascolo, M. Musolesi, and V. Latora. Distance matters: Geo-social metrics for online social networks. In *WOSN 2010*, 2010.
15. A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.
16. J. Tang, H. Gao, and H. Liu. mtrust: discerning multi-faceted trust in a connected world. In *WSDM*, pages 93–102. ACM, 2012.
17. J. Tang and H. Liu. Feature selection with linked data in social media. In *SDM*, 2012.
18. J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *KDD*, 2012.
19. L. Tang and H. Liu. Scalable learning of collective behavior based on sparse social dimensions. In *CIKM*, pages 1107–1116. ACM, 2009.
20. L. Tang, X. Wang, and H. Liu. Uncovering groups via heterogeneous interaction analysis. In *ICDM*, Miami, FL, USA, Dec. 6-9 2009.
21. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.
22. X. Wang, L. Tang, H. Gao, and H. Liu. Discovering overlapping groups in social media. In *ICDM*, Sydney, Australia, December 14 - 17 2010.
23. S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *SDM*, page 274. Society for Industrial Mathematics, 2005.
24. R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *WWW*, pages 981–990. ACM, 2010.
25. T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *KDD*, pages 927–936. ACM, 2009.