Clustering with Multiple Graphs

Wei Tang
Department of Computer Sciences
The University of Texas at Austin
Austin, U.S.A
wtang@cs.utexas.edu

Zhengdong Lu
Inst. for Computational Engineering & Sciences
The University of Texas at Austin
Austin, U.S.A
luz@cs.utexas.edu

Inderjit S. Dhillon
Department of Computer Sciences
The University of Texas at Austin
Austin, U.S.A
inderjit@cs.utexas.edu

Abstract—In graph-based learning models, entities are often represented as vertices in an undirected graph with weighted edges describing the relationships between entities. In many real-world applications, however, entities are often associated with relations of different types and/or from different sources, which can be well captured by multiple undirected graphs over the same set of vertices. How to exploit such multiple sources of information to make better inferences on entities remains an interesting open problem. In this paper, we focus on the problem of clustering the vertices based on multiple graphs in both unsupervised and semi-supervised settings. As one of our contributions, we propose Linked Matrix Factorization (**LMF**) as a novel way of fusing information from multiple graph sources. In **LMF**, each graph is approximated by matrix factorization with a graph-specific factor and a factor common to all graphs, where the common factor provides features for all vertices. Experiments on SIAM journal data show that (1) we can improve the clustering accuracy through fusing multiple sources of information with several models, and (2) **LMF** yields superior or competitive results compared to other graph-based clustering methods.

Keywords-clustering; multiple sources; graph; semi-supervised learning

I. INTRODUCTION

Relational data are ubiquitous, and the associated modeling and inference tasks have become important topics in both machine learning and data mining. The common tools modeling relational data often represent them as an undirected graph with vertices representing entities and (weighted or unweighted) edges describing the "relationships" between entities. In many application domains, these relationships are of different types and/or are obtained from different sources, which can be well represented by multiple undirected graphs over the same set of vertices with edges from different graphs capturing the heterogeneous relations. As one example of such multiple graphs, let us consider the proximity between researchers. Two researchers are considered to be similar if they have co-authored some papers, while it is also reasonable to assume two authors to have similar interest (probably to a lower level) if they both cited the same papers in their published work or they published in the same venues. These different type of relationships between authors naturally form multiple undirected graphs over the same set of authors. How to exploit the multiple sources

of information to make better inferences about entities and relationships is an interesting open problem.

In this paper, we consider the particular graph mining task of clustering vertices into several groups in the presence of multiple types of proximity relations. We give an extensive comparison of several graph-based clustering algorithms, as well as their semi-supervised extensions. One major contribution of this paper is a novel method for extracting common factors from multiple graphs, called Linked Matrix Factorization (LMF), based on which various clustering methods can naturally apply. Experiments on both synthetic and real-world data show the efficacy of the proposed methods in combining the information from multiple sources. In particular, LMF yields superior results compared to other graph-based clustering methods in both unsupervised and semi-supervised settings.

Road Map The remainder of this paper is organized as follows. Section II discusses the characteristics of the data and the inadequacy of clustering with individual graphs. Section III discusses the extension of unsupervised clustering methods to multiple graphs. Section IV is devoted to the formulation and optimization of Linked Matrix Factorization. Section V extends the unsupervised model in Section III to semi-supervised scenarios where constraints on the cluster assignments are known. Section VI reports the experimental results in both unsupervised and semi-supervised scenarios. Finally Section VII summarizes the paper and discusses related work and directions for future research.

II. DATA CHARACTERISTICS

We first motivate our work by discussing the clustering problem on some real-world data. In many scientific publication domains such as CiteSeer or arXiv, the relationships between documents can often be described as multiple graphs with different link types.

A. SIAM Journal Data Set

In this paper, we consider the data from eleven journals and proceedings for the period 1999-2004 published by the Society for Industrial and Applied Mathematics (SIAM). There are a total of 5022 articles in the data set, from which we generated two subsets:
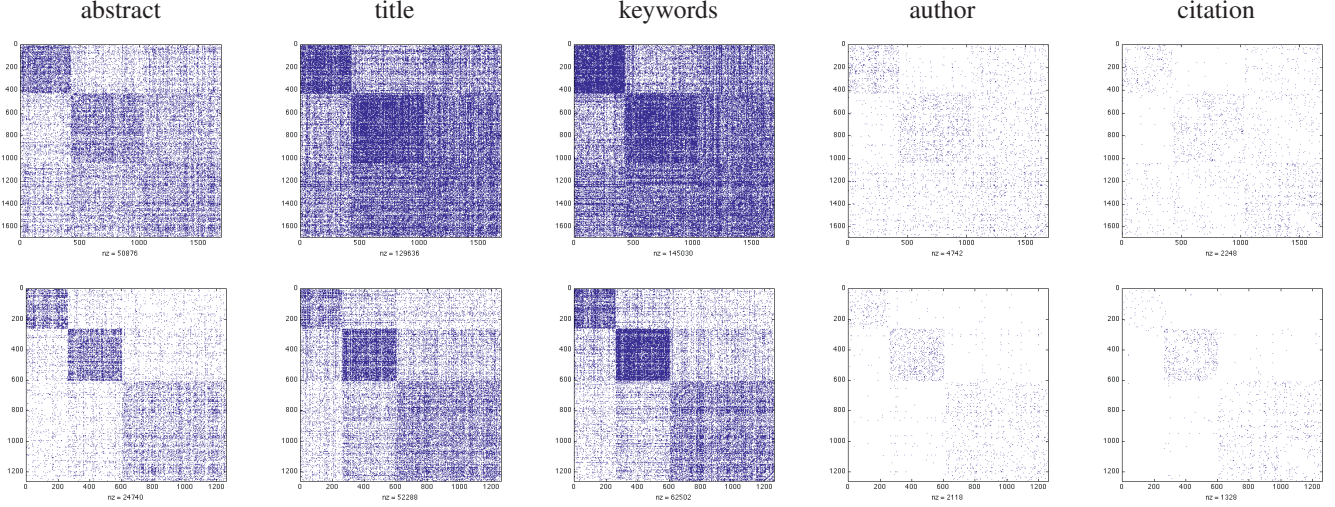
Figure 1. Spy plots of SIAM-similar data set (upper row) and SIAM-different data set (lower row). SIAM-different data set can be seen to be easier to cluster.

- SIAM-different: containing 1260 articles published in SIAM J DISCRETE MATH, SIAM J OPTIMIZ and SIAM J SCI COMPUT;
- SIAM-similar: containing 1690 articles published in SIAM J MATRIX ANAL A, SIAM J NUMER ANAL and SIAM J SCI COMPUT.

Our task is to discover the natural cluster structure of journals based on the document similarities extracted from different sources. Note that SIAM-different is composed of three journals from different research areas and hence is easier to cluster, whereas SIAM-similar contains three journals on highly related research topics and is more difficult to cluster.

In both subsets, we consider document similarities from five different sources. The first three are obtained from document-term matrices; in particular, each document can be represented as a vector of non-trivial words from different parts of the articles, namely abstract, title or keywords. We calculate the cosine similarity between each pair of documents within these different contexts to form the first three similarity matrices. The last two similarity matrices are obtained via the author and citation relations, respectively. Details about the five link types are described below:

- The abstract similarity matrix $A^{(1)}$ is constructed from the document-abstract matrix. $A_{ij}^{(1)}$ is the cosine similarity between the abstracts of documents i and j.
- The title similarity matrix $A^{(2)}$ is formed from the document-title matrix. $A_{ij}^{(2)}$ is the cosine similarity between the titles of documents i and j.
- The keyword similarity matrix $A^{(3)}$ is computed from the document-keyword matrix. $A_{ij}^{(3)}$ is the cosine similarity between the keywords of documents i and j.
- The author similarity matrix $A^{(4)}$ represents the number of common authors for each pair of documents.

- The citation similarity matrix $A^{(5)}$ has the citation relation between each pair of documents. $A_{ij}^{(5)} = A_{ji}^{(5)} = 1$ if there is citation between documents i and j, and 0 otherwise.

Figure 1 shows the "adjacency" matrices for documents composed from the five different sources, where we plot the presence of (non-zero) edges (called *spy plot*) with documents listed according to the intended clusters. Clearly, information from different sources show very different characteristics. For example, the co-authorship graph is usually much sparser than the proximity based on abstracts, but intuitively each co-authorship edge is more informative. Due to the extreme sparsity, some graphs alone do not contain complete information of the structure. Indeed, the co-author relationship shown in Figure 1 contains over 100 disconnected components, and is therefore unable to reveal the 3-cluster structure inherent in the data. It is useful but challenging to combine the distinct characteristics of different graphs–for example, the sparse but informative relations as well as abundant but less informative ones.

B. Clustering with Individual Graphs

We adopt *Normalized Mutual Information* (NMI) to measure the clustering performance. Figure 1 shows spy plots for all the five adjacency matrices belonging to the SIAM-different and SIAM-similar data sets, with documents being aligned to their published journals. Clearly each graph contains certain information about the relationships between documents. If we apply spectral clustering [1] on each individual graph, we get the clustering results shown in Table I in terms of NMI. It is clear from Figure 1 and Table I that although the edges in the last two graphs are highly consistent with the cluster structure of journals, they do not contain enough information to recover the clusters

alone. As we will show in later sections, combining all the graphs, especially with our proposed LMF model, can yield significantly improved clustering results.

	SIAM-different	SIAM-similar
abstract	0.5893	0.2037
title	0.0324	0.2021
keywords	0.3731	0.2502
author	0.0042	0.0017
citation	0.0211	0.0078

III. UNSUPERVISED CLUSTERING MODELS

Let us take one step back and consider the more general problem of clustering with multiple graphs. Suppose we are given M undirected graphs whose adjacency matrices are $A^{(m)}, m = 1, 2, \cdots, M$, each of size $N \times N$, with vertices in all graphs corresponding to the same entities. We intend to find a clustering of the vertices based on the information from multiple sources.

Besides clustering each graph individually, we also have the following baseline models for combining the information from multiple graphs.

Summation of Graphs We find a combined adjacency matrix $A = \sum_{m=1}^{M} A^{(m)}$. With this new adjacency matrix A, we can perform spectral partitioning which can be achieved by computing the smallest eigenvectors of the graph Laplacian $L = D - A$, where D is the diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$. The use of eigenvectors can also be motivated as minimizing the "roughness" of vector $\mathbf{f} = [f_1, \cdots, f_N]^T$ over all the graphs:

$$\mathcal{G} = \sum_{m=1}^{M} \mathbf{f}^T L^{(m)} \mathbf{f} = \sum_{m=1}^{M} \sum_{i,j=1}^{N} A_{ij}^{(m)} (f_i - f_j)^2, \quad (1)$$

where $L^{(m)}$ is the graph Laplacian for the mth graph. Alternatively, we can use the normalized adjacency matrix, $\tilde{A}^{(m)} = (D^{(m)})^{-1/2} A^{(m)} (D^{(m)})^{-1/2}$.

Summation of Spectral Kernels We first construct spectral kernels for each graph, i.e., kernel $K^{(m)}$ based on the eigenspectrum of the graph Laplacian $L^{(m)}$, and then use the summation $K = \sum_{m=1}^{M} K^{(m)}$. as the kernel summarizing all graphs. One particular example (called step-function kernel in [2]) is the model

$$K^{(m)} = \sum_{k=1}^{d} \mathbf{v}_k^{(m)} (\mathbf{v}_k^{(m)})^T$$

where $\mathbf{v}_k^{(m)}$ is the kth smallest eigenvector of graph Laplacian $L^{(m)}$ and $d \ll N$ is the number of eigenvectors used per individual graph. Clustering can then be obtained by performing kernel K-means on kernel K. Other choices of $K^{(m)}$ include the heat diffusion kernel and regularized inverse of graph Laplacian[2], but the discussion of them is omitted here due to their inferior performance on our task.

Consensus Clustering Consensus clustering reconciles clustering results about the same data set coming from different sources. In this paper we follow the models in [3], where three consensus clustering algorithms are proposed: Cluster-based Similarity Partitioning Algorithm, HyperGraph Partitioning Algorithm, and Meta-Clustering Algorithm. In our experiments (Section VI), we only report the best result from these three methods.

IV. LINKED MATRIX FACTORIZATION

One major limitation of the baseline models is that they treat all graphs on an equal basis, and therefore cannot discriminate the informative sources and uninformative or noisy ones. A more sensible alternative is to extract the structure information shared by all the sources, and hence filter out irrelevant information or noise. Here we present Linked Matrix Factorization (LMF), a novel model for finding the common factor for all graphs .

A. Model

One natural model for unsupervised graph clustering is to approximate the given graph through a low-rank matrix factorization $A \approx P \Lambda P^T$, where P is an $N \times d$ matrix and Λ is an $d \times d$ symmetric matrix. Since we are given multiple graphs and the underlying entities are shared among graphs, a common factor matrix is desirable to link the multiple matrix factorizations together. Therefore, the objective of clustering over multiple graphs by matrix factorization can be formulated as minimizing

$$\mathcal{G} = \frac{1}{2} \sum_{i=1}^{M} \|A^{(m)} - P\Lambda^{(m)} P^T\|_F^2 + \frac{\alpha}{2} \left(\sum_{m=1}^{M} \|\Lambda^{(m)}\|_F^2 + \|P\|_F^2 \right), \quad (2)$$

where matrix P is the common factor shared among graphs, $\Lambda^{(m)}$ captures the characteristics of each graph (note that we do not constrain $\Lambda^{(m)}$ to be diagonal), $\| \cdot \|_F$ denotes the Frobenius norm and α is the regularization parameter. Matrix P can be regarded as a low dimensional embedding of entities characterized by multiple graphs, the differences being captured by $\Lambda^{(m)}$. The regularization terms on both P and $\Lambda^{(m)}$ are added to improve numerical stability and to avoid overfitting. In addition to the generic form given in (2), there are several possible alternative modeling choices. For example, instead of using the squared Frobenius norm, we could choose the relative entropy or other divergence measures for comparing $A^{(m)}$ and $P\Lambda^{(m)} P^T$. If the graphs were not symmetric, we could instead model each graph as $P\Lambda^{(m)} Q^T$. One could also enforce the columns of P to be orthonormal and drop the regularization term. However, in this paper, we will only focus on the case where each $A^{(m)}$ is an undirected symmetric graph and the approximation error is measured by the squared Frobenius norm.

B. Optimization

Note that the solutions to LMF are not unique. For instance, let matrices P^* and $\Lambda^{(m)*}$ $(i = 1, \ldots, M)$ be

the solutions to the optimization problem (2), then for any orthogonal matrix $R \in \mathbb{R}^{d \times d}$ (R only needs to be non-singular if there is no regularization term), the matrices $P^* R$ and $R^{-1} \Lambda^{(m)^*} R^{-1}$ will also be solutions. Moreover, the objective function is not jointly convex in P and $\Lambda^{(m)}$. Hence, we adopt an effective alternating minimization algorithm to find a locally optimal solution to LMF. First, matrix P is optimized while fixing each $\Lambda^{(m)}$; then, each matrix $\Lambda^{(m)}$ is optimized while fixing matrix P. This procedure is repeated until convergence. In optimizing matrix P and each $\Lambda^{(m)}$, we apply a quasi-Newton method, Limited memory BFGS (L-BFGS) [4], to optimize each factor in the inner loop.

The bottleneck in the L-BFGS algorithm is the evaluation of the objective in (2) and its gradient with respect to P and each $\Lambda^{(m)}$, respectively. Taking the derivative of (2) with respect to P and $\Lambda^{(m)}$ yields

$$\frac{\partial \mathcal{G}}{\partial P} = -2 \sum_{i=1}^{M} (A^{(m)} - P \Lambda^{(m)} P^T) P \Lambda^{(m)} + \alpha P, \quad (3)$$

$$\frac{\partial \mathcal{G}}{\partial \Lambda^{(m)}} = -P^T (A^{(m)} - P \Lambda^{(m)} P^T) P + \alpha \Lambda^{(m)}. \quad (4)$$

One can gain computational efficiency by taking advantage of the sparsity of $A^{(m)}$. In particular, the first term of the objective in (2) can be rewritten as:

$$\mathcal{G}' = \frac{1}{2} \sum_{m=1}^{M} \|A^{(m)}\|_F^2 - 2 \operatorname{Tr}(\Lambda^{(m)} P^T A^{(m)} P)$$
$$+ \operatorname{Tr}(P^T P \Lambda^{(m)} P^T P \Lambda^{(m)}), \quad (5)$$

which can be evaluated in $O(d(nnz + Nd))$ time for each graph (nnz represents the number of nonzero entries averaged over all graphs). Similarly, computing the gradient in (3) and (4) takes $O(d(nnz + Nd))$ time for each graph by utilizing the sparsity of $A^{(m)}$. Since the evaluation of the objective and its gradient share some computational steps, we can actually compute them at the same time within one loop over the multiple graphs. The total time complexity is $O(Md(nnz + Nd))$.

V. SEMI-SUPERVISED CLUSTERING MODELS

In many real-world applications, we often do not realistically expect the clustering to discover intended structure in a total unsupervised fashion. In those situations, we can often benefit from various types of weak supervision or side information. Here we consider the following two types of instance-level constraints on cluster assignments, which can naturally emerge in various situations [5]: *must-link*, which constrains entity i and entity j in the *same* cluster and *cannot-link*, which constrains entity i and entity j in *different* clusters.

There is a large body of work on using these *pairwise constraints* to boost the performance of clustering algorithms (see [5] for more details), but it has never been previously used in the context of combining multiple graph relations.

We consider two ways to incorporate these pairwise constraints into the clustering algorithm, both of which require viewing the unsupervised learning methods described in Sections III and IV as means of feature extraction for vertices. This applies explicitly for LMF (the features for vertex i are given by the ith row of P), and implicitly for spectral kernel cases where the obtained kernel can be viewed as the inner product of feature vectors. The first semi-supervised method is metric learning, which directly adapts the distance metric in the corresponding feature space to fit the given pairwise constraints. The second method is to express the pairwise constraints as a penalty term in the unsupervised learning objective function, based on which the feature vectors for clustering are learned.

Metric Learning Metric learning seeks a distance metric in feature space that fits our clustering or classification preference [6], [7]. The problem can be formulated as learning a squared Mahalanobis distance or equivalently the positive definite matrix, which satisfies the must-link and cannot-link constraints. Among the metric learning models, we consider Information-Theoretic Metric Learning (ITML) [6] due to its scalability to handle millions of pairwise constraints. Due to the space limit, we omit the description of ITML. ITML can serve as a post-processing step for the unsupervised learning in Section III to incorporate the pairwise constraints.

Semi-supervised Feature Extraction One limitation of the metric learning as a post-processing step is that it is often futile to correct the learned bad feature. One can often alleviate this by learning more "discriminative" features with the semi-supervision from the pairwise constraints. We consider the following objective function for feature $F \equiv \{\mathbf{f}_1, \cdots, \mathbf{f}_N\}$,

$$\mathcal{L}(F) = e(F, \{A^{(1)}, \cdots, A^{(M)}\}) + \gamma s(F; \mathcal{M}, \mathcal{C}), \quad (6)$$

where $e(F, \{A^{(1)}, \cdots, A^{(M)}\})$ stands for the "empirical error" term from unsupervised learning described in Section III (called *unsupervised term*), $s(F; \mathcal{M}, \mathcal{C})$ stands for the extra penalty term from given pairwise constraints (called *supervised term*), and the parameter γ controls the balance between the two terms. The unsupervised term could either be the objective function for LMF as in (2), or the measurement of roughness associated with the (combined) graph Laplacian as in (1). The supervised term is designed to ensure that the features of must-linked pairs are close and the features of cannot-linked pairs are far away from each other,

$$s(F; \mathcal{M}, \mathcal{C}) = \sum_{(i,j) \in \mathcal{M}} \|\mathbf{f}_i - \mathbf{f}_j\|^2 - \sum_{(i,j) \in \mathcal{C}} \|\mathbf{f}_i - \mathbf{f}_j\|^2.$$

For the LMF model, the optimization for the semi-supervised objective can be performed using the same optimization routine with minimal change. For the graph spectral methods based on a graph Laplacian, much like the unsupervised

case, the combined objective function can be recast into a eigenvalue problem.

VI. Experiments

In this section, we present results of clustering with multiple graphs on both synthetic and SIAM journal data. As we will show in the results, 1) we can improve the clustering performance by simultaneously modeling multiple sources of information, and 2) the pairwise constraints help in the semi-supervised learning scenario.

To evaluate the performance of LMF in the unsupervised setting, we compare it with several baseline methods introduced in Section III: 1)SpecC: spectral clustering algorithm [1] on single graph; 2)mSpC-A: spectral clustering algorithm on the sum of adjacency matrices; 3) mSpC-B: spectral clustering algorithm on the sum of normalized adjacency matrices; 4)SpecK: sum of spectral kernels from each graph; 5)Consensus: consensus clustering with SpecC as the base component.

A. SIAM Data: Unsupervised Clustering

The results of unsupervised clustering on SIAM-similar and SIAM-different are presented in Table II, with each column representing a different combination of individual graphs. The results are consistent with our observation that SIAM-similar is a much harder problem than SIAM-different. We observe that LMF is the only method that can consistently benefit from including more graphs. LMF leads with 7 out of 8 graph combinations on both SIAM-different and SIAM-similar. Most importantly, the best performance from all models (NMI = 0.714) is achieved by LMF with all five graphs.

In the above experiments, we set the parameters for LMF rather arbitrarily. Nevertheless it is meaningful to examine the impact of rank and regularization on the clustering performance based on LMF. The result is omitted due to space limit.

B. SIAM: Semi-supervised Clustering

In this section, we discuss semi-supervised clustering results on both SIAM data sets with pairwise constraints. As we will show, metric learning and semi-supervised feature extraction demonstrate different behavior, while both of them help in improving the clustering performance.
Metric Learning We considered kernelized metric learning with the following choices of initial kernel K_0: 1) SpC-ML: spectral kernel based on the sum of adjacency matrices, 2) SpK-ML: sum of spectral kernels based on individual graphs, 3) LMF-ML(L): linear kernel PP^T based on LMF feature P, and 4) LMF-ML(G): Gaussian kernel with rows of P as feature vectors: $K_0(i,j) = \exp(-\|p_i - p_j\|^2/\sigma^2)$. Figure 3 (upper row) presents the results of metric learning on four different kernels with the number of randomly selected constraints varying from 0 to 3000. As seen from the figure,

SIAM-different			
	$\{1,2,3\}$	$\{1,2,3,4\}$	$\{1,2,3,5\}$	$\{1{-}5\}$
mSpC-A	**0.657**	0.649	0.621	0.630
mSpC-B	0.626	0.683	0.684	0.701
SpecK	0.636	0.455	0.637	0.638
Consensus	0.587	0.597	0.559	0.444
LMF	0.611	**0.698**	**0.689**	**0.714**

SIAM-similar			
	$\{1,2,3\}$	$\{1,2,3,4\}$	$\{1,2,3,5\}$	$\{1{-}5\}$
mSpC-A	0.226	0.238	0.229	0.238
mSpC-B	0.244	0.235	0.238	0.234
SpecK	0.237	0.240	0.237	0.237
Consensus	0.212	0.202	0.186	0.151
LMF	**0.246**	**0.249**	**0.251**	**0.253**

SIAM-different

mSpC–A			mSpC–B			SpecK		
245	8	7	242	7	11	0	231	29
30	593	33	17	603	36	0	606	50
57	6	281	19	6	319	22	23	299

Consensus			LMF		
17	11	232	**241**	8	11
6	442	208	13	**614**	29
18	4	322	21	8	**315**

SIAM-similar

mSpC–A			mSpC–B			SpecK		
268	69	274	289	74	248	319	74	218
3	405	15	7	405	11	14	404	5
176	249	231	211	224	221	239	206	211

Consensus			LMF		
262	68	281	**399**	39	173
50	354	19	2	**366**	55
257	164	235	219	176	**261**

Figure 2. The confusion matrices of competing methods on the SIAM data set, where rows represent the actual classes and columns the clusters.

the three linear kernels (SpC-ML, SpK-ML, and LMF-ML(L)) do not respond well to the pairwise constraints, while the Gaussian kernel based on LMF feature (LMF-ML(G))(interestingly, the Gaussian kernel gives a much poorer result in the unsupervised case, as seen in the figure), can lead to superior performance after 2000 constraints. This behavior can be explained by the fact that the linear kernels, although informative by themselves have low rank and hence do not provide enough modeling flexibility for ITML, while the Gaussian kernel gives enough room for ITML to improve performance.
Semi-supervised Feature Extraction In this section, we report experimental results conducted to investigate the effectiveness of Semi-supervised Feature Extraction. As discussed in Section V, we can have two semi-supervised feature extraction algorithms based on their unsupervised counterparts: 1)LMF-SSFE: feature extraction based on
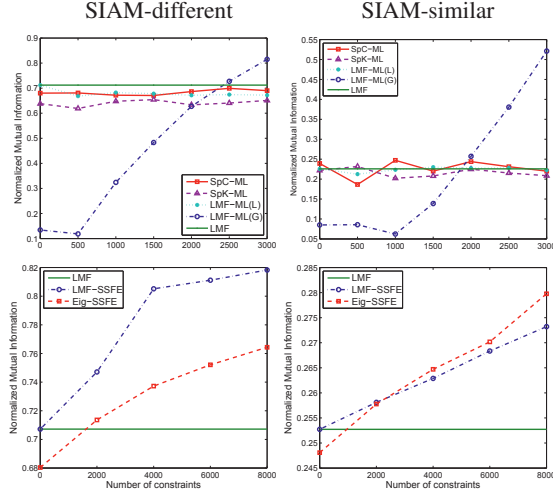
Figure 3. Semi-supervised clustering result on SIAM data. Upper row: metric learning. Lower row: feature extraction

LMF, and 2)Eig-SSFE: based on the eigenvectors of graph Laplacian, as in (1). Figure 3 (lower row) plots the clustering performance, measured in NMI, as a function of the number of constraints on SIAM-different and SIAM-similar data sets. As indicated by Figure 3 (lower row), both algorithms can benefit from pairwise constraints, but LMF-SSFE does significantly better on SIAM-different and performs comparably to Eig-SSFE on SIAM-similar.

VII. CONCLUSION AND DISCUSSION

In this paper, we discussed the general problem of clustering based on multiple similarity graphs in both unsupervised and semi-supervised settings. We extend several graph-based clustering methods to handle multiple graphs. As one of our major contribution, we proposed Linked Matrix Factorization (LMF) as a novel method for learning the characteristics common to all given graphs. Experiments show that: 1) in an unsupervised setting, LMF can effectively extract informative and reliable features for vertices, and yield better clustering performance than single graph methods and other graph-based models for combining multiple graphs, and 2) LMF, as a feature extraction model, responds fairly well to pairwise constraints.

Our work is interestingly related to multiple-view learning, where each object is described from more than one from multiple aspects ("views"). Most models in multiple-view learning are for supervised or semi-supervised classification [8], [9], where a combined view is achieved through forcing the classification function on different views to be close. Another direction is proposed by Chaudhuri et al [10] where multi-view clustering is performed on features extracted with canonical correlation analysis (CCA) between views. In both cases, it is required that each view already contain complete information for classification. Other related work includes extension of normalized cut to multiple graphs [11].

Despite the encouraging results of proposed model, several questions remained unanswered. One of the questions is how to incorporate the reliability of different sources into the clustering model. Another direction, as suggested in [10], is to explicitly model the common characteristics of all sources through canonical correlation analysis of the feature vectors, which is in the same spirit as LMF. It will be interesting and useful to suitably extend these methods to the problem of multiple graphs, which will be one of the directions of our future research.

VIII. ACKNOWLEDGEMENT

REFERENCES

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 1997.

[2] A. Smola and R. Kondor, "Kernels and regularization on graphs," in *Proc. of COLT'03*, 2003.

[3] A. Strehl and J. Ghosh, "Cluster ensemble - a knowledge reuse framwork for combining multiple partitions," *JMLR*, vol. 3, pp. 583–617, 2003.

[4] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[5] S. Basu, I. Davidson, and K. Wagstaff, *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 2008.

[6] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon, "Information-theoretic metric learning," in *Proc. of ICML'07*, 2007.

[7] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Proc. of NIPS 15*, vol. 15, 2003.

[8] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. of COLT'98*, 1998.

[9] V. Sindhwani and D. S. Rosenberg, "An rkhs for multi-view learning and manifold co-regularization," in *Proc. of ICML'08*, 2008, pp. 976–983.

[10] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. of ICML'09*, 2009.

[11] D. Zhou and C. Burges, "Spectral clustering and transductive learning with multiple views," in *Proc. of ICML'07*, 2007.