2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

Identifying Cohesive Subgroups and their Correspondences in Multiple Related Networks

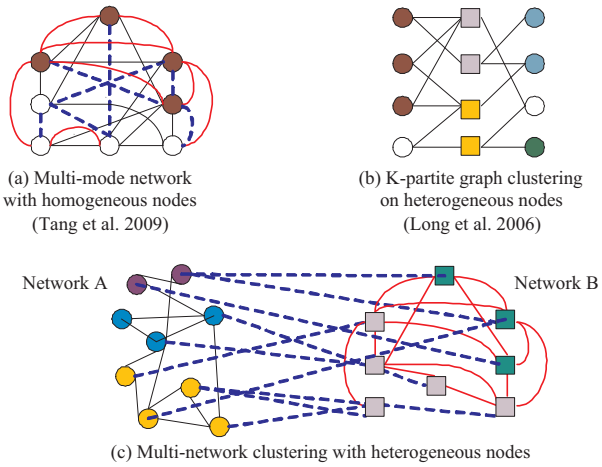
Prakash Mandayam-Comar, Pang-Ning Tan, Anil K Jain
Department of Computer Science and Engineering
Michigan State University
East Lansing, USA
Email: (mandayam,ptan,jain)@cse.msu.edu

***Abstract*—Identifying cohesive subgroups in networks, also known as clustering is an active area of research in link mining with many practical applications. However, most of the early work in this area has focused on partitioning a single network or a bipartite graph into clusters/communities. This paper presents a framework that simultaneously clusters nodes from multiple related networks and learns the correspondences between subgroups in different networks. The framework also allows the incorporation of prior information about potential relationships between the subgroups. We have performed extensive experiments on both synthetic and real-life data sets to evaluate the effectiveness of our framework. Our results show superior performance of simultaneous clustering over independent clustering of individual networks.**

I. Introduction

Clustering of network data [14], [15], [18], [10], [20], [19] has been an active research area in data mining and machine learning, spurred by the availability of rich collection of relational data in biological, social, and bibliographic network domains. The objective of clustering is to partition a network into cohesive groups of nodes in such a way that the nodes within a group are highly connected with each other and are mostly disconnected from nodes in other groups. Identifying closely-knit groups in a network has many practical applications, such as for functional module identification in biological networks, community finding in online social network users, and subgroup detection in criminal and terrorist networks.

Network data are typically represented as graphs and most of the early research in this area has focused on clustering nodes in a single graph [14], [15], [21], [20]. There have been recent efforts to extend the problem to clustering multiple graphs. For example, Tang et al. [17] proposed a linked matrix factorization approach for fusing information from multiple graph sources. However, their approach was designed to cluster multiple graphs in which each graph represents a specific type of proximity relation between the same set of nodes in a given multi-mode network. Lin et al. [11] also investigated a similar problem using a relational hypergraph factorization approach to detect communities of users based on various social contexts and interactions.



(a) Multi-mode network with homogeneous nodes (Tang et al. 2009)

(b) K-partite graph clustering on heterogeneous nodes (Long et al. 2006)

(c) Multi-network clustering with heterogeneous nodes

Figure 1. Comparison between multi-network clustering (bottom figure) against multi-mode network and k-partite graph clustering (top figures).

The focus of this paper is to simultaneously cluster multiple networks of heterogeneous nodes, taking into consideration the correspondences between the subgroups. Our formulation differs from previous studies in co-clustering [4], [1] and k-partite graph clustering [12], [3] in that it incorporates the connectivity between nodes of the same type into the clustering process (unlike k-partite graph clustering which considers only links between nodes of different types). The distinction between the various clustering problems is illustrated in Figure 1.

Our work on clustering multiple networks of heterogenous nodes is motivated by its many potential applications. For example, it can be used to simultaneously find clusters of scientific papers and clusters of authors working in the same research areas. Similarly, it can also be used to perform joint clustering on Wikipedia articles and editors of the Wikipedia pages. The multiple networks may also represent relational data from different domains. For example, one could perform joint clustering of Wikipedia editors and *Digg*[1] users, where

[1]Digg.com is a social networking web site that allows users to share news stories with other users.

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.226

the links between Wikipedia editors and Digg users are established based on the content similarity between the edited Wikipedia pages and the submitted news stories in Digg. The advantages of multi-network clustering are that

- Attribute set of nodes in individual networks may not be rich enough for the purpose of clustering.
- An individual network may have noisy or partially observed links. In such a case the link structure may be enhanced by considering information from other associated networks.

A naive approach for multi-network clustering is to partition each network separately. Such an approach is useful when the link structure and subgroup information in different networks are independent of each other. However if the networks are related to each other, then the link structures in the individual networks are not only characteristic of the respective networks but often contains implicit information about the underlying clusters of other related networks. In such a scenario, we expect a joint clustering would enhance the performance of the clustering algorithm.

Our contribution in this paper is to present a framework for simultaneous clustering of multiple networks by jointly factorizing their adjacency matrices to obtain the clusters. Our framework is equally applicable to clustering multiple networks created from heterogeneous nodes of the same source (e.g., Wikipedia articles and editors) or nodes from different sources (e.g., Wikipedia editors and Digg users) as long as the corresponding links between nodes in different networks can be established. Another important contribution is that the framework can be extended to incorporate prior knowledge about the potential correspondences between subgroups in different networks. We have performed extensive experiments on both synthetic and real-world data to compare the performance of joint clustering against independent clustering. Our results assert the superior performance of joint clustering over independent graph clustering, especially when prior information is used.

The remainder of this paper is organized as follows. Section II presents the related work on clustering network data. Section III formalizes the multi-network clustering problem. Section IV describes the proposed framework along with formal proofs of convergence and correctness. Section V presents the experimental results, followed by conclusions and suggestions for future work in Section VI

II. Related Work

Clustering of nodes in a network [14], [15], [20] has been studied for a long time as a graph partitioning problem. The general goal is to partition the graph in such a way that the intra-cluster links (links within each partition) are maximized and inter-cluster links (links between partitions) are minimized. To this extent, several approaches have been successfully applied, including techniques based on spectral clustering [2], [16], multi-level graph partitioning [6], [7], and matrix factorization approaches [21]. All of these algorithms focused on partitioning a homogeneous set of nodes into clusters or communities. There has also been considerable research towards identifying multiple overlapping clusters in networks [19] and finding evolving communities in a dynamic network [11], [18].

Recent research has also focused on simultaneous clustering of bipartite graphs constructed from related sets of heterogeneous data, such as documents-words, products-users, blogs-bloggers, etc. Such clustering problems are often referred to as co-clustering or multi-way clustering in the literature. Long et al. [13] investigated the problem of co-clustering as a matrix factorization problem and derived multiplicative update formulas for identifying the clusters. Dhillon et al. [4] presented a framework for co-clustering that minimizes the loss in mutual information between the original joint distribution of related data and the corresponding joint distribution of the clustered data. The essence of different co-clustering algorithms has been captured in [12] where inter-related data types are represented as a k-partite graph and clusters are learned using Bregman's divergence as the objective function.

Other learning problems on multiple networks such as classification and ranking have recently attracted the attention of the research community. Kato et al. [8] developed a label propagation algorithm on multiple networks that automatically integrates structure information brought in by multiple networks. Ding et al.[22] has combined information from multiple graphs for document recommendations. Cheng et al.[3] also developed a recommendation algorithm based on performing a query dependent random walk on a k-partite graph. In summary, the past work can be categorized into clustering on single network, co-clustering k-partite graphs and learning problems such as classification and ranking on multiple networks. To the best of our knowledge, there has been no work on jointly identifying related communities using prior information from multiple heterogenous networks.

III. Preliminaries

For brevity, we consider a multi-network clustering problem for a given pair of networks. Our formulation can be extended to more than two networks.

Let $\mathcal{G}_1(V_1, E_1)$ and $\mathcal{G}_2(V_2, E_2)$ be a pair of graphs associated with two networks, where V_i is the set of nodes associated with graph $\mathcal{G}_i$ and $E_i \subseteq V_i \times V_i$ is the corresponding set of edges between the nodes. The objective of multi-network clustering is to create sets of partitions $\{\mathcal{P}_{1j}\}_{j=1}^{k_1}$ and $\{\mathcal{P}_{2j}\}_{j=1}^{k_2}$ such that $V_1 = \bigcup_{j=1}^{k_1} \mathcal{P}_{1j}$ and $V_2 = \bigcup_{j=1}^{k_2} \mathcal{P}_{2j}$. We seek a pair of functions $g_1 : V_1 \rightarrow [0,1]^{k_1}$ and $g_2 : V_2 \rightarrow [0,1]^{k_2}$ such that $g_i(v_j) = (c_1, c_2, ..., c_{k_i})$, where each $c_j \in [0,1]$ is the degree of membership node v_j belongs to cluster partition $\mathcal{P}_{ij}$.

In the independent clustering approach, the cluster membership functions g_1 and g_2 are learnt separately using their corresponding adjacency matrices. To simplify the notation, let A be the adjacency matrix associated with graph $\mathcal{G}_1$, where $A_{ij} = 1$ if $(v_i, v_j) \in E_1$. Similarly, B is the adjacency matrix for graph $\mathcal{G}_2$, where $B_{ij} = 1$ if $(v_i, v_j) \in E_2$. In addition, we assume there is a third set of edges $E_3 \subseteq V_1 \times V_2$ connecting the nodes between $\mathcal{G}_1$ and $\mathcal{G}_2$. We denote the adjacency matrix for these edges as C, i.e., $C_{ij} = 1$ if $(v_i, v_j) \in E_3$.

In this study, the cluster partitions are obtained by decomposing the adjacency matrix representation of a graph into a product of its latent factors. In particular, we seek to minimize the distance function $D(A\|B)$ between the adjacency matrix A and the product of latent factors B, where:

$$D(A\|B) = \sum_{ij} A_{ij} \log \left(\frac{A_{ij}}{B_{ij}} \right) - A_{ij} + B_{ij} \qquad (1)$$

Note that if $\sum_{ij} A_{ij} = \sum_{ij} B_{ij} = 1$, the distance function reduces to Kullback-Leibler divergence measure.

Depending on the application domain, the adjacency matrix C is either readily available as part of the data or needs to be estimated from the data. For example, consider the document-document network (A) and author-author network (B). Both networks are naturally linked by a document-author bipartite graph C. In another example, consider two networks constructed from Wikipedia editors and Digg users. Suppose we want to simultaneously cluster these two user networks such that each cluster represent users with interest in certain topics like science, sports, entertainment, etc. Here there is no natural link matrix C that is readily available to perform joint clustering. Nevertheless, it can be estimated from the data. Two users from different networks are linked if the Wikipedia pages one of them have edited has high similarity value to the news stories submitted by the Digg user.

IV. Proposed Framework

This section outlines our proposed framework for identifying cohesive subgroups in multiple networks. Our framework uses the non-negative matrix factorization technique given in [9], [5]. Let $A \in \mathcal{R}_+^{n \times n}$ and $B \in \mathcal{R}_+^{m \times m}$ be the adjacency matrices of the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, whereas $C \in \mathcal{R}_+^{n \times m}$ be the adjacency matrix for the links between nodes in $\mathcal{G}_1$ and $\mathcal{G}_2$. Here $\mathcal{R}_+$ represents set of non negative real numbers. Note that our framework is applicable to both directed and undirected graphs.

A. Joint Clustering of Multiple Networks

To simultaneously cluster the networks, we minimize the following objective function:

$$\mathcal{J} = D(A \parallel XUX^T) + D(B \parallel YWY^T) + D(C \parallel XVY^T)$$

where $X \in \Re_+^{N \times k_1}$ and $Y \in \Re_+^{M \times k_2}$ are the corresponding cluster membership matrices for the two networks. The decomposition of A into a 3-factor XUX^t instead of a 2-factor XX^t enables the framework to deal with directed links [23], [5]. The matrix V reflects the correspondence between the subgroups derived from the two networks.

The objective function can be written as follows:

$$\begin{aligned} \mathcal{J} &= \min_{XYUVW} \sum_{ij} A_{ij} \log \frac{A_{ij}}{[XUX^T]_{ij}} - A_{ij} + [XUX^T]_{ij} \\ &+ \sum_{ik} C_{ik} \log \frac{C_{ik}}{[XVY^T]_{ik}} - C_{ik} + [XVY^T]_{ik} \\ &+ \sum_{ks} B_{ks} \log \frac{B_{ks}}{[YWY^T]_{ks}} - B_{ks} + [YWY^T]_{ks} \qquad (2) \end{aligned}$$

Taking the partial derivatives of $\mathcal{J}$ with respect to X and Y yield the following(We omit the partial derivatives with respect to U, V and W due to lack of space)

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial X_{ij}} &= \sum_{a=1}^{N} \left[\frac{-A_{ia}[XU^T]_{aj}}{[XUX^T]_{ia}} + [XU^T]_{aj} - \frac{A_{ai}[XU]_{aj}}{[XUX^T]_{ai}} \right. \\ &+ \left. [XU]_{aj} \right] + \sum_{a=1}^{M} \left[\frac{-C_{ia}[YV^T]_{aj}}{[XVY^T]_{ia}} + [YV^T]_{aj} \right] \\ \frac{\partial \mathcal{J}}{\partial Y_{ij}} &= \sum_{a=1}^{M} \left[\frac{-B_{ia}[YW^T]_{aj}}{[YWY^T]_{ia}} + [YW^T]_{aj} - \frac{B_{ai}[YW]_{aj}}{[YWY^T]_{ai}} \right. \\ &+ \left. [YW]_{aj} \right] + \sum_{a=1}^{N} \left[\frac{-C_{ai}[XV]_{aj}}{[XVY^T]_{ai}} + [XV]_{aj} \right] \qquad (3) \end{aligned}$$

Given the objective function (2) and its partial derivatives, one can solve for X, Y, U, V and W using a gradient descent approach. Here, we give a converging iterative matrix factorization based update formulas for the unknown factors:

$$X_{ij} = X_{ij} \frac{\sum_{a=1}^{N} \left(\frac{A_{ia}[XU^T]_{aj}}{[XUX^T]_{ia}} + \frac{A_{ai}[XU]_{aj}}{[XUX^T]_{ai}} \right) + \sum_{a=1}^{M} \frac{C_{ia}[YV^T]_{aj}}{[XVY^T]_{ia}}}{\left(\sum_{a=1}^{N} [XU + XU^T]_{aj} + \sum_{a=1}^{M} [YV^T]_{aj} \right)} \qquad (4)$$

$$Y_{ij} = Y_{ij} \frac{\sum_{a=1}^{M} \left(\frac{B_{ia}[YW^T]_{aj}}{[YWY^T]_{ia}} + \frac{B_{ai}[YW]_{aj}}{[YWY^T]_{ai}} \right) + \sum_{a=1}^{N} \frac{C_{ai}[XV]_{aj}}{[XVY^T]_{ai}}}{\left(\sum_{a=1}^{M} [YW + YW^T]_{aj} + \sum_{a=1}^{N} [XV]_{aj} \right)} \qquad (5)$$

$$V_{ij} = V_{ij} \left[\frac{\sum_{a=1}^{N} \sum_{b=1}^{M} \frac{C_{ab}}{[XVY^T]_{ab}} X_{ai} Y_{bj}}{\sum_{a=1}^{N} \sum_{b=1}^{M} X_{ai} Y_{bj}} \right] \qquad (6)$$

$$U_{ij} = U_{ij} \left[\frac{\sum_{a=1}^{N} \sum_{b=1}^{N} \frac{A_{ab}}{[XUX^T]_{ab}} X_{ai} X_{bj}}{\sum_{a=1}^{N} \sum_{b=1}^{N} X_{ai} X_{bj}} \right] \qquad (7)$$

$$W_{ij} = W_{ij} \left[\frac{\sum_{a=1}^{M} \sum_{b=1}^{M} \frac{B_{ab}}{[YWY^T]_{ab}} Y_{ai} Y_{bj}}{\sum_{a=1}^{M} \sum_{b=1}^{M} Y_{ai} Y_{bj}} \right] \qquad (8)$$

The proofs of correctness and convergence of the update formulas are given in the Appendix section.

B. Incorporating Prior Information

Many times, we may have additional information about the correspondence between clusters in multiple networks. In what follows we give a motivation to incorporate this prior information into the objective function.

Example 1. *Consider a citation network between research articles and a co-authorship network between researchers. Suppose the articles are grouped into the following topics:* ***Algorithms, Artificial Intelligence, Databases, Cell Biology****, and* ***Genetics****. Typically, an author may work on multiple related topics, therefore the article partitions are not reflected as it is in the author network, rather they are further grouped into coarser clusters, namely,* ***Computer Science*** *and* ***Biotechnology****. The author cluster (****Computer Science****) is related to the first three article clusters, while* ***Biotechnology*** *is related to* ***Cell Biology*** *and* ***Genetics****. We expect such prior information will enhance the joint clustering results. This information can be encoded in a* 5×2 *prior matrix:*

$$P = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

where the rows are the article clusters and the columns are the author clusters.

To incorporate prior, we first need to interpret the role of the V matrix in the objective function. As mentioned earlier, V is the between network cluster correspondence matrix. The elements of V matrix reflect the relationship between the clusters between the two networks.

The user supplied prior information P can be incorporated into the objective function (2) as

$$\begin{aligned} \mathcal{J} &= D(A \parallel XUX^T) + D(B \parallel YWY^T) \\ &+ D(C \parallel X(\lambda V + (1-\lambda)P)Y^T) \end{aligned} \quad (9)$$

where λ is a parameter provided by the user. We call $V' = \lambda V + (1-\lambda)P$ as the adjusted correspondence matrix. The λ parameter in V' controls the tradeoff between fitting V' directly to the data and fitting V' to the prior matrix P. If $\lambda = 0$, then the correspondence between clusters is given by the prior matrix. If $\lambda = 1$, then the formulation reduces to the joint clustering framework given in Equation (2). In situations where the proportion of data scattered between the clusters in two networks is unknown, we can use a non-informative prior where P_{ij} is 0 or 1 indicating whether the i^{th} article cluster is related to j^{th} author cluster (see Example 1).

V. Experimental Evaluation

We have evaluated the effectiveness of proposed algorithm on both synthetic and real world network data.

Algorithm 1 Multi-network Clustering Algorithm

Input: Matrices A, B, C, and $MaxIter$.
Output: Matrices X, Y, and V.

Initialize X^{old}, Y^{old}, U^{old}, V^{old}, and W^{old} to to random matrices.
for $i = 1$ to $MaxIter$ **do**
  update X^{new} using (4)
  update Y^{new} using (5) with X^{new}
  update V^{new} using (6) with X^{new} and Y^{new}
  update U^{new} using (7) with X^{new}
  update W^{new} using (8) with Y^{new}
  set $X^{old} \leftarrow X^{new}$; $Y^{old} \leftarrow Y^{new}$; $U^{old} \leftarrow U^{new}$; $W^{old} \leftarrow W^{new}$; $V^{old} \leftarrow V^{new}$;
end for

A. Synthetic Dataset

Our synthetic dataset is generated as follows. First, the number of nodes and number of clusters (k) in each network are given. In our experiment, the number of clusters in each network is fixed to be 4 with 1000 data points in each cluster. Within each cluster i, a link is created between any two nodes with probability p_1. On the other hand, an inter-cluster link is created between a node in cluster i and nodes in other clusters in the network with probability p_2. In addition, links are also created between nodes from different networks. We create links between networks $\mathcal{G}_1$ and $\mathcal{G}_2$ with probabilities q_1 and q_2, where the former is the probability of link between corresponding clusters and the latter is the probability of noisy link between non-corresponding clusters. For example, if networks $\mathcal{G}_1$ and $\mathcal{G}_2$ have 4 clusters each, q_1 is probability of link between $cluster_i$ in network $\mathcal{G}_1$ and $cluster_i$ in network $\mathcal{G}_2$. q_2 is the probability of link between $cluster_i$ in $\mathcal{G}_1$ and $cluster_j$ in $\mathcal{G}_1$ with $i \neq j$.

B. Real-world data set

Real-world data differs significantly from the synthetic data set in that it has varying cluster size, varying proportion of intra and inter cluster links and noise. To study the performance of the algorithm on a real-life data set, we downloaded data from Wikipedia and digg, a popular online social bookmarking Web site.

1) Wikipedia Dataset: We use the Wikipedia dump from Oct-09-2009 for our experiments. We have chosen four topics as the ground truth clusters—Biology, Natural Science, Computer Science and Social Science. Each of the four topics are further divided into subtopics which are shown in Table I. We collected roughly 20K articles, with 5K articles in each category. After removing stubs and other smaller articles we were left with 10K articles and 53K editors (who have edited the articles). We removed articles/editors that do not have sufficient links (less than 3 links) with other articles/editors in our corpus. Our final data set contains

6403 articles and 5361 editors. A visual representation of the adjacency matrices of the article and editor networks is shown in Figure 2. Our goal of clustering is to identify the 12 sub-categories in the article network and relate them to 4 categories in the editor network.

Figure 2. Spy plot for Article (Left) and User (Right) networks in Wikipedia

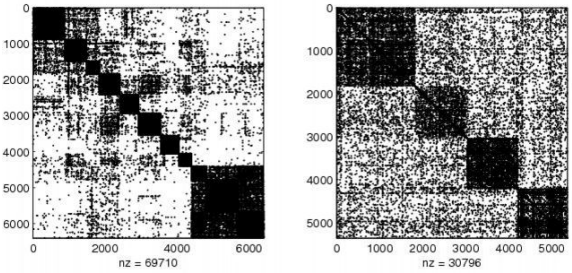


Table I
DATA CATEGORY AND SUB CATEGORY

Category User clusters	Sub-Categories Article clusters
Political Science	Civil-Rights Liberties(878); Imperialism(601); Nationalism(368);
Natural Science	Physics(568); Earth Sciences(513); Astronomy(613)
Computer Science	Algorithms(112); Operating Systems(395); Computer Architecture(350)
Biology	Zoology(392); Anatomy(897); Cell-Biology(716);

The Wikipedia data set is particularly challenging. Firstly, the editors do not seem to have a fixed domain of interest. As seen in the spy plot in Figure 2, a good proportion of editors have edited articles in all the four categories. We assign a ground truth label to each user based on the category for which the user has made the most number of contributions. Secondly, although each editor has his/her own Wikipedia page, many of these pages do not contain enough useful features that can be used to identify the cluster of an editor. Thirdly, the links between articles tend to be noisy. The article-article spy plot in Figure 2 shows 9 visually distinct groups even though there are 12 article clusters. This is because, in our sample, articles in **Algorithms** are highly connected to two other computer science topics.

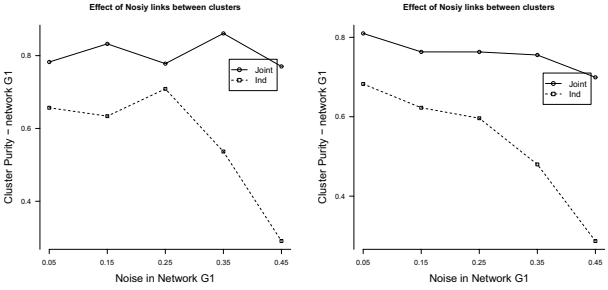
2) Digg Data set: The digg dataset is used in conjunction with a sample of the Wikipedia dataset to illustrate the applicability of our framework to multiple networks from different domains. We first sampled 1938 digg users who have bookmarked URLs on three topics: *Politics, Computer Science*, and *Natural Science*. Our sample contains only those users whose bookmarked URLs are mostly in one topic to avoid assigning multiple topics to each user. We formed a user-user adjacency matrix from the URL-user matrix. Two users are linked if they have at least τ URLs in common. Each URL bookmarked at digg.com has a title and a short description about the content of the web site. The digg url-word matrix and Wikipedia article-word matrix are then used to form a digg url - Wiki article matrix (G) where the entry G_{ij} is the cosine similarity between i^{th} URL and j^{th} wiki article. Note that we considered only those Wikipedia users who have edited articles in the three topics mentioned above.

C. Clustering Results

In this section we discuss the results of joint clustering on both synthetic and real-world data sets.

1) Experiments with Synthetic Data: Here we compare the performance of joint vs independent clustering by varying the inter-cluster link probability (p_2) in each network. These inter-cluster links can be interpreted as noisy links that confuse the clustering process. For network $\mathcal{G}_1$, we fix p_1 = 0.5 and vary p_2 from 0.05 to 0.45 with step size 0.1. The parameters for network $\mathcal{G}_2$ are set to p_1=0.6 and vary $p_2 \in \{0.25, 0.45\}$. We fix $q_1 = 0.6$ and $q_2 = 0.5$.For each step size, we ran the algorithm 20 times and report the average cluster purity for each network (see Figure 3). Clearly the joint clustering gives higher purity than independent clustering.

Figure 3. Effect of varying Inter-cluster Links in network G1. Left: Network G2(p_1 = 0.5 p_2 = 0.25), Right: Network G2(p_1 = 0.5, p_2=0.45)



We also investigated the effect of noisy links between two networks. The parameters for network $\mathcal{G}_1$ are ($p_1 = 0.6, p_2 = 0.25$) and network $\mathcal{G}_2$ are ($p_1 = 0.6, p_2 = 0.3$). We set the probability of establishing a link between two related clusters in the different networks as q_1=0.45. The probability of a noisy link between two clusters from the different networks (q_2) is varied from 0.05 to 0.45. As shown in Figure 4, at low and medium noise levels joint clustering significantly outperforms independent clustering. when the noise level is high the independent clustering is as good as joint clustering.

2) Wikipedia: Independent Vs Joint Clustering: We first clustered the article network and user network independently and used it as our baseline result. As shown in Table II, independent clustering gives a purity of 0.36 and 0.42 for articles and users respectively. There are smaller isolated sub clusters within each cluster. In other words, the network

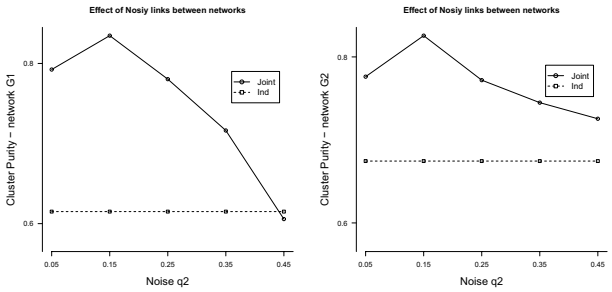


Figure 4. Effect of Varying Between Network Links (q_2)

data is such that each cluster is a forest with many connected components. There are no features to combine these connected components into one logical unit. On the other hand, the joint clustering gives a purity of 0.47 and 0.59 for articles and users respectively. This is because, the users act as intermediary features that join many of the sub clusters in the article network.

3) Wikipedia: Incorporating Prior: In this section we analyze the effect of incorporating the prior matrix into the model. Figure 5 shows the estimated cluster correspondence matrix $V' = \lambda V + (1 - \lambda)P$ for different λs. Notice that when $\lambda = 1$, we do not use any prior information and the algorithm wrongly associates the first user cluster with five article clusters (large V'_{ij}). Similarly, it has predominantly associated the user cluster 2 (Natural Sciences) with single article cluster (Physics). When $\lambda = 0$, the clustering results are purely based on the non-informative prior P. When $\lambda = 0.5$, the algorithm can still correctly identify the corresponding clusters from the two networks. In addition, the clustering results using prior (with $\lambda = 0.5$) gives higher purity for user clusters than without using prior ($\lambda = 1$).

Table II
AVERAGE CLUSTER PURITY ON WIKIPEDIA DATASET

Experiment	Article	User
Independent clustering	0.36	0.42
Joint clustering without Prior ($\lambda = 1$)	0.47	0.59
Joint clustering with Prior ($\lambda = 0.5$)	0.47	0.69

4) Wikipedia and Digg: Here, We linked the digg users with Wikipedia editors based on the similarity between words in the bookmarked URLs and words in the edited Wikipedia articles. There are three clusters in each network. Independent clustering of Wikipedia users gave an average purity of 0.43 whereas joint clustering gave an improved purity of 0.66. For Digg dataset, independent clustering gave a purity of 0.60 which marginally improved to 0.61.

VI. CONCLUSIONS

In this paper we have discussed the problem of learning cohesive subgroups and their correspondences in multiple related networks. Our experiments reveal that the joint clustering of multiple networks gives better results in terms of cluster purity. We have also introduced the idea of using a prior to guide the clustering process. Scalability of the formulation to networks having millions of nodes is a potential direction for future research. Towards this end, we plan to investigate online matrix factorization methods for joint partitioning of multiple related networks.

VII. ACKNOWLEDGEMENT

This material is based upon work supported in part by ONR grant number N00014-09-1-0663 and ARO grant number W911NF-09-I-0566.

REFERENCES

[1] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *KDD*, pages 509–514. ACM, 2004.

[2] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *Proceedings of the 30th Int'l Design Automation Conf*, pages 749–754. ACM, 1993.

[3] H. Cheng, P.-N. Tan, J. Sticklen, and W. F. Punch. Recommendation via query centered random walk on k-partite graph. In *Proceedings of the 2007 7th IEEE Int'l Conf on Data Mining*, pages 457–462. IEEE Computer Society, 2007.

[4] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the 9th ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining*, pages 89–98. ACM, 2003.

[5] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD Int'l Conf on Knowledge discovery and Data Mining*, pages 126–135. ACM, 2006.

[6] B. Hendrickson and R. W. Leland. A multi-level algorithm for partitioning graphs. In *Proceedings of Supercomputing '95*, 1995.

[7] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[8] T. Kato, H. Kashima, and M. Sugiyama. Robust label propagation on multiple networks. *Neural Networks, IEEE Transactions on*, 20(1):35–44, Jan. 2009.

[9] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *In Advances in Neural Information Processing Systems*, volume 13, pages 556–562, 2001.

[10] H. Li, Z. Nie, L. Wang-Chien, C. L. Giles, and J.-R. Wen. Scalable community discovery on textual data with relations. In *Proceedings of the 17th ACM Conf on Information and Knowledge Management*, pages 1203–1212, 2008.

	λ = 0				λ = 0.50				λ = 1			
	UC-1	UC-2	UC-3	UC-4	UC-1	UC-2	UC-3	UC-4	UC-1	UC-2	UC-3	UC-4
AC-1	1	0	0	0	0.95	0.06	0.10	0.06	1.04	0.16	0.15	0.18
AC-2	1	0	0	0	1.21	0.17	0.13	0.10	1.17	0.15	0.22	0.17
AC-3	1	0	0	0	0.83	0.47	0.08	0.05	0.79	0.09	0.13	0.42
AC-4	0	1	0	0	0.07	1.11	0.05	0.03	0.28	0.12	1.48	0.29
AC-5	0	1	0	0	0.04	0.91	0.05	0.02	1.37	0.11	0.18	0.16
AC-6	0	1	0	0	0.25	1.13	0.07	0.08	0.22	1.39	0.12	0.26
AC-7	0	0	1	0	0.20	0.15	1.05	0.21	0.10	0.05	1.14	0.09
AC-8	0	0	1	0	0.27	0.71	0.69	0.19	0.07	0.12	1.04	0.07
AC-9	0	0	1	0	0.30	0.16	0.82	0.14	0.06	0.07	0.84	0.18
AC-10	0	0	0	1	0.32	0.10	0.18	0.72	0.73	0.35	0.39	0.66
AC-11	0	0	0	1	0.11	0.07	0.15	1.20	0.37	0.42	0.27	1.13
AC-12	0	0	0	1	0.44	0.24	0.43	0.69	0.32	0.24	0.23	0.66

Figure 5. The value of cluster correspondence matrix $\lambda V + (1-\lambda)P$ for $\lambda = 0, 0.5, 1$ respectively on Wikipedia data set. **UC** refers to user cluster and **AC** refer to article cluster.

[11] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. MetaFac: community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining*, pages 527–536, 2009.

[12] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *KDD*, pages 317–326. ACM, 2006.

[13] B. Long, Z. M. Zhang, and P. S. Yu. Co-clustering by block value decomposition. In *Proceedings of the 11th ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining*, pages 635–640. ACM, 2005.

[14] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2003.

[15] J. Scripps and P.-N. Tan. Clustering in the presence of bridge nodes. In *Proceedings of the SIAM Int'l Conf on Data Mining*, Bethesda, MD, April 2006.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 200.

[17] W. Tang, Z. Lu, and I. Dhillon. Clustering with multiple graphs. In *Proceedings of the 9th IEEE Int'l Conf on Data Mining*, pages 1016–1021, Miami, FL, December 2009.

[18] C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining*, pages 717–726, 2007.

[19] F. Wei, W. Qian, C. Wang, and A. Zhou. Detecting overlapping community structures in networks. *World Wide Web*, 12(2):235–261, June 2009.

[20] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM Int'l Conf on Knowledge Discovery and Data Mining*, pages 927–936, 2009.

[21] Z. Zhang, T. Li, C. Ding, and X. Zhang. Binary matrix factorization with applications. In *Proceedings of the IEEE Int'l Conf on Data Mining*, pages 391–400, 2007.

[22] D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. L. Giles. Learning multiple graphs for document recommendations. In *Proceeding of the 17th Int'l Conf on World Wide Web*, pages 141–150. ACM, 2008.

[23] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th Annual Int'l ACM SIGIR Conf on Research and development in information retrieval*, pages 487–494. ACM, 2007.

VIII. Appendix-A

In this section we provide the formal proofs that show the update formula (4) - (8) monotonically decrease the objective function (2). The proofs can be shown through the use of the auxiliary functions.

Definition 1. *$G(w, \acute{w})$ is an auxiliary function for F(w) if following two conditions are satisfied*

$$G(w, \acute{w}) \geq F(w), \qquad G(w, w) = F(w) \tag{10}$$

Lemma 1. *If $G_1(w, \acute{w})$ and $G_2(w, \acute{w})$ are auxiliary functions for $F_1(w)$ and $F_2(w)$ respectively, then G = G_1 + G_2 is the auxiliary function for F = F_1 + F_2. Further, F is non decreasing under the update formula*

$$w^{t+1} = arg\min_{w} G(w, w^t) \tag{11}$$

Proof: The proof for G as an auxiliary function for F follows directly from definition. Below, we give the proof for second part.

$$\begin{aligned} F(w^{t+1}) &= F_1(w^{t+1}) + F_2(w^{t+1}) \\ &\leq G_1(w^{t+1}, w^t) + G_2(w^{t+1}, w^t) \\ &\leq G_1(w^t, w^t) + G_2(w^t, w^t) \\ &= F(w^t) \end{aligned}$$

The third line follows from the fact that w^{t+1} minimizes the auxiliary function G. Thus, $G(w^{t+1}, w^t) \leq G(w^t, w^t)$ and $F(w^{t+1}) \leq F(w^t)$, which completes the proof. ■

Since Equation (2) involves a summation of various distance functions $D(\cdot\|\cdot)$, Lemma 1 suggests that it is sufficient to show that minimizing the auxiliary function of the summation decreases the overall objective function to prove the correctness and convergence of our algorithm. Here we give the auxiliary function for the term $D(A\|XUX^T)$ which is quadratic in X and the auxiliary function for other terms can be similarly derived. The update formula for X which minimizes $D(A\|XUX^T)$ is given by

$$X_{ij} = X_{ij}\left(\frac{\sum_{a=1}^{N}\left(\frac{A_{ia}[XU^T]_{aj}}{[XUX^T]_{ia}} + \frac{A_{ai}[XU]_{aj}}{[XUX^T]_{ai}}\right)}{\left(\sum_{a=1}^{N}[XU + XU^T]_{aj}\right)}\right) \quad (12)$$

The objective function can be written as

$$F_1 = \sum_{i,j} A_{ij} \log \frac{A_{ij}}{\sum_{kl} X_{ik}U_{kl}X_{lj}^T} - A_{ij} + \sum_{kl} X_{ik}U_{kl}X_{lj}^T \quad (13)$$

Now define

$$\alpha_{k,l} = \frac{X_{ik}^{(t)}U_{kl}X_{lj}^{(t)T}}{\sum_{rs} X_{ir}^{(t)}U_{ks}X_{sj}^{(t)T}} \quad (14)$$

Clearly, $\sum_{rs} \alpha_{rs} = 1$. Using Jensens inequality we get

$$-\log \sum_{kl} X_{ik}U_{kl}X_{lj}^T \leq -\sum_{kl} \alpha_{kl} \frac{X_{ik}U_{kl}X_{lj}^T}{\alpha_{kl}} \quad (15)$$

Substituting (15) in (13) we get $D(A \parallel XUX^T) \leq$

$$\sum_{ij}\left[A_{ij}\log A_{ij} - A_{ij} - A_{ij}\sum_{kl}\alpha_{kl}\log\frac{X_{ik}U_{kl}X_{lj}^T}{\alpha_{kl}} + \sum_{kl} X_{ik}U_{kl}X_{lj}^T\right] \quad (16)$$

plugging in α_{kl} in above equation gives the following bound on the objective function

$$\sum_{ij}\left[A_{ij}\log A_{ij} - A_{ij} - A_{ij}\sum_{kl}\frac{X_{ik}^{(t)}U_{kl}X_{lj}^{(t)T}}{\sum_{rs}X_{ir}^{(t)}U_{rs}X_{sj}^{(t)T}}\left(\log X_{ik}U_{kl}X_{lj}^T - \log\frac{X_{ik}^{(t)}U_{kl}X_{lj}^{(t)T}}{\sum_{rs}X_{ir}^{(t)}U_{rs}X_{sj}^{(t)T}}\right) + \sum_{kl} X_{ik}U_{kl}X_{lj}^T\right]$$

which is the auxiliary function for F_1. We denote it as $G_1(X, X^{(t)})$. Now taking derivative of G_1 with respect to X_{pq} we get,

$$\frac{\partial G}{\partial X_{pq}} = -\sum_{jl} A_{pj}\frac{X_{pq}^{(t)}U_{ql}X_{lj}^{(t)T}}{\sum_{rs}X_{pr}^{(t)}U_{rs}X_{sj}^{(t)T}X_{pq}} + \sum_{jl} U_{ql}X_{lj}^{(t)T} - \sum_{ik} A_{ip}\frac{X_{ik}^{(t)}U_{kq}X_{qp}^{(t)T}}{\sum_{rs}X_{ir}^{(t)}U_{rs}X_{sp}^{(t)T}X_{pq}} + \sum_{ik} X_{ik}^{(t)}U_{kq} = 0$$

We get

$$X_{pq} = X_{pq}^{(t)}\left[\frac{\sum_j \frac{A_{pj}[UX^{(t)T}]_{qj}}{[X^{(t)}UX^{(t)T}]_{pj}} + \sum_i \frac{A_{ip}[X^{(t)}U]_{iq}}{[X^{(t)}UX^{(t)T}]_{ip}}}{\sum_{j=1}^{N}[UX^{(t)T}]_{qj} + \sum_{i=1}^{N}[X^{(t)}U]_{iq}}\right] \quad (17)$$

which is same as (12). Similarly, the term $D(C \parallel XVY^T)$ can be written as

$$F_2 = \sum_{ij} C_{ij}\log\frac{C_{ij}}{\sum_{kl}X_{ik}V_{kl}Y_{lj}^T} - C_{ik} + \sum_{kl} X_{ik}V_{kl}Y_{lj}^T$$

Now define

$$\beta_{kl} = \frac{X_{ik}^{(t)}V_{kl}Y_{lj}^{(t)T}}{\sum_{rs}X_{ir}^{(t)}V_{rs}Y_{sj}^{(t)T}} \quad (18)$$

Once again $\sum_{kl}\beta_{kl} = 1$ and following the same procedure as above, we get the auxiliary function for $D(C \parallel XVY^T)$ to be $G_2(X, X^{(t)}) =$

$$\sum_{ij}\left[C_{ij}\log C_{ij} - C_{ij} - C_{ij}\sum_{kl}\frac{X_{ik}^{(t)}V_{kl}Y_{lj}^{(t)T}}{\sum_{rs}X_{ir}^{(t)}V_{rs}Y_{sj}^{(t)T}}\log X_{ik}V_{kl}Y_{lj}^T - \log\frac{X_{ik}^{(t)}V_{kl}Y_{lj}^{(t)T}}{\sum_{rs}X_{ir}^{(t)}V_{rs}Y_{sj}^{(t)T}} + \sum_{kl} X_{ik}V_{kl}Y_{lj}^T\right]$$

Taking derivative of this with respect to X_{pq} and equating it to zero, we get

$$X_{pq} = X_{pq}^{(t)}\left(\frac{\sum_{i=1}^{M}\frac{C_{pi}[YV^T]_{iq}}{[XVY^T]_{pi}}}{\left(\sum_{i=1}^{M}[YV^T]_{iq}\right)}\right) \quad (19)$$

Minimizing the original objective function (2) with respect to X, we have

$$\begin{aligned}\min_X \mathcal{J}(X) &= \min_X F_1(X) + F_2(X) \\ &\leq \min_X G_1(X, X^{(t)}) + G_2(X, X^{(t)}) \quad (20)\end{aligned}$$

Taking derivative of $G_1(X, X^{(t)}) + G_2(X, X^{(t)})$ with respect to X and equating it to zero we get update formulae (4). By Lemma 1, this objective function monotonically decreases the objective function (15). Thus we have proved the following theorem

Theorem 1. *For fixed Y, U, V and W, the update formula for X, given in Equations* (4) *monotonically decreases the objective function in* (2)*.*

Similarly, for fixed X, U, V and W the update formulae for Y given in (5) monotonically decreases the objective function and the same is true for other three update formula. Thus we have shown that **Algorithm 1** monotonically decreases the loss function (2).