Community Detection with Edge Content in Social Media Networks

Guo-Jun Qi[1], Charu C. Aggarwal[2], Thomas Huang[1]

[1]*Department of Electrical and Computer Engineering*
University of Illinois at Urbana-Champaign
{qi4, t-huang1}@illinois.edu

[2]*IBM T.J. Watson Research Center*
charu@us.ibm.com

***Abstract*—The problem of community detection in social media has been widely studied in the social networking community in the context of the structure of the underlying graphs. Most community detection algorithms use the links between the nodes in order to determine the dense regions in the graph. These dense regions are the communities of social media in the graph. Such methods are typically based purely on the linkage structure of the underlying social media network. However, in many recent applications, edge content is available in order to provide better supervision to the community detection process. Many natural representations of edges in social interactions such as shared images and videos, user tags and comments are naturally associated with content on the edges. While some work has been done on utilizing node content for community detection, the presence of edge content presents unprecedented opportunities and flexibility for the community detection process. We will show that such edge content can be leveraged in order to greatly improve the effectiveness of the community detection process in social media networks. We present experimental results illustrating the effectiveness of our approach.**

***Index Terms*—ignore**

I. Introduction

Social networking has become an increasingly important application in recent years, because of its unique ability to enable social contact over the internet for geographically dispersed users. A social network can be represented as a graph, in which nodes represent users, and links represent the connections between users. An increased level of interest in the field of social networking has also resulted in a revival of graph mining algorithms. Therefore, a number of techniques have recently been designed for a wide variety of graph mining and management problems [2].

An important problem in the area of social networking is that of community detection. In the problem of community detection, the goal is to partition the network into dense regions of the graph. Such dense regions typically correspond to entities which are closely related, and can hence be said to belong to a *community*. The determination of such communities is useful in the context of a variety of applications in social-network analysis, including customer segmentation, recommendations, link inference, vertex labeling and influence analysis. As a result, a considerable amount of research has been devoted towards algorithms for solving this problem.

Most known techniques for community detection use only the information about the linkage behavior [1], [6], [14], [17], [27] for the purposes of community prediction and clustering. However, a lot of rich information is encoded in the *content of the interactions among the actors in the network*. Some recent work [26], [24] has shown that the use of *vertex* content can be helpful in improving the quality of the communities. However, we will see that edge content provides a number of unique distinguishing characteristics of the communities which cannot be modeled by node content. Some examples of networks with edge-content are as follows:

- In email networks, a communication between two participants can be considered an edge, which has content corresponding to the text which is communicated between two participants. Clearly, participants with similar content of communication are much more likely to belong to the same community than those which do not. This observation also applies to other forms of text or chat networks, or even (threaded) community boards which enable interaction between specific pairs of participants.
- In social media networks such as *Flickr*, users may tag an image with keywords. In such cases, it may be possible to construct a network of both people and images in which the edge content corresponds to the keywords which are used for tagging. Clearly such keywords provide important and useful knowledge about the nature of the underlying community.
- In many social media sites, users may share authorship or browsing behavior for the same content. In such cases,one can create an *actor-centric network* in which edges are placed between users that share the same content, and the shared content is associated with that edge. Thus, each content-based sharing may induce an edge between two participant nodes.

Edges provide a much richer characterization of community behavior, *because the content models the characteristics of pairwise interactions rather than individual actors.* In general, pairwise interaction content provides very specific information about the nature of the relationship between a particular pair of individuals. This implies that the different kinds of interactions

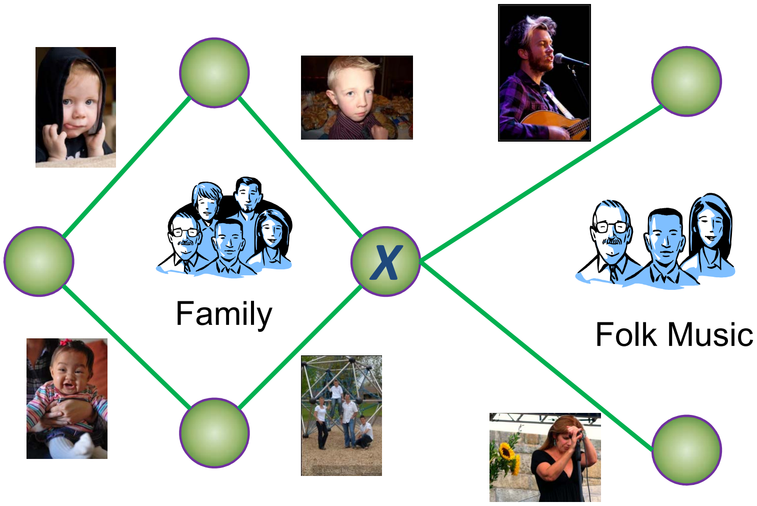


Fig. 1. Illustration of a social media network. The nodes represent users while the edges represent the favored images shared by the users. It is intuitively evident that the nodes can be easily partitioned into the *family* and *folk music* groups.

of a single individual may be used in order to reflect their membership in different communities. Figure 1 illustrates an example of a social media network. The nodes represent users while the edges represent the favored images shared by the users. In this example, it is evident that the content information associated with the edges can be naturally categorized into two types, corresponding to the *family* and the *folk music* themes. This naturally induces two kinds of edge-based interaction groups. It is particularly interesting to examine the edge content patterns of one of the central users marked by X in the figure. The actor X has edges which correspond to both themes of folk music and family, but with different sets of associates. While the interests of X are ambiguous from an *aggregate node-centric perspective*, it is clear from *specific pairwise edge interactions* that X should be placed in two separate communities with (largely) different members. This is a fairly common occurrence in social networks, because different parts of the same individual's interactions may show different patterns. This reflects the fact that a given individual may have different facets to her life, which are revealed only in her interactions with different people. The content on the different edges can be used to distinguish the nature of the community involvement. This also suggests that rather than directly trying to cluster the nodes, it may sometimes be more useful to focus on clustering of edges, and then creating an induced community of nodes which may allow for (node) overlaps across communities. We note that such overlapping nodes are often a challenge to community detection algorithms; however the edge content provides unparalleled insights which can be leveraged in order to create interesting communities.

When a community contains edges which are associated with similar content, and are also linked together tightly, the *interest area or expertise* of a community may also be identified on this basis. This can be useful when it identifies subject matter that is most relevant to the community. Such an approach can be very useful in problems such as expertise search. While some work [26], [24] has been done on incorporating content in the problem of community detection, most of the previous work is only designed for the case where the content is associated with the *nodes* rather than the *edges*. Edge-based content is much more challenging, because the different interests of the same actor node may be reflected in different edges. This paper will design a unique approach for community detection by tightly integrating the structural and content aspects of the network with the use of a matrix-factorization approach. We will show that such an approach provides unique insights which are not possible with the use of pure link-based or content-based methods.

This paper is organized as follows. The remainder of this section discusses related work. In section 2, we present a matrix factorization algorithm for edge-induced community detection. For simplicity, we will first present an algorithm which is based purely on structure only. Then, we present methods for incorporation of content information into the matrix factorization algorithm. The experimental results are presented in section 3. Section 4 presents the conclusions and summary.

II. Related Work

The problem of community detection has also been studied in the context of many graph-theoretic clustering algorithms. In its simplest form, a community may be considered as a group of nodes which are densely connected by edges. For example, a variety of node clustering algorithms for graphs with the use of shingling techniques, matrix co-clustering techniques, and tile determination in matrices [9], [10] can be used for community detection in graphs. The problem is also related to that of finding dense cliques or dense regions in the underlying graph [1], [16], [27]. These techniques are designed for generic graphs rather than the specific case of social networks. The problem of community detection [6], [14], [17],

[13], [22] in social networks has also been widely studied because of the increasing importance of social networking applications. A survey of a number of important algorithms for community detection is provided in [22]. Discussion of important statistical properties of web communities is discussed in [14]. A second related line of research is to use purely content-based clustering methods [4], [5], [18], [20]. However, such methods miss the rich information which is often encoded in the links in the underling network. Some recent work [26], [24] uses a combination of relational attributes and link information for clustering purposes. However, this method is designed for the case when the attributes are associated with the *nodes* rather than the *edges*. Some research [21] has been performed for visualizing the social network when the content is associated with the edges. The technique is designed to provide an intuitive visual understanding, and provides a good understanding of how the different regions in the various modes of the network relate to one another. However, it is not specifically designed for determining communities in an automated way with clear objective criteria.

III. Community Detection with Edge Content

Before discussing the algorithm in detail, we will introduce some notations and definitions. We assume that we have a social network $G = (\mathcal{V}, \mathcal{E})$ containing the vertex set $\mathcal{V}$ and the edge set $\mathcal{E}$. Each vertex in $\mathcal{V}$ corresponds to an actor in the network, and an edge corresponds to a relationship between this pair of actors. Associated with each edge e in $\mathcal{E}$, the content associated with it is in the form of a text document t_e. The problem of content-driven community detection is defined as follows:

Problem 1: Given a graph containing the vertices $\mathcal{V} = \{v_1 \ldots v_n\}$, an edge set $\mathcal{E}$ which is defined between the different vertices, and for each edge $e \in \mathcal{E}$, the content associated with is denoted by t_e, partition the **edge set** $\mathcal{E}$ into k communities $\mathcal{C}_1 \ldots \mathcal{C}_k$ which are based both on their linkages and the text content. Determine the vertex community $\mathcal{P}_i$ induced by the edge set $\mathcal{C}_i$.

The definition above is quite informal, as it does not specifically suggest what the objective functions for link-based and content-based similarity might be. We will set up a more formal framework later for algorithmic development. One interesting observation is that most community detection methods are focussed on *partitioning the nodes based on linkage*, and we are interested in *partitioning the edges* based on *both* linkage and content. Each edge community $\mathcal{C}_i$ induces a corresponding vertex community $\mathcal{P}_i$, and the different vertex communities may overlap. The intuition behind this model is that the same actor in a social network may have different interests, which may be reflected by their varied edge content to other actors. An example of this is the node X in Figure 1, who clearly belongs to two separate communities. The differences of the interactions of X with other actors will be evident from the differences in the underlying edge content in the two communities. Furthermore, it is reasonable to suggest that the individual belongs to both communities and should be included in both. The content on these edges can provide an idea of the nature of the interactions, and should therefore be carefully used for the community discovery process in social media networks. Therefore, communities can be partitioned in a more principled way with the use of the media content information on the edges. Therefore, while edges can be partitioned in a more coherent way, the vertices are harder to partition in a clean way, and are likely to have overlaps. In the extreme case, when there are no links, the problem defaults to the pure content-based clustering problem. The goal of this paper is to use the structural and content information judiciously in order to obtain the most effective results. We next define the concept of *induced vertex communities* from *edge communities*.

Definition 1 (Edge Induced Community): The induced community for a set of edges $\mathcal{C}_i$, is the set of vertices $\mathcal{P}_i$ which correspond to the end points of all edges in $\mathcal{C}_i$.

It is important to understand that most community detection problems are traditionally defined directly in the form of vertex partitions rather than as an induced set from edge partitions. The reason for this indirect approach is that in scenarios where substantial edge content is available, one can characterize content-based interests in a more coherent way with the use of the corresponding content on the edges. In practice, communities often have substantial overlaps, and these overlaps correspond to the different interests of the same actor; the interactions in terms of edge content provide the understanding needed to characterize the nature of these overlaps.

We note that while traditional community detection is designed with links only, the addition of edge content results in much greater *interpretability* to the clustering process, because it provides an intensional understanding of how the clusters relate to the content on the edges. Furthermore, it is possible that vertices which are poorly linked may sometimes belong to the same community because of a very high amount of similarity between the content itself. Thus, in some cases in which link connectivity and content-based similarity do not agree, it is important to set up criteria which can crisply regulate these tradeoffs. Thus, the problem is inherently a multi-criteria problem in which we wish to define an objective function $\mathcal{O}^l$, which corresponds to the *linkage based connectivity/density*, and an objective function $\mathcal{O}^c$ which reflects the content-based similarity among the text documents. In the next section, we formally formulate an edge-based clustering algorithm based on both link connectivity and content-based similarity.

A. Edge-Induced Matrix-Factorization

The core idea is to design a novel *edge-induced matrix factorization* (EIMF) algorithm for embedding of edges into a latent vector space based on link structure of the social network. This latent embedding algorithm is designed to discover the indicative factors for the underlying community structure based on the linkage connectivity among edges and vertices, and is naturally designed to incorporate content information. The link structure between edges and vertices are jointly explored in the *EIMF* algorithm and their latent vectors satisfy the constraints implied by their structural relationships. We will show that such a feature transformation can be used in conjunction with a standard k-means clustering algorithm [12] in order to discover effective communities.

The edge content can be naturally incorporated into the *EIMF* algorithm along with link connectivity so that the latent vectors of the edges with similar content are clustered together. The ability to discover a feature space which retains vector-based locality based on link structure as well as edge content is critical, because such a feature transformation enables the use of a simple vector-based k-means clustering algorithm [12] for community detection. Such an approach has the following properties:

- We construct a latent representation of vertices and edges, which are not independent of one another. The latent embedding of edges and vertices are based on their mutual structural dependency. Correspondingly, the latent vectors of a vertex can be represented as a function of the latent vectors of the incident edges. The key is to design a latent method which can jointly explore the structure of nodes and edges. Such an approach is particularly effective for community detection.
- We will see that the latent representation provides a natural way to incorporate the edge content. This complements the link structure, and improves the robustness of the clustering process, especially when the community structure is not completely clear from the edge and vertex patterns.

1) Mathematical Model: In this section, we will discuss the edge-induced matrix factorization model. For simplicity, we will first present a model which constructs a latent representation based on structural information only. In a later section, we will show how edge content can be naturally incorporated in this model. By using this two-step approach to presentation, the exposition is greatly simplified.

As introduced earlier, the social network is denoted by the pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertex set $\mathcal{V}$ contains the n nodes $\{v_1, \cdots, v_n\}$, and the and edge set $\mathcal{E}$ contains the m edges $\{e_1, \cdots, e_m\}$. Let Γ denote a $m \times n$ link matrix between edge and vertex set that encodes the underlying link structure. For each edge e_i and vertex v_j, the value of $\Gamma_{i,j}$ is set to 1 if v_j is incident to e_i. Otherwise, we have $\Gamma_{i,j} = 0$.

The goal of matrix factorization [28] is to derive a high-quality latent vector representation E for the edges based on an analysis of the link matrix Γ. The latent representation of the edge set $\mathcal{E}$ is denoted by E and the latent representation of vertex set $\mathcal{V}$ is denoted by V. Here, E is a $k \times m$ matrix with each column corresponding to a k-dimensional feature vector for each edge in the network. The latent edge matrix E can be expressed in terms of its latent column vectors as $\{\mathbf{f}_e(e_1), \cdots, \mathbf{f}_e(e_m)\}$. Each column of V is the latent feature vector for the corresponding vertex. Correspondingly, the latent vertex matrix V can be expressed in terms of its latent column vectors as $\{\mathbf{f}_v(v_1), \cdots, \mathbf{f}_v(v_n)\}$. The goal is to use these feature vectors of edges and vertices to capture the principal factors of the underlying link structure so that the edges and vertices in the same community are more likely to cluster together in the latent space. The core of the approach is to therefore determine a latent representation which can *effectively expose the community factors within the content and matrix structure*. Once such an optimum representation of the latent vectors of E are obtained, it is possible to obtain high quality communities by applying well known clustering methods (such as the k-means method) to the latent vectors in E in order to discover communities of edges. As discussed earlier, once the edge-based communities are known, then vertices can be assigned to the k communities induced by the edge partitions. Therefore, we will focus on designing such a latent representation which can expose the community factors well with the use of both structure and content.

We use the matrix factorization technique to design E and V such that the matrix product of E^T and V approximately represents the link matrix Γ. The matrix factorization technique sets up an optimization problem in order to determine these matrices approximately. Therefore, we wish to determine the optimum values $E = E^\star$ and $V = V^\star$, where these optimum values are defined by minimizing the error of the approximation. In other words, we have:

$$E^\star, V^\star = \underset{E,V}{\arg\min} \left\| E^T \cdot V - \Gamma \right\|_F^2 \tag{1}$$

where $\| \cdot \|_F$ denotes the Frobenius norm of a matrix. Essentially, the optimization problem aims at approximating each entry $\Gamma_{i,j}$ of the link matrix by $\mathbf{f}_e(e_i)^T \cdot \mathbf{f}_v(v_j)$, so that the obtained latent vectors can algebraically reflect the link relations between edges and vertices. Ideally, the latent vectors of an edge and a vertex should be orthogonal unless they are incident to each other. One can also impose the nonnegative constraints on E and V. The solution of such a model requires algorithms that are similar to probabilistic latent semantic analysis (PLSA) [11] and nonnegative matrix factorization (NMF) [23].

Conventional methods for matrix factorization [23] have gained success in link matrix analysis. However, these methods ignore the fact that the latent feature vectors of edges and vertices do not exist independently. In fact, they are closely related and ought to be induced by combining those of the edges incident upon it. In other words, it indicates that the latent vector of a vertex that describes its community factors can be derived by mixing the latent factors of the incident edges. Therefore, for any vertex v_j and its incident edges, its latent vector $\mathbf{f}_v(v_j)$ can be induced in terms of the incident edges:

$$\mathbf{f}_v(v_j) = \frac{1}{d\,(v_j)} \sum_{e_i \in \delta(v_j)} \mathbf{f}_e(e_i) \tag{2}$$

Here, $\delta(v_j)$ denotes the set of edges incident upon v_j, and $d(v_j) = |\delta(v_j)|$ is the degree of v_j. For example, in an email communication network, the latent vector that characterizes an individual (i.e., a vertex) can be obtained as a combination of the latent vectors from the communicated emails. The imposition of such a restriction helps us expose the community factors in the underlying vertices and edges much more effectively.

Let us define a $m \times n$ matrix Δ, whose entry $\Delta_{i,j}$ is defined to be $\frac{1}{d(v_j)}$ if $e_i \in \delta(v_j)$. Otherwise, the value of $\Delta_{i,j}$ is set to 0. Then, Eq. (2) can be compactly rewritten in matrix form as follows:

$$V = E \cdot \Delta \tag{3}$$

By substituting Eq. (3) into Eq. (1), we can obtain the optimal matrix factorization by minimizing the following:

$$\mathcal{O}^l\left(E\right) = \left\| E^T \cdot E \cdot \Delta - \Gamma \right\|_F^2 \tag{4}$$

This objective function captures the link structure of the network. Therefore, we denote the expression by $\mathcal{O}^l$. Later, we will discuss how to naturally combine the edge content with this latent model with the use of two different methods.

Before discussing the incorporation of content, we would like to make an observation about the the objective function of (4), which provides it with some advantages in terms of solvability. Unlike conventional matrix factorization, this objective function is jointly convex with respect to E and V. This makes the problem much easier to solve, especially when there are many local minima in the non-convex conventional model. This makes the *EIMF* approach more tractable for optimization purposes. One can solve this convex optimization problem by gradient-based methods, such as the conjugate gradient method and quasi-Newton method. At each step of these optimization methods, the main computation stems from computing the gradient of the above objective function with respect to the matrix E. This gradient can be expressed as follows:

$$\begin{aligned} &\nabla_E \mathcal{O}^l\left(E\right) = 2 \cdot E \cdot \Delta \cdot \Delta^T \cdot E^T \cdot E \\ &+2 \cdot E \cdot E^T \cdot E \cdot \Delta \cdot \Delta^T - 2 \cdot E \cdot \Gamma \cdot \Delta^T - 2 \cdot E \cdot \Delta \cdot \Gamma^T \end{aligned} \tag{5}$$

In a later section, we will show how such a gradient can be used effectively with content in order to solve this problem effectively.

B. An Example

To illustrate the difference of EIMF from the PCA-like matrix factorization, we conduct a case study here for an example as illustrated in Figure 2. The left part of the network forms a community (e.g., a golf club) with a clique of four vertices v_1, v_2, v_3 and v_4, which are fully connected with each other. Meanwhile, the right part forms the other community (e.g., the colleagues) with a clique of three vertices v_4, v_5 and v_6. We find that v_4 belongs to both communities in this example.

First, we perform the proposed edge-induced matrix factorization (EIMF). We have the incidence matrix

$$\Gamma = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{6}$$

and the matrix

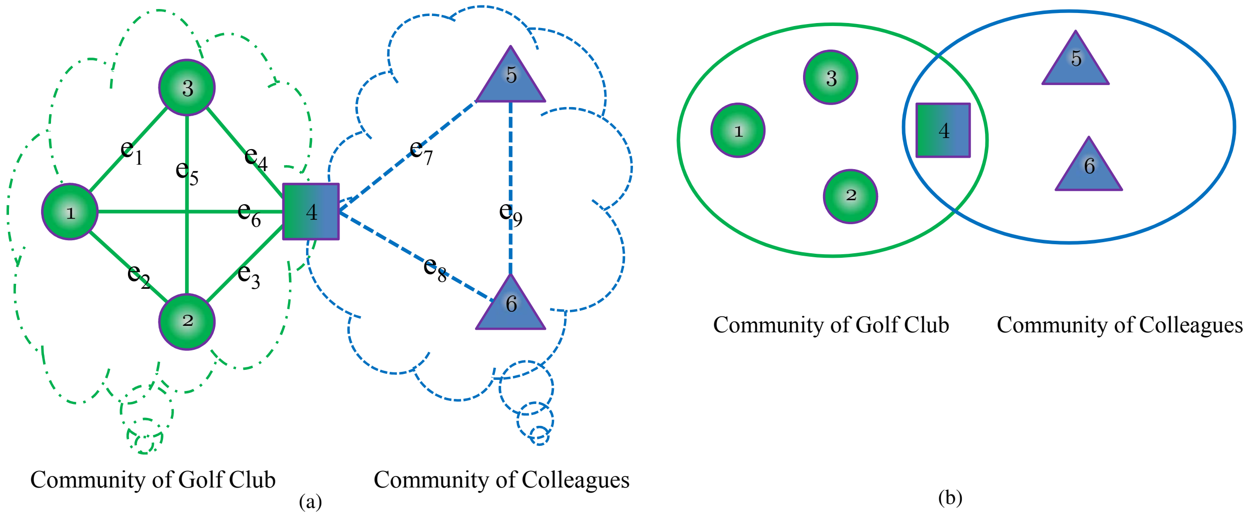
$$\Delta = \begin{bmatrix} 1/3 & 0 & 1/3 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 1/5 & 0 & 0 \\ 0 & 0 & 1/3 & 1/5 & 0 & 0 \\ 0 & 1/5 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/5 & 0 & 0 \\ 0 & 0 & 0 & 1/5 & 1/2 & 0 \\ 0 & 0 & 0 & 1/5 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix} \tag{7}$$

By minimizing Formulation (4), we can obtain the one-dimensional latent factors (i.e., real value) associated with each edge for community detection as shown in Figure 3. We can find that the obtained latent edge factors for the two communities are well separated. Especially, smaller factors indicate the corresponding edges are more likely to belong to the golf community while the larger factors indicate the memberships of colleague community. Moreover, for the edges in golf community, we find that e_1, e_2 and e_5 have smaller factors than that of e_3 and e_4. It is reasonable since e_1, e_2 and e_5 are fully contained in the golf club community in the sense that their incident vertices v_1, v_2 and v_3 completely belong to this community. On the contrary, the other two edges e_3 and e_4 only partially belong to this community as one of their incident vertex v_4 only has the mixed membership on both communities. The similar discussion is applied to the colleague community in the right part of the network. We also show the factors of the vertices computed by Eq. (2) in Figure 3(b). It is worth noting that the vertex v_4 has a latent factor between that of two communities, which indicates a mixed membership.

On the contrary, Figure 6(a) illustrates the results of the PCA-like matrix factorization. Comparing with the results as illustrated in Figure 3, we can find that the factor of v_4 does not properly combine the factors of its incident edges e_3, e_4 and e_6, which violates the relation in Eq. (2). As a result, one can find that it fails to detect the mixed membership of this vertex in both communities since the factor of v_4 falls into the colleague community with the same factor as v_5 and v_6. Moreover, the factors of e_3, e_4 and e_6 lie on the boundary of two communities and fail to indicate which community these edges belong to.

C. Incorporating Edge Content

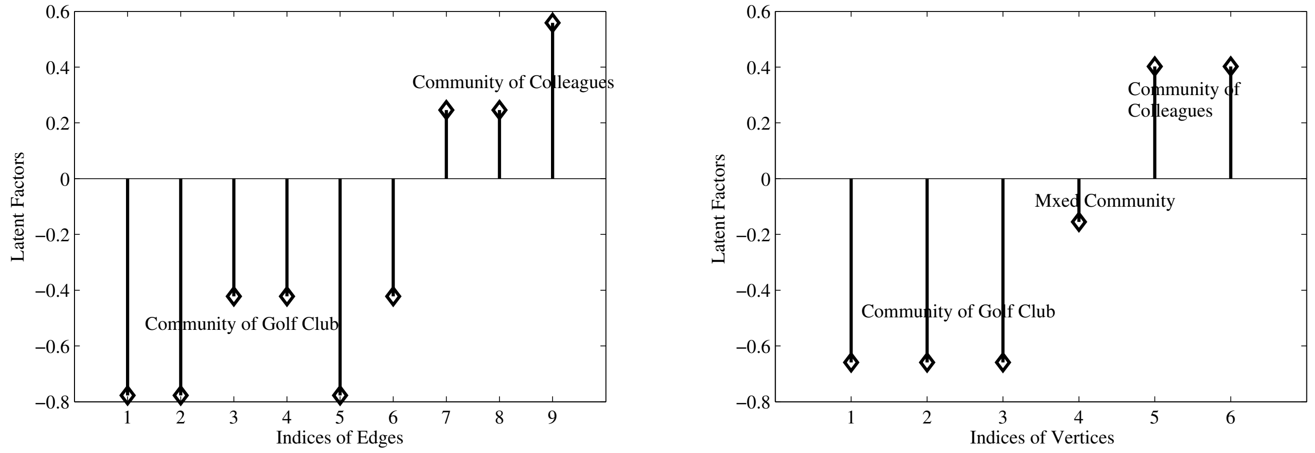
Since the model discussed in the previous section explicitly includes latent variables for the edges, it can be naturally ex-



(a)

(b)

Fig. 2. Illustration of a case study of edge-induced community discovery: (a)An example of Community Network; (b) Membership of vertices. In the subfigure (a), the left part forms a golf club community with a clique of four vertices, in which they communicate with each other. Similarly, the right part forms another colleague community with a clique of the other three vertices. Note that the vertex 4 belongs to both communities. The figure is best viewed in color.



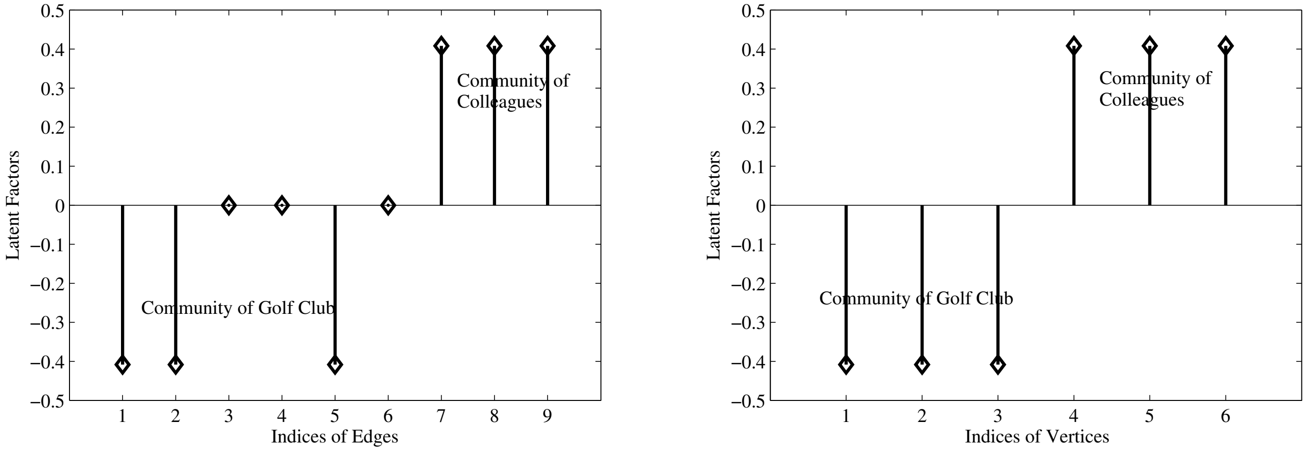
(a) Features of edges by EIMF

(b) Features of vertices by EIMF

Fig. 3. Use EIMF to find the indicative one-dimensional latent factors that discriminate the two communities in Figure 2. (a) the latent vectors (i.e., scalars) for each edge; (b) the latent factors for each vertex by Eq. (2). From the subfigure (a), we can find the latent factors of the edges for the two communities are well separated which successfully indicate the membership of each edge.

tended to the case of edge content. As mentioned earlier, edge content may provide additional rich information to detect the potential communities especially when individual interactions are inherently focussed towards multiple communities. For example, a vertex which is linked to multiple communities can often be disambiguated with the use of the content on the edges. Furthermore, the level of linkage within different communities may vary considerably, and the content is helpful in distinguishing the noise from the meaningful linkages. In particular, some of the less strongly linked vertices may sometimes belong to the same community if they share a large percentage of edges with similar content. On the other hand, it is also possible for some densely-linked vertices belong to the distinct communities if their associated edge contents are quite different. In this subsection, we will develop an additional content-based objective function $\mathcal{O}^c$ and show that it can be combined with $\mathcal{O}^l$ in order to derive an integrated feature representations with both link-connectivity and content-based similarity.

For each edge e_i, we assume that the content associated with it is denoted by c_i. For example, in an email network, c_i could be a text document which denotes the email communication between the two vertices. For the purpose of this paper, we assume that the content associated with each edge is text, though the general principles used in this paper can be extended to other content-types as well. We assume that the d-dimensional feature vector to represent the document c_i is denoted by $\mathbf{f}_c(c_i)$, in which, for example, each dimension represents the



(a) Features of edges by MF

(b) Features of vertices by MF

Fig. 4. Use the conventional *matrix factorization* (MF) to find the latent factors for discriminating the two communities in Figure **??**. (a) the latent factors (i.e., scalars) associated with each edge; (b) the latent factors associated with each vertex. Comparing with the results as illustrated in Figure 3, we can find the obtained latent vectors of edges and vertices do not satisfy the edge-induced assumption. Specifically, the latent factors of e_3, e_4 and e_6 lie on the boundary of two communities and fail to indicate their community memberships; moreover, the latent factors of v_4 does not properly mix those of its incident edges so that it fails to indicate the mixed membership of this vertex in both communities.

occurrence frequency of a word. For notational purposes, we introduce a $d \times m$ matrix C to denote these extracted feature vectors. In this matrix, the ith column contains the content feature vector $\mathbf{f}_c(c_i)$ associated with the edge e_i.

We design two different methods for incorporating the edge content into the community detection process. The first approach is designed with a direct use of the similarity between the feature vectors of different edges. For examples, consider two edges e_i and e_j, with associated content denoted by c_i and c_j, and corresponding feature vectors denoted by $\mathbf{f}_c(c_i)$ and $\mathbf{f}_c(t_j)$ respectively. In this case, one can compute the cosine similarity $S_{i,j}$ between the two feature vectors as follows:

$$S_{i,j} = \frac{\mathbf{f}_c(c_i)^T \cdot \mathbf{f}_c(c_j)}{\sqrt{\mathbf{f}_c(c_i)^T \cdot \mathbf{f}_c(t_i)} \cdot \sqrt{\mathbf{f}_c(c_j)^T \cdot \mathbf{f}_c(c_j)}} \tag{8}$$

While we have used the cosine similarity function because of its well-known effectiveness for the text domain, we note that our approach is not specific to the use of a particular similarity function. In general, the overall approach can be used in conjunction with different similarity functions for different content-types. We denote these similarity measures as the entries of a $m \times m$ matrix S.

Let s_i be the sum of elements of the ith row vector of the similarity matrix S. Let D be the diagonal matrix with $\{s_1, \cdots, s_m\}$ as its diagonal elements. Let L be the normalized Laplacian matrix of S, which is given by the relationship:

$$L = D^{-1/2} \cdot (D - S) \cdot D^{-1/2} \tag{9}$$

Then, one can use the Laplacian transformation in order to minimize the following content-based objective function in terms of the underlying *latent edge vectors*:

$$\begin{aligned} \mathcal{O}^c(E) &= \min_E S_{i,j} \cdot \left\| \frac{\mathbf{f}_e(e_i)}{\sqrt{s_i}} - \frac{\mathbf{f}_e(e_j)}{\sqrt{s_j}} \right\|^2 \\ &= \min_E \{tr\left(E^T \cdot L \cdot E\right)\} \end{aligned} \tag{10}$$

Here $tr(\cdot)$ represents the trace of the matrix. The determination of the optimal latent edge vectors provides a latent representation which exposes the communities based on the content-similarity between edges. Such a latent representation would be very useful for the community construction process. Then, by properly combining $\mathcal{O}^c$ and $\mathcal{O}^l$, we can obtain the optimal latent edge vectors in E by minimizing the following objective function:

$$\begin{aligned} \mathcal{O}(E) &= \mathcal{O}^l(E) + \lambda \cdot \mathcal{O}^c(E) \\ &= \left\| E^T \cdot E \cdot \Delta - \Gamma \right\|_F^2 + \lambda \cdot \mathrm{tr}\left(E^T \cdot L \cdot E\right) \end{aligned} \tag{11}$$

The objective function has two terms corresponding to structure and content, which are regulated with the use of the balancing parameter λ. As in the case of the structure-based objective function $\mathcal{O}^l$, this function is also convex. Therefore, we can use any gradient-based convex optimization solvers without being stuck in a local minimum. The gradient of this solver with respect to E is computed as follows:

$$\begin{aligned} \nabla_E \mathcal{O}(E) = {} & 2 \cdot E \cdot \Delta \cdot \Delta^T \cdot E^T \cdot E + 2 \cdot E \cdot E^T \cdot E \cdot \Delta \cdot \Delta^T \\ & -2 \cdot E \cdot \Gamma \cdot \Delta^T - 2 \cdot E \cdot \Delta \cdot \Gamma^T + 2 \cdot \lambda \cdot L \cdot E \end{aligned} \tag{12}$$

We denote this method by *EIMF-Lap* (or *EIMF-Laplacian*) since this method uses the Laplacian approach in the optimization process. The *EIMF-Lap* approach combines the link and content objective functions with a balancing parameter λ. One drawback of this is that it involves extra effort of tuning a proper parameter.

In order to avoid this, we can design an approach which learns a linear projection that maps the document feature vectors $\mathbf{f}(c_i)$ to the edge feature vectors $\mathbf{f}_e(e_i)$ with the use of a $k \times d$ transformation matrix W. The latent edge vector $\mathbf{f}_e(e_i)$ is expressed by multiplying the transformation matrix W with the content vector $\mathbf{f}_c(c_i)$ as follows:

$$\mathbf{f}_e(e_i) = W \cdot \mathbf{f}_c(c_i) \tag{13}$$

This can also be expressed as $E = W \cdot C$ in the matrix form. We note that the matrix W is not known, and needs to be learned in order to optimize the content-based community formation. This linear projection can capture the link structure by incorporating it into the *EIMF* approach in Eq. (4). Thus, the transformation matrix W can be derived by minimizing the following:

$$\Omega(W) = \left\| C^T \cdot W^T \cdot W \cdot C \cdot \Delta - \Gamma \right\|_F^2 \tag{14}$$

This method parameterizes E with the use of the linear projection matrix W. Therefore, it suffices to optimize the expression with respect to W rather than E. Once the optimum value of W has been obtained, we can determine the appropriate latent representation by using the relationship between E and W.

We denote this method by *EIMF-LP* (or *EIMF-linear projection*) in the following sections. The objective function $\Omega(W)$ can be minimized with the use of any convex optimization solver, which uses the gradient descent with respect to the parameter W. The gradient of Ω with respect to W is computed in each step as follows:

$$\begin{aligned} \nabla_W \Omega(W) = 2 \cdot W \cdot C \cdot \Delta \cdot \Delta^T \cdot C^T \cdot W^T \cdot W \cdot C \cdot C^T + \\ +2 \cdot W \cdot C \cdot C^T \cdot W^T \cdot W \cdot C \cdot \Delta \cdot \Delta^T \cdot C^T - \\ -2 \cdot W \cdot C \cdot \Gamma \cdot \Delta^T \cdot C^T - 2 \cdot W \cdot C \cdot \Delta \cdot \Gamma^T \cdot C^T \end{aligned} \tag{15}$$

As we will see later, we can greatly speed up many of these update techniques with the use of iterative learning techniques.

D. Multiplicative Update Rule

The direct optimization of the objective functions in the *EIMF-Lap* and *EIMF-LP* methods by convex programming solvers may be computationally expensive, especially for the large-scale social networks. In order to reduce computational costs, we propose a multiplicative update algorithm based on Oja's iterative learning rule [15] [25].

First, let us consider the objective function defined by *EIMF-Lap* in Eq. (10). We intend to decompose the gradient in Eq. (12) into its positive and negative compoents. First, we the define positive and negative components of the Laplacian matrix L used in the expression. These are $L_+ = I$ and $L_- = D^{-1/2} S D^{-1/2}$ respectively. The gradient $\nabla_E \mathcal{O}(E)$ of Eq. (12) can be decomposed into its set of positive components ∇_+ and the set of negative components ∇_- as follows:

$$\nabla_E \mathcal{O}(E) = \nabla_+ - \nabla_- \tag{16}$$

On ignoring constant terms in the gradient, it is evident from Eq. (12), that ∇_+ and ∇_- may be defined as follows:

$$\begin{aligned} \nabla_+ &= E \Delta \Delta^T E^T E + E E^T E \Delta \Delta^T + \lambda L_+ E \\ \nabla_- &= E \Gamma \Delta^T + E \Delta \Gamma^T + \lambda L_- E \end{aligned} \tag{17}$$

Then, one can use the results in [15] in order to define an iterative learning based update rule with the use of ∇_+ and ∇_- as follows:

$$E_{ij}^{\text{new}} \leftarrow E_{ij} - \eta_{ij} \left[\nabla_E O(E) \right]_{ij} = E_{ij} - \eta_{ij} \left[\nabla_+ - \nabla_- \right]_{ij} \tag{18}$$

Here η_{ij} is a positive learning rate. One can choose $\eta_{ij} = \dfrac{E_{ij}}{[\nabla_+]_{ij}}$, and the update rule becomes a multiplicative update rule:

$$\begin{aligned} E_{ij}^{\text{new}} &\leftarrow E_{ij} - \frac{E_{ij}}{[\nabla_+]_{ij}} \left[\nabla_+ - \nabla_- \right]_{ij} = E_{ij} \frac{[\nabla_-]_{ij}}{[\nabla_+]_{ij}} \\ &= E_{ij} \frac{\left[E \Gamma \Delta^T + E \Delta \Gamma^T + \lambda L_- E \right]_{ij}}{\left[E \Delta \Delta^T E^T E + E E^T E \Delta \Delta^T + \lambda L_+ E \right]_{ij}} \end{aligned} \tag{19}$$

The value of E can be initialized to a nonnegative matrix, and the above multiplicative update rule can be used to maintain nonnegativity. The value of E_{ij} increases when $[\nabla_-]_{ij} - > [\nabla_+]_{ij}$, and therefore $\left[\nabla_E O(E) \right]_{ij} < 0$. On the other hand, E_{ij} decreases if $\left[\nabla_E O(E) \right]_{ij} > 0$. The multiplicative rule converges in two cases. The first case is when $[\nabla_-]_{ij} = [\nabla_+]_{ij}$. This implies that $\nabla_E \mathcal{O}(E) = 0$ is the stationary point of the objective function. The second case is when $E_{ij} \to 0$, which yields the sparsity in E.

A similar discussion can be applied to the second objective function corresponding to *EIMF-LP*. In this case, one can decompose the gradient as follows:

$$\nabla_W \Omega(W) = \nabla_+ - \nabla_-$$

The positive and negative components (denoted by ∇_+ and ∇_- respectively) can be defined as follows:

$$\begin{aligned} \nabla_+ &= W C \Delta \Delta^T C^T W^T W C C^T + W C C^T W^T W C \Delta \Delta^T C^T \\ \nabla_- &= W C \Gamma \Delta^T C^T + W C \Delta \Gamma^T C^T \end{aligned}$$

As in the previous case, these can be used in conjunction with the the results in [15] in order to define the following multiplicative update rule:

$$\begin{aligned} W_{ij}^{\text{new}} &\leftarrow W_{ij} \frac{[\nabla_-]_{ij}}{[\nabla_+]_{ij}} \\ &= \frac{W_{ij} \left[W T \Gamma \Delta^T C^T + W C \Delta \Gamma^T C^T \right]_{ij}}{\left[W C \Delta \Delta^T C^T W^T W C C^T + W C C^T W^T W C \Delta \Delta^T C^T \right]_{ij}} \end{aligned} \tag{20}$$

As in the previous case, the value of W can be initialized to be nonnegative, and the update rule subsequently maintains it. The iterative update of W_{ij} converges whenever either a stationary point is achieved (corresponding to $[\nabla_W \Omega(W)]_{ij} = 0$), or the solution satisfies the sparsity property (corresponding to $W_{ij} \to 0$). Since the link matrices Γ and Δ are usually very sparse (there are only $2m$ nonzero entries in these two matrices where m is the number of edges in the social network), the multiplicative update rules corresponding to Eq. (19) and Eq. (20) can be computed efficiently in each step. With the obtained nonnegative representation, the cluster label for each edge can be assigned based on the maximum factor in each latent vector.

IV. Experimental Results

In this section, we will compare the effectiveness of the EIMF algorithms with the other state-of-the-art community detection algorithms on two data sets. In the following, we will describe the data sets, performance metrics and the experimental setup in detail.

A. Data Sets

The following two data sets were used for evaluation:

- *Enron Email Data Set:* This data set consists of a large number of email messages between employees of the Enron corporation, which were collected and used for a legal investigation of its financial troubles in the year 2000. The Enron corpus contained $200,399$ messages belonging to 158 members of senior management of Enron. Each user had an average of about 757 messages. From a network modeling perspective, the users correspond to modes that are linked by the email communications (edges) between them. The edges are associated with the content of email messages. The feature vectors are extracted from each email by counting the frequencies of extracted word tokens. One useful characteristic of a particular version of the data set, was that it was annotated by students at the University of California at Berkeley. This was very useful for evaluation purposes. This subset of $1,700$ emails are labeled by 53 categories, which focuses on business-related emails and the California Energy Crises. As we will see later, such labeling can be leveraged in order to measure the quality of the clustering.[1] Based on these annotated categories, the users are assigned to the 53 clusters based on their email communications according to the edge-induced rule.
- *Flickr Social Network Data Set:* This data set contained 15 popular Flickr user groups, including "family", "auto" , "concerts", "pet portraits", "kids and nature", "street art," "wide party," "folk music," "magic city," "party favors", "British politics", "youth basketball", "fast food", "fancy dress party", and "great sky." These groups are collected using the keyword-based group search functionality provided by Flickr. The most popular tags were used as queries. This social media network has $4,703$ users in 15 groups, and each user can join more than one group. We note that users have the ability to mark their favored images in these groups. We use these favored images in order to create a graph of users in which the edges reflect an interest in the same image. In order to enable this, s totsl of $26,920$ favored images were collected from Flickr. In order to construct the social media network, two users are linked by edges if they favor the same images. For each image, users also tag some keywords to describe its content. The edge content is the union of the user tags on the associated images. The user tags are stemmed and the stop words and meaningless keywords are removed. In general, the user tags provided a richly descriptive characterization of the underlying images.

B. Performance Metrics

As mentioned in the previous section, the data sets are associated with class labels in addition to the content. These class labels turned out to be used in order to measure the effectiveness of the community detection process. In order to measure the effectiveness of the community detection algorithm, two metrics are used in the experiments. These were the *pairwise F-measure (PWF)* and *average cluster purity (ACP)* respectively. These metrics are both supervised metrics, which are constructed with the use of the community ground truth (or class labels) collected in the data sets. Since clustering is an unsupervised problem, the ground truth information was not used during the clustering process. The class information about the communities is only used for evaluation purposes. This provides a robust evidentiary measure about the quality of the clustering.

Pairwise Precision, Recall and F-measure. We note that our community detection approach allows for overlaps, and can assign each node to more than one cluster. Furthermore, the ground-truth may also allow for overlaps, when multiple groups were associated with a node. Therefore, we need to revise the commonly used pairwise precision and recall measures for clustering algorithms [24], in order to create a meaningful measure. Let G denote the set of node pairs that share *at least* one cluster class. Similarly, let H denote the set of node pairs that are assigned to *at least once* to the same cluster by the algorithm. Then, we can compute the pairwise precision and recall as follows:

$$\mathrm{pr} = \frac{|H \cap G|}{|H|}, \mathrm{rc} = \frac{|H \cap G|}{|G|}$$

The afore-mentioned measures of precision and recall can be used in order to define the *pairwise F-measure* as follows:

$$\mathrm{PWF} = \frac{2 \times \mathrm{pr} \times \mathrm{rc}}{\mathrm{pr} + \mathrm{rc}}$$

A higher value of the *pairwise F-measure (PWF)* suggests that the underlying clustering is of good quality.

Average Cluster Purity: The average cluster purity is computed as the average percentage of the *dominant community* in the different clusters. Formally, let $\mathcal{C} = \{C_1, \cdots, C_K\}$ be the k clusters determined by the algorithms. Let us assume that the number of points in C_i, are denoted by n_i. The corresponding set of n_i vertices is denoted by $\{v_{1,i}, \cdots, v_{n_i,i}\}$. Let $M_{l,i}$ denote the set of communities that $v_{l,i}$ truly belongs to in the ground truth of labels. Then, the *average cluster purity (ACP)* is defined as follows:

$$ACP = \frac{1}{k} \sum_{i=1}^{k} \sum_{l=1}^{n_i} \frac{\delta\left(dom_i \in M_{l,i}\right)}{n_i}$$

Here, $\delta(\cdot)$ is an indicator function, which indicates whether the dominant class dom_i of cluster C_i matches with at least one of the labels for a vertex.

[1] The data set along with documentation may be found at: http://bailando.sims.berkeley.edu/enron_email.html.

TABLE I
COMPARISON OF DIFFERENT COMMUNITY DETECTION ALGORITHMS ON ENRON EMAIL DATA SET AND FLICKR SOCIAL MEDIA DATA SET.

Algorithms		Enron Email Data Set				Flickr Social Media Data Set			
		precision	recall	PWF	ACP	precision	recall	PWF	ACP
Link only	Newman	0.3333	0.2	0.25	0.2018	0.3263	0.1615	0.2161	0.1029
	LDA-Link	0.21	0.1221	0.1544	0.1126	0.2566	0.1195	0.1631	0.1122
	MF	0.4643	0.1016	0.1667	0.1945	0.1788	0.0664	0.0968	0.064
Content	LDA-WORD	0.4172	0.2636	0.3231	0.2787	0.3333	0.3438	0.3385	0.1789
	NCUT-Content	0.4855	0.3223	0.3874	0.3576	0.4469	0.3076	0.3644	0.2746
Link+ node content	LDA-Link-Word	0.5597	0.3675	0.4437	0.4027	0.4706	0.3692	0.4138	0.3164
	NCUT-Link-Content	0.6986	0.3835	0.4952	0.4231	0.5024	0.3181	0.3896	0.3432
Link + edge content	EIMF-Lap	0.6752	0.5526	0.6078	0.5641	0.5634	0.4441	0.4967	0.4348
	EIMF-LP	0.6522	0.5696	0.6081	0.549	0.6038	0.4017	0.4824	0.4734

C. Community Detection Results

In order to validate the effectiveness of our algorithms, we need to show that the obtained communities are more effective than other competing methods. For this purpose, we used the following baselines:

- We used some *link-based techniques* in which we cluster the nodes using known structural methods in community detection. In particular, we tested with the use of the Newman's algorithm [6], the LDA-Link [7]), and normalized cut (NCUT) [19] which is a spectral clustering algorithm. These algorithms only use the link structure to partition the nodes in social networks into communities.
- We used a *pure content-based approach* where we are simply clustering the documents on the edges, with the use of a text clustering approach. We used the LDA-Word [3] and NCUT-content algorithms in order to cluster the content on the edges in the *Enron* and *Flickr* data sets. Once, the edges have been partitioned, they are used to induce the nodes into different communities based on the edges incident upon them.
- A community detection approach, which uses content in the **nodes** along with the links. In such a case, we also need an equivalent way for modeling the content at the nodes as opposed to the edges for the same scenario. For example, for the Enron data set, the content at a node is the concatenation of all emails sent by a participant, and for the Flickr data set, the content at a node is the union of all user tags associated with the favored images. In this category, we use LDA-Link-Word and NCUT-Link-Content for comparison. LDA-Link-Word refers to the mixed membership model in [7], and NCUT-Link-Content refers to the spectral clustering algorithm with both link and content similarity between nodes [19].

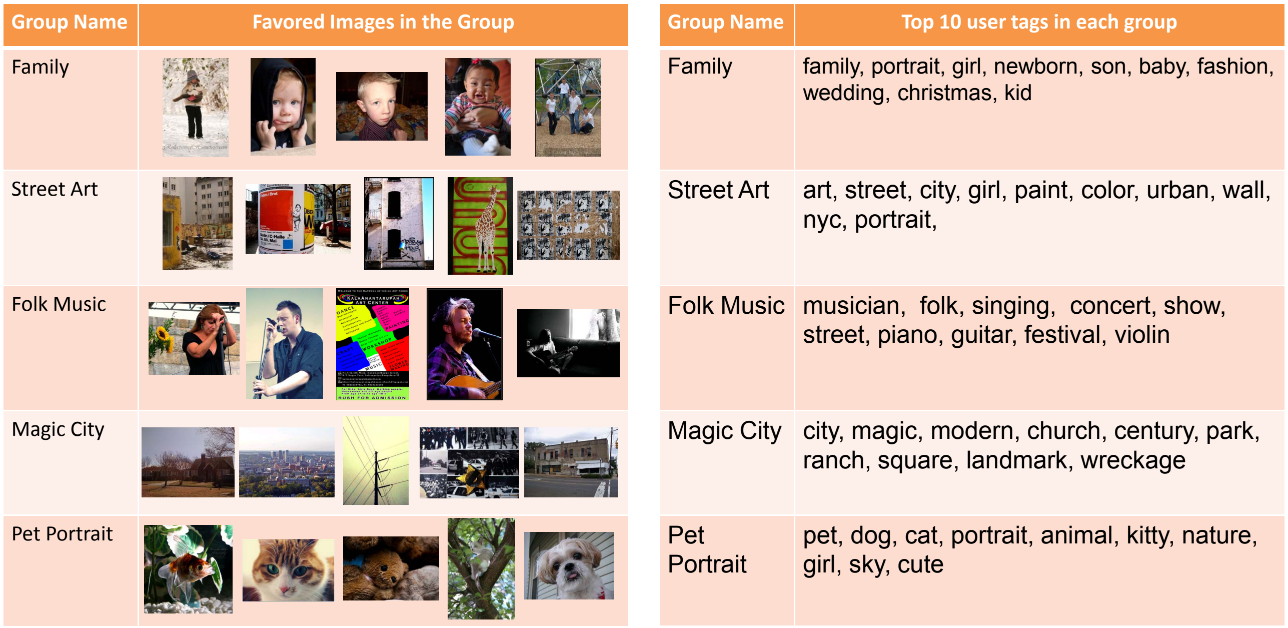
In the case of *EIMF-Lap*, the dimension of the latent space eas set to be equal to the number of clusters. This was 53 in the case of the Enron email data set and 15 in the Flickr social media data set respectively. For *EIMF-LP*, the transformation matrix also projects the representation into a 53 and 15 dimensional latent space. Once the latent space was obtained, a k-means clustering was applied in order to partition the embedded points in the latent space into different clusters for community detection.

The results for the different algorithms and data sets are illustrated in tabular form in Table I. We present the results in terms of pairwise precision, recall and F-measure. On both data sets, the two edge-content-based algorithms *EIMF-LP* and *EIMF-Lap* outperform the other algorithms, including the pure content and pure link-based algorithms, and also the algorithms which combine both link and node content. This confirms when detecting community structure in a network with content information on the edges, *EIMF-LP* and *EIMF-Lap* algorithms can achieve better performance than the other baseline algorithms. An interesting observation is that the algorithms with pure content-based information obtain better performances than the pure linkage-based algorithms. This suggests that the edge content may often contain useful information for the community detection process.

Another observation is that the algorithms which combine both edge content and links perform better than those combining node content and links. This is because in the email and social media networks, the content is naturally attached on edges rather than nodes. The algorithms that combine the node content and links concatenate the edge content together to represent the node content. This often reduces the effectiveness of the algorithm, because it mixes the content information from diverse edges. In many cases, this may lead to a reduction in the ability of the algorithm to discriminate among different communities.

We also provide some case studies about the kinds of communities we obtained. Figure 5 illustrates some examples of groups partitioned by the *EIMF-LP algorithm* on the Flickr social media network. Each row shows the favored images in this group whose name is given by the dominant group associated these images. We can find that the images in each group share a common theme and are consistent with the topics set up by the user tags in the corresponding group. This suggests that the method is able to determine coherent communities with the use of this approach.

We also tested the sensitivity of the *EIMF-Lap* method to different choices of the parameter λ. We illustrate the variation of the algorithm with λ in Figure 6. The value of λ is illustrated on the X-axis, and it varies from 0 to 3.0 with 0.5 as the step size. It is evident from the results that when no content information is incorporated ($\lambda = 0$), the *EIMF-*



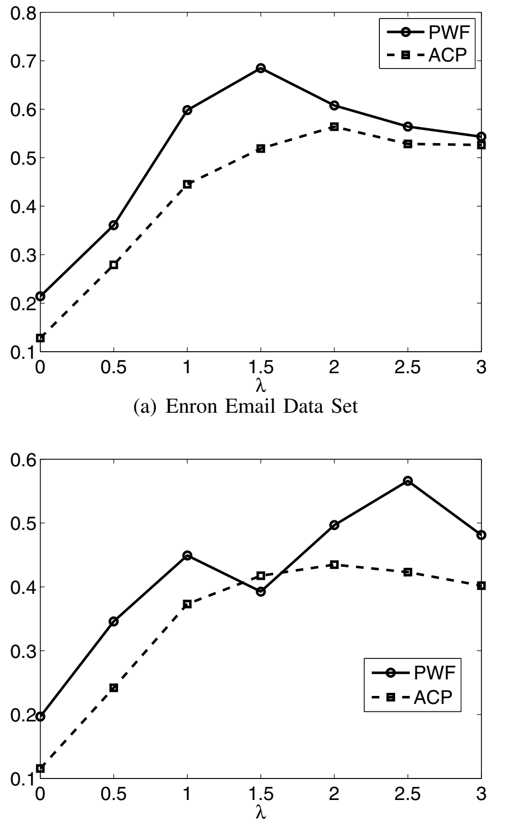
Group Name	Favored Images in the Group
Family	
Street Art	
Folk Music	
Magic City	
Pet Portrait	

(a) Favored images in each group.

Group Name	Top 10 user tags in each group
Family	family, portrait, girl, newborn, son, baby, fashion, wedding, christmas, kid
Street Art	art, street, city, girl, paint, color, urban, wall, nyc, portrait,
Folk Music	musician, folk, singing, concert, show, street, piano, guitar, festival, violin
Magic City	city, magic, modern, church, century, park, ranch, square, landmark, wreckage
Pet Portrait	pet, dog, cat, portrait, animal, kitty, nature, girl, sky, cute

(b) Associated user tags in each group.

Fig. 5. Illustration of groups partitioned by the EIMF-LP algorithm on Flickr social media network. (a) Favored images in each group; (b)User tags associated with each group. Each row shows some examples of the favored images and associated user tags.



(a) Enron Email Data Set

(b) Flickr Social Media Data Set

Fig. 6. Clustering performance with variation of balancing parameter λ for EIMF-Lap.

Lap algorithm does not perform well on either of the data sets. However, as λ increases, more content information is combined together with link structure and it performs better. However, such advantages drop off after a certain point, because the use of a value of λ which is too large reduces the importance of the link structure. This verifies that the edge content does help in modeling the community structure, and therefore the underlying effectiveness of the community detection algorithm. We experimentally used $\lambda = 2.0$ in the experiments It is evident from Table I, that the *EIMF-Lap* algorithm achieves competitive results on both data sets. By further tuning the parameter λ specific to the Enron and Flickr data sets, its performance could achieve better performance than *EIMF-LP*. On the other hand, the advantage of *EIMF-LP* is that it does not depend on such a parameter, and thus it does not need to be tuned. This makes the *EIMF-LP* algorithm more easily to apply in practical applications.

V. Conclusions and Summary

In this paper, we proposed an algorithm for community detection with edge content. Edge content provides unique insights into communities because it characterizes the nature of the interactions between participants more effectively. This is because the use of purely structural information cannot easily characterize the nature of the interactions between participants effectively. Similarly, the information which is available only at the nodes may not be able to easily distinguish the different interactions of nodes that belong to multiple communities. The use of edge content enables richer insights which can be used for more effective community detection. Our experimental results show the robustness of the approach over a number of content- and media-based data sets.

Acknowledgments

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. Guo-Jun Qi is also supported by an IBM Fellowship.

References

[1] J. Abello, M. G. Resende, and S. Sudarsky. Massive quasi-clique detection. In *LATIN*, 2002.

[2] C. Aggarwal and H. Wang. *Managing and Mining Graph Data*. Springer, 2010.

[3] D. M. Blei, A. Y. Ng and M. I. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research 3: pp. 993C1022, January 2003.

[4] D. Cutting, D. Karger, J. Pedersen, J. Tukey. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. *Proceedings of the SIGIR*, 1992.

[5] M. Franz, T. Ward, J. S. McCarley and W. J. Zhu, Unsupervised and supervised clustering for topic tracking, *ACM SIGIR Conference*, 2001.

[6] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. In *Phys. Rev. E 70, 066111*, 2004.

[7] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed membership models of scientific publications. In *PNAS 101*, 2004.

[8] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power law relationships of the internet topology. In *SIGCOMM*, 1999.

[9] D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *VLDB*, 2005.

[10] A. Gionis, H. Mannila, and J. K. Seppänen. Geometric and combinatorial tiles in 0-1 data. In *PKDD*, 2004.

[11] T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, 1999.

[12] A. Jain, and R. Dubes. Algorithms for Clustering Data, *Prentice Hall*, Englewood Cliffs, NJ, 1998.

[13] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *WWW*, 1999.

[14] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW*, 2008.

[15] E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, (5):927 – 935, 1992.

[16] J. Pei, D. Jiang, and A. Zhang. On mining cross-graph quasi-cliques. In *ACM KDD Conference*, 2005.

[17] V. Satulouri and S. Parthasarathy. Scalable graph clustering using stochastic flows: Applications to community discovery. In *KDD Conference*, 2009.

[18] H. Schutze, C. Silverstein. Projections for Efficient Document Clustering, *ACM SIGIR Conference*, 1997.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

[20] C. Silverstein, J. Pedersen. Almost-constant time clustering of arbitrary corpus sets. *Proceedings of the ACM SIGIR*, pages 60-66, 1997.

[21] J. Sun, S. Papadimitriou, C.-Y. Lin, N. Cao, S. Liu, W. Qian. MultiVis: Content-based Social Network Exploration Through Multi-way Visual Analysis, *SIAM Conference on Data Mining*, 2009.

[22] W. Tang and H. Liu. Graph mining applications to social network analysis. In *Managing and Mining Graph Data, Ed. Charu Aggarwal, Haixun Wang*, 2010.

[23] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, page 267C273. ACM Press, 2003.

[24] T. Yang, R. Jin, Y. Chi, S. Zhu. Combining link and content for community detection: a discriminative approach. *ACM KDD Conference*, 2009.

[25] Z. Yang and J. Laaksonen. Multiplicative updates for non-negative projections. *Neurocomputing*, (71):363–373, February 2007.

[26] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *Proceedings of VLDB*, 2(1):718–729, 2009.

[27] Z. Zeng, J. Wang, L. Zhou, and G. Karypis. Out-of-core coherent closed quasi-clique mining from large dense graph databases. In *ACM Transactions on Database Systems, Vol 31(2)*, 2007.

[28] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.