Semi-Supervised Clustering via Matrix Factorization*

Fei Wang† Tao Li‡ Changshui Zhang§

January 22, 2008

Abstract

The recent years have witnessed a surge of interests of semi-supervised clustering methods, which aim to cluster the data set under the guidance of some supervisory information. Usually those supervisory information takes the form of pairwise constraints that indicate the similarity/dissimilarity between the two points. In this paper, we propose a novel matrix factorization based approach for semi-supervised clustering. In addition, we extend our algorithm to co-cluster the data sets of different types with constraints. Finally the experiments on UCI data sets and real world *Bulletin Board Systems (BBS)* data sets show the superiority of our proposed method.

1 Introduction

Clustering, which aims to efficiently organize the data set, is an old problem in machine learning and data mining community. Most of the traditional clustering algorithms aim at clustering *homogeneous data*, *i.e.* the data points are all of a single type. However, in many real world applications, the data set to be analyzed involves more than one type. For example, *words* and *documents* in *document analysis*, *users* and *items* in *collaborative filtering*, *experimental conditions* and *genes* in *microarray data analysis*. The challenge is that the different types of data points are *not independent* of each other, on the contrary, usually there exist close relationships between different types of data, and it is difficult for the traditional clustering algorithms to utilize those relationship information efficiently.

Consequently, *co-clustering* techniques, which aim to cluster different types of data simultaneously by making efficient use of the relationship information, are proposed. For instance, *Dhillon* [8] proposed a Bipartite Spectral Graph Partitioning approach to co-cluster words and documents, *Cho. et al* [6] proposed to co-cluster the experimental conditions and genes for microarray data by minimizing the *Sum-Squared Residue*, *Long et al.* [20] proposed a general principled model, called *Relation Summary Network*, to co-cluster the heterogeneous data on a *k-partite graph*.

Despite their successful empirical results and rigorous theoretical analysis, these co-clustering algorithms only make use of inter-type relationship information. However, in many applications, we also have some intra-type data information. For example, in the typical user-movie rating problem, we usually have a database which not only contains ratings (i.e., relations between users and movies), but also contains user entities with user attributes (e.g., age, gender, education), movie entities with movie attributes (e.g., year, genre, director). Therefore how to effectively combine all those information to guide the process of clustering is a problem that is definitely worthy of researching.

One intuitive way for incorporating the intra-type data information is to ask some experts to label some data points of different types based on their attributes. These labeled points are then used as seeds to further guide or correct the co-clustering algorithms which are purely based on analyzing the inter-type relationship matrices or tensors. However, the problems with this approach are: (1) the labeling process is expensive and time-consuming; (2) sometimes it is hard to give an explicit label set for each type of data points. Taking the user-movie rating problem as an example, we do not know how many classes we should categorize the movies into, and how to define the labels of the user classes.

Based on the above considerations, in this paper, we propose to represent the intra-type information as constraints to guide the clustering process. Particularly, we consider the following two types of constraints.

- *must-link*—the two data points must belong to the same class;
- *cannot-link*—the two data points cannot belong to the same class.

In general it is much easier for someone to give such constraints based on the data attributes (one can refer to figure 1 as an example).

*The work of Fei Wang and Changshui Zhang is funded by China Natural Science Foundation No.60675009. The work of Tao Li is partially supported by NSF IIS-0546280.

†Department of Automation, Tsinghua University.
‡School of Computer Science, Florida International University.
§Department of Automation, Tsinghua University.

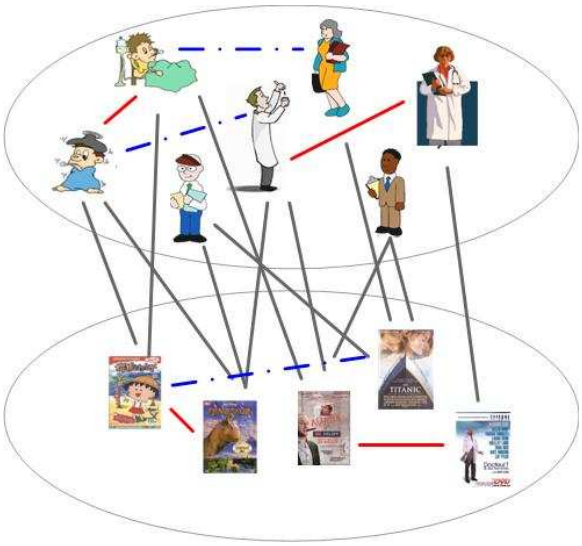


Figure 1: An example of the inter-type relationships and intra-type constraints. It is easy for users to judge whether the two movies belong to the same class by their contents, titles, or actors. Similarly, it is also not hard to judge whether the two person belong to the same class by their ages, jobs, or hobbies. In this figure, the red lines stand for the must-links, and the blue dashed lines represent the cannot-links.

Given the inter-type relationship information and intra-type relationship constraints, we propose a general *constrained co-clustering* framework to cluster the multiple type data points simultaneously. We show that the traditional semi-supervised clustering methods [15] are special cases of our framework when the data set is of only one single type. Finally the experimental results on several real world data sets are presented to show the effectiveness of our method.

The rest of this paper is organized as follows. In section 2 we introduce our *Penalized Matrix Factorization* (*PMF*) algorithm for constrained clustering. In section 3 and 4 we generalize our *PMF* based method to co-cluster dyadic and multi-type data sets with constraints. The experimental results are illustrated in section 5, followed by the conclusions and discussions in section 6.

2 Semi-Supervised Clustering Using Penalized Matrix Factorization

In this section we introduce our *penalized matrix factorization* (*PMF*) for semi-supervised co-clustering. First we introduce the notations that will be frequently used in the rest of this paper.

Table 1: Some frequently used notations

n	The number of data points
C	The number of clusters
$\mathbf{x}_i$	The i-th data point in $\mathbb{R}^d$
$\mathbf{X}$	The data matrix of size $d \times n$
$\mathbf{f}_c$	The cluster center of the c-th cluster
$\mathbf{F}$	The cluster center matrix
$\mathbf{G}$	The cluster indicator matrix
$\mathbf{G}_i$	The cluster indicator matrix of the i-th type data
$\mathbf{\Theta}$	The constraint matrix
$\mathbf{\Theta}_i$	The constraint matrix on the i-th type data
$\mathbf{R}$	The relationship matrix
$\mathbf{R}_{ij}$	The relationship matrix between the i-th and j-th types of data

2.1 Notations Throughout the paper, we use bold uppercase characters to denote matrices, bold lowercase characters to denote vectors. The meanings of some frequently used notations are summarized in table 1.

2.2 Problem Formulation In this subsection, we first review the basic problem of constrained clustering and then introduce a novel algorithm called *Penalized Matrix Factorization* (*PMF*) to solve it.

Given a data set $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, the goal of *clustering* is to partition the data set into C clusters $\pi = \{\pi_1, \pi_2, \cdots, \pi_C\}$ according to some principles. For example, the classical *kmeans* algorithm achieves this goal by minimizing the following cost function

$$J_{km} = \sum_{c} \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2,$$

where $\mathbf{f}_c$ is the center of cluster π_c. If we define three matrices $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$, $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_C] \in \mathbb{R}^{n \times C}$, $\mathbf{G} \in \mathbb{R}^{n \times C}$ with its (i,j)-th entry $\mathbf{G}_{ij} = g_{ij}$, where

$$g_{ij} = \begin{cases} 1, & if\ \mathbf{x}_i \in \pi_j \\ 0, & otherwise \end{cases} \tag{2.1}$$

then we can rewrite J_{km} in the following matrix form

$$J_{km} = \left\| \mathbf{X} - \mathbf{F}\mathbf{G}^T \right\|_F^2 \tag{2.2}$$

where $\|\cdot\|_F$ denotes the matrix *Frobenius norm*. Therefore, the goal of *kmeans* is to solve $\mathbf{G}$ by minimizing J_{km}, which can be carried out by *matrix factorization* techniques after some relaxations [12][11].

However, the organization of the data set in such a purely unsupervised way usually makes the results

unreliable since there is not any guidance from the data labels. Therefore, in recent years some researchers have proposed *semi-supervised clustering* algorithms [4][15], which aim to cluster $\mathcal{X}$ into C clusters under the guidance of some prior knowledge on the data labels. One type of such prior knowledge assumes that only part (usually a limited part) of the training data are labeled [3], while the other type of prior knowledge is even weaker in that it only assumes the existence of some pairwise constraints indicating similarity or dissimilarity relationships between training examples [4]. In this paper, we will consider the prior knowledge in the latter case.

Typically, the knowledge that indicates two points belong to the same class is referred to as *must-link constraints* $\mathcal{M}$, and the knowledge that indicates two points belong to different classes is referred to as *cannot-link constraints* $\mathcal{C}$. This type of information can be incorporated into traditional partitional clustering algorithms by adapting the objective function to include penalties for violated constraints. For instance, the *Pairwise Constrained KMeans* (*PCKM*) algorithm proposed in [4] modifies the standard sum of squared errors function in traditional *kmeans* to take into account both object-centroid distortions in a clustering $\pi = \{\pi_1, \pi_2, \cdots, \pi_C\}$ and any associated constraint violations, *i.e.*

$$J(\pi) = \sum_{c} \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2 + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M} \\ s.t.\ l_i \neq l_j}} \theta_{ij} + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C} \\ s.t.\ l_i = l_j}} \tilde{\theta}_{ij},$$

where $\{\theta_{ij} \geqslant 0\}$ represent the penalties for violating the must-link constraints, and $\{\tilde{\theta}_{ij} \geqslant 0\}$ denote the penalties for violating the cannot-link constraints. Therefore, the goal of *semi-supervised clustering* is to find an optimal partition of $\mathcal{X}$ which can minimize $J(\pi)$. Li, Ding and Jordan [19] first formulated the semi-supervised clustering problem using nonnegative matrix factorization framework and developed updating algorithms based on their earlier work of semi-NMF [12]. Motivated from their work [19], we will develop a novel penalized matrix factorization based approach to co-cluster the data sets of different types with constraints.

2.3 Penalized Matrix Factorization for Constrained Clustering Following [15], we change the penalties of violations in the constraints in $\mathcal{M}$ into the *awards* as

$$\begin{aligned} J(\pi) &= \sum_{c} \sum_{\mathbf{x}_i \in \pi_c} \|\mathbf{x}_i - \mathbf{f}_c\|^2 - \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M} \\ s.t.\ l_i = l_j}} \theta_{ij} + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C} \\ s.t.\ l_i = l_j}} \tilde{\theta}_{ij} \\ &= \sum_{c} \sum_{\mathbf{x}_i} g_{ic} \|\mathbf{x}_i - \mathbf{f}_c\|^2 + \sum_{c} \sum_{i,j} g_{ic} g_{jc} \Theta_{ij} \end{aligned}$$

where

$$\Theta_{ij} = \begin{cases} \tilde{\theta}_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C} \\ -\theta_{ij}, & (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \\ 0, & otherwise \end{cases} \tag{2.3}$$

Defining matrix $\mathbf{\Theta} \in \mathbb{R}^{n \times n}$ with its (i,j)-th entry $\mathbf{\Theta}_{ij} = \Theta_{ij}$, and applying the same trick as in Eq.(2.2), we can rewrite $J(\pi)$ as

$$J(\pi) = \left\| \mathbf{X} - \mathbf{F}\mathbf{G}^T \right\|_F^2 + tr(\mathbf{G}^T \mathbf{\Theta} \mathbf{G}) \tag{2.4}$$

Note that for a general semi-supervised clustering algorithm, we are given $\mathbf{X}$ and $\mathbf{\Theta}$, and we want to solve $\mathbf{F}$ and $\mathbf{G}$. By definition, the elements in $\mathbf{G}$ can only take binary values, which makes the minimization of π unsolvable, therefore we propose to relax the constraint on $\mathbf{G}$ and solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G}} \quad & J(\pi) = \left\| \mathbf{X} - \mathbf{F}\mathbf{G}^T \right\|_F^2 + tr(\mathbf{G}^T \mathbf{\Theta} \mathbf{G}) \\ s.t. \quad & \mathbf{G} \geqslant 0,\ \mathbf{G}^T\mathbf{G} = \mathbf{I} \end{aligned} \tag{2.5}$$

In our later derivations, we find that it is hard to solve the above optimization problem when both constraints being satisfied. Therefore, following the discussion in [12][19], we drop the orthogonal condition $\mathbf{G}^T\mathbf{G} = \mathbf{I}$ and solve the following relaxed optimization problem

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G}} \quad & J(\pi) = \left\| \mathbf{X} - \mathbf{F}\mathbf{G}^T \right\|_F^2 + tr(\mathbf{G}^T \mathbf{\Theta} \mathbf{G}) \\ s.t. \quad & \mathbf{G} \geqslant 0. \end{aligned} \tag{2.6}$$

Compared to the traditional (*nonnegative*) *matrix factorization* problem [12][10][18][11], we can find that the only difference in $J(\pi)$ in the inclusion of the penalty term $tr(\mathbf{G}^T\mathbf{\Theta}\mathbf{G})$, hence we call Eq.(2.6) a *Penalized Matrix Factorization* (*PMF*) problem, in the following we introduce a simple iterative algorithm to solve it.

2.3.1 The Algorithm The basic algorithm procedure for solving Eq.(2.6) is shown in table 2.

2.3.2 Correctness of the Algorithm The objective function $J(\pi)$ in Eq.(2.6) can be expanded as

$$J(\pi) = tr(\mathbf{X}^T\mathbf{X} - 2\mathbf{F}^T\mathbf{X}\mathbf{G} + \mathbf{G}\mathbf{F}^T\mathbf{F}\mathbf{G}^T + \mathbf{G}^T\mathbf{\Theta}\mathbf{G}). \tag{2.7}$$

Thus the correctness of the algorithm in table 2 is guaranteed by the following theorem.

THEOREM 2.1. *If the update rule of* $\mathbf{F}$ *and* $\mathbf{G}$ *in table 2 converges, then the final solution satisfies the KKT optimality condition.*

Table 2: Penalized Matrix Factorization for Constrained Clustering

Inputs: Data matrix $\mathbf{X}$, Constraints matrix $\mathbf{\Theta}$.
Outputs: $\mathbf{F}$, $\mathbf{G}$.
1. Initialize $\mathbf{G}$ using Kmeans as introduced in [11];
2. Repeat the following steps until convergence:
   (a). Fixing $\mathbf{G}$, updating $\mathbf{F}$ by $\mathbf{F} = \mathbf{X}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$;
   (b). Fixing $\mathbf{F}$, updating $\mathbf{G}$ by

$$\mathbf{G}_{ij} \longleftarrow \mathbf{G}_{ij}\sqrt{\frac{(\mathbf{X}^T\mathbf{F})^+_{ij}+[\mathbf{G}(\mathbf{F}^T\mathbf{F})^-]_{ij}+\left(\mathbf{\Theta}^-\mathbf{G}\right)_{ij}}{(\mathbf{X}^T\mathbf{F})^-_{ij}+[\mathbf{G}(\mathbf{F}^T\mathbf{F})^+]_{ij}+\left(\mathbf{\Theta}^+\mathbf{G}\right)_{ij}}}.$$

Proof. Following the standard theory of constrained optimization, we introduce the Lagrangian multipliers $\boldsymbol{\beta}$ and construct the following Lagrangian function

$$L = J(\pi) - tr(\boldsymbol{\beta}\mathbf{G}^T) \tag{2.8}$$

Then combining Eq.(2.8) and Eq.(2.7), we derive that

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{F}} &= -2\mathbf{X}\mathbf{G} + 2\mathbf{F}\mathbf{G}^T\mathbf{G}\\ \frac{\partial L}{\partial \mathbf{G}} &= -2\mathbf{X}^T\mathbf{F} + 2\mathbf{G}\mathbf{F}^T\mathbf{F} + 2\mathbf{\Theta}\mathbf{G} - \boldsymbol{\beta}\end{aligned}$$

Fixing $\mathbf{G}$, letting $\frac{\partial L}{\partial \mathbf{F}} = 0$, we obtain

$$\mathbf{F} = \mathbf{X}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1} \tag{2.9}$$

Fixing $\mathbf{F}$, letting $\frac{\partial L}{\partial \mathbf{G}} = 0$, we get

$$\boldsymbol{\beta} = -2\mathbf{X}^T\mathbf{F} + 2\mathbf{G}\mathbf{F}^T\mathbf{F} + 2\mathbf{\Theta}\mathbf{G}$$

the *KKT complementary condition* for the nonnegativity of $\mathbf{G}$ is

$$(-2\mathbf{X}^T\mathbf{F}+2\mathbf{G}\mathbf{F}^T\mathbf{F}+2\mathbf{\Theta}\mathbf{G})_{ij}\mathbf{G}_{ij} = \boldsymbol{\beta}_{ij}\mathbf{G}_{ij} = 0 \tag{2.10}$$

This is the fixed point equation that the solution must satisfy at convergence. Therefore, let

$$\begin{aligned}\mathbf{\Theta} &= \mathbf{\Theta}^+ - \mathbf{\Theta}^-\\ \mathbf{F}^T\mathbf{F} &= (\mathbf{F}^T\mathbf{F})^+ - (\mathbf{F}^T\mathbf{F})^-\\ \mathbf{X}^T\mathbf{F} &= (\mathbf{X}^T\mathbf{F})^+ - (\mathbf{X}^T\mathbf{F})^-\end{aligned}$$

where $\mathbf{\Theta}^+$, $\mathbf{\Theta}^-$, $(\mathbf{F}^T\mathbf{F})^+$, $(\mathbf{F}^T\mathbf{F})^-$, $(\mathbf{X}^T\mathbf{F})^+$, $(\mathbf{X}^T\mathbf{F})^-$ are all nonnegative. Then given an initial guess of $\mathbf{G}$, the successive update of $\mathbf{G}$ using

$$\mathbf{G}_{ij} \longleftarrow \mathbf{G}_{ij}\sqrt{\frac{(\mathbf{X}^T\mathbf{F})^+_{ij} + [\mathbf{G}(\mathbf{F}^T\mathbf{F})^-]_{ij} + \left(\mathbf{\Theta}^-\mathbf{G}\right)_{ij}}{(\mathbf{X}^T\mathbf{F})^-_{ij} + [\mathbf{G}(\mathbf{F}^T\mathbf{F})^+]_{ij} + \left(\mathbf{\Theta}^+\mathbf{G}\right)_{ij}}} \tag{2.11}$$

will converge to a local minima of the problem. Since at convergence, $\mathbf{G}^{(\infty)} = \mathbf{G}^{(t+1)} = \mathbf{G}^{(t)} = \mathbf{G}$, *i.e.*

$$\mathbf{G}_{ij} = \mathbf{G}_{ij}\sqrt{\frac{(\mathbf{X}^T\mathbf{F})^+_{ij} + [\mathbf{G}(\mathbf{F}^T\mathbf{F})^-]_{ij} + \left(\mathbf{\Theta}^-\mathbf{G}\right)_{ij}}{(\mathbf{X}^T\mathbf{F})^-_{ij} + [\mathbf{G}(\mathbf{F}^T\mathbf{F})^+]_{ij} + \left(\mathbf{\Theta}^+\mathbf{G}\right)_{ij}}} \tag{2.12}$$

which is equivalent to

$$(-2\mathbf{X}^T\mathbf{F} + 2\mathbf{G}\mathbf{F}^T\mathbf{F} + 2\mathbf{\Theta}\mathbf{G})_{ij}\mathbf{G}^2_{ij} = 0,$$

which is equivalent to Eq.(2.10). □

2.3.3 Convergence of the Algorithm Now the only remaining problem is to prove the algorithm in table 2 will finally converge. The same as in [17], we use the *auxiliary function* approach to achieve this goal. The *auxiliary function* is defined as follows.

DEFINITION 2.1. (**Auxiliary Function**) [17] *A function $Z(\mathbf{G}, \mathbf{G}')$ is called an* auxiliary function *of function $L(\mathbf{G})$ if*

$$Z(\mathbf{G}, \mathbf{G}') \geqslant L(\mathbf{G}),\ Z(\mathbf{G}, \mathbf{G}) = L(\mathbf{G})$$

hold for any $\mathbf{G}, \mathbf{G}'$.

Let $\{\mathbf{G}^{(t)}\}$ be the series of matrices obtained from the iterations of the algorithm in table 2, where the superscript (t) represents the iteration number. Now let's define

$$\mathbf{G}^{(t+1)} = arg\min_{\mathbf{G}} Z(\mathbf{G}, \mathbf{G}^{(t)}),$$

where $Z(\mathbf{G}, \mathbf{G}')$ is the auxiliary function for $L(\mathbf{G}) = J(\pi)$ in Eq.(2.7), then by its construction, we have

$$L(\mathbf{G}^{(t)}) = Z(\mathbf{G}^{(t)}, \mathbf{G}^{(t)}) \geqslant Z(\mathbf{G}^{(t+1)}, \mathbf{G}^{(t)}) \geqslant L(\mathbf{G}^{(t+1)}).$$

Thus $L(\mathbf{G}^{(t)})$ is monotonically decreasing. The thing remaining is to find an appropriate $Z(\mathbf{G}, \mathbf{G}')$ and its global minima. We have the following theorem.

THEOREM 2.2. *Updating $\mathbf{F}$ and $\mathbf{G}$ using the rules Eq.(2.9) and Eq.(2.11) will finally converge.*

Proof. Using the preliminary theorem in appendix II, and let $\mathbf{B} = \mathbf{X}^T\mathbf{F}$, $\mathbf{A} = \mathbf{F}^T\mathbf{F}$, we can get the auxiliary function $Z(\mathbf{G}, \tilde{\mathbf{G}})$. Thus let $J(\mathbf{F}, \mathbf{G}) = J(\pi)$ in Eq.(2.7), we get

$$J(\mathbf{F}^0, \mathbf{G}^0) \geqslant J(\mathbf{F}^1, \mathbf{G}^0) \geqslant J(\mathbf{F}^1, \mathbf{G}^1) \geqslant \cdots$$

So $J(\mathbf{F}, \mathbf{G})$ is monotonically decreasing. Since $J(\mathbf{F}, \mathbf{G})$ is obviously bounded below, the theorem is proved. □

3 Penalized Matrix Tri-Factorization for Dyadic Constrained Co-Clustering

In previous section, we have introduced a novel *PMF* algorithm to solve the semi-supervised (constrained) clustering algorithm. One limitation of such algorithm is that it can only tackle the problem when there is only one single type of data objects, *i.e.* it can only process the *homogeneous* data. However, as we discussed in the introduction, many real world data sets are *heterogeneous*. In this section, we will extend the *PMF* algorithm in table 2 and propose a *tri-PMF* algorithm which can cope with the *dyadic constrained co-clustering* problem.

3.1 Problem Formulation In the typical setting of the dyadic co-clustering problem, there are two types of data objects, $\mathcal{X}_1$ and $\mathcal{X}_2$ with size n_1 and dn_2, and we are given a relationship matrix $\mathbf{R}_{12} \in \mathbb{R}^{n_1 \times n_2}$, such that $\mathbf{R}_{12}(i,j)$ represents the relationship between i-th point in $\mathcal{X}_1$ and the j-th point in $\mathcal{X}_2$. Usually $\mathbf{R}_{21} = \mathbf{R}_{12}^T$. The goal of co-clustering is to cluster $\mathcal{X}_1$ and $\mathcal{X}_2$ simultaneously by making use of $\mathbf{R}_{12}$.

Many algorithms have been proposed to solve the dyadic co-clustering algorithm [2][6][8][9]. The authors in [11] proposed a novel algorithm called *nonnegative matrix tri-factorization* (*tri-NMF*) and showed that the solution of *tri-NMF* corresponds to the relaxed solution of clustering the row and column of a relation matrix. More concretely, following the notations we introduced above, the nonnegative matrix tri-factorization aims to solve the following optimization problem

$$\min_{\mathbf{G}_1 \geqslant 0, \mathbf{G}_2 \geqslant 0, \mathbf{S} \geqslant 0} \|\mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S} \mathbf{G}_2^T\|^2, \tag{3.13}$$

where $\mathbf{G}_1$ and $\mathbf{G}_2$ correspond to the cluster indicator matrix of $\mathcal{X}_1$ and $\mathcal{X}_2$ respectively. Note that the original *NMF* problem requires $\mathbf{R}_{12}$ to be nonnegative. In the co-clustering scenario, we can relax this constraint and solve the following optimization problem

$$\min_{\mathbf{G}_1 \geqslant 0, \mathbf{G}_2 \geqslant 0} \|\mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S} \mathbf{G}_2^T\|^2, \tag{3.14}$$

which can be called *semi-tri-NMF* problem following [12].

As discussed in the introduction, we may also have some other information on $\mathcal{X}_1$ and $\mathcal{X}_2$ when we acquire them. In this paper, the information is considered in the form of pairwise constraints on the same type of data objects, *i.e.*, must-link and cannot-link constraints on $\mathcal{X}_1$ and $\mathcal{X}_2$ respectively. Therefore the goal of constrained dyadic co-clustering is to solve the following optimization problem

$$\min_{\mathbf{G}_1 \geqslant 0, \mathbf{G}_2 \geqslant 0} \|\mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S} \mathbf{G}_2^T\|^2 + P(\mathcal{X}_1) + P(\mathcal{X}_2), \tag{3.15}$$

Table 3: Penalized Matrix Tri-Factorization for Dyadic Constrained Co-Clustering

Inputs: Relation matrix $\mathbf{R}_{12}$,
Constraints matrices $\mathbf{\Theta}_1, \mathbf{\Theta}_2$.
Outputs: $\mathbf{G}_1$, $\mathbf{S}$, $\mathbf{G}_2$.
1. Initialize $\mathbf{G}_1$ using Kmeans as introduced in [11];
2. Initialize $\mathbf{G}_2$ using Kmeans as introduced in [11];
3. Repeat the following steps until convergence:
   (a). Fixing $\mathbf{G}_1, \mathbf{G}_2$, updating $\mathbf{S}$ using
   $\mathbf{S} \longleftarrow (\mathbf{G}_1^T \mathbf{G}_1)^{-1} \mathbf{G}_1^T \mathbf{R}_{12} \mathbf{G}_2 (\mathbf{G}_2^T \mathbf{G}_2)^{-1}$;
   (b). Fixing $\mathbf{S}, \mathbf{G}_2$, updating $\mathbf{G}_1$ using
   $$\mathbf{G}_{1ij} \leftarrow \mathbf{G}_{1ij} \sqrt{\frac{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T)_{ij}^+ + [\mathbf{G}_1(\mathbf{S}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S})^-]_{ij} + (\mathbf{\Theta}_1^-\mathbf{G}_1)_{ij}}{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T)_{ij}^- + [\mathbf{G}_1(\mathbf{S}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S})^+]_{ij} + (\mathbf{\Theta}_1^+\mathbf{G}_1)_{ij}}};$$
   (c). Fixing $\mathbf{G}_1, \mathbf{S}$, updating $\mathbf{G}_2$ using
   $$\mathbf{G}_{2ij} \leftarrow \mathbf{G}_{2ij} \sqrt{\frac{(\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S})_{ij}^+ + [\mathbf{G}_2(\mathbf{S}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}^T)^-]_{ij} + (\mathbf{\Theta}_2^-\mathbf{G}_2)_{ij}}{(\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S})_{ij}^- + [\mathbf{G}_2(\mathbf{S}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}^T)^+]_{ij} + (\mathbf{\Theta}_2^+\mathbf{G}_2)_{ij}}}.$$

where $P(\mathcal{X}_1)$ and $P(\mathcal{X}_2)$ denote the penalties of the constraint violations on $\mathcal{X}_1$ and $\mathcal{X}_2$, respectively.

3.2 Constrained Dyadic Co-Clustering via Penalized Matrix Tri-Factorization Similar to Eq.(2.4), we assume here that $P(\mathcal{X}_1)$ and $P(\mathcal{X}_2)$ are of the following quadratic forms.

$$P(\mathcal{X}_1) = tr(\mathbf{G}_1^T \mathbf{\Theta}_1 \mathbf{G}_1) \tag{3.16}$$

$$P(\mathcal{X}_2) = tr(\mathbf{G}_2^T \mathbf{\Theta}_2 \mathbf{G}_2) \tag{3.17}$$

where $\mathbf{\Theta}_1 \in \mathbb{R}^{n_1 \times n_1}$ and $\mathbf{\Theta}_2 \in \mathbb{R}^{n_2 \times n_2}$ are the penalty matrices on $\mathcal{X}_1$ and $\mathcal{X}_2$, such that $\mathbf{\Theta}_1(i,j)$ ($\mathbf{\Theta}_2(i,j)$) represents the penalty of violating the constraints between the i-th and j-th points in $\mathcal{X}_1$ ($\mathcal{X}_2$) as in Eq.(2.3). Then the goal of constrained dyadic co-clustering is just to solve the following optimization problem

$$\min_{\mathbf{G}_1 \geqslant 0, \mathbf{G}_2 \geqslant 0} J = \|\mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S} \mathbf{G}_2^T\|^2 + tr(\mathbf{G}_1^T \mathbf{\Theta}_1 \mathbf{G}_1) + tr(\mathbf{G}_2^T \mathbf{\Theta}_2 \mathbf{G}_2) \tag{3.18}$$

which is a problem to factorize $\mathbf{R}_{12}$ into three matrices $\mathbf{G}_1, \mathbf{S}, \mathbf{G}_2$ with some constraints and penalties, thus we call the problem *Penalized Matrix Tri-Factorization* (*tri-PMF*).

3.2.1 The Algorithm Table 3 provides a simple iterative algorithm to solve the optimization problem Eq.(3.15).

3.2.2 Correctness of the Algorithm Returning to the problem Eq.(3.15), we can first expand J by

$$\begin{aligned} J = & tr(\mathbf{R}_{12}^T \mathbf{R}_{12} - 2\mathbf{G}_2^T \mathbf{R}_{12}^T \mathbf{G}_1 \mathbf{S} + \mathbf{G}_1^T \mathbf{G}_1 \mathbf{S} \mathbf{G}_2^T \mathbf{G}_2 \mathbf{S}^T) \\ & + tr(\mathbf{G}_1^T \mathbf{\Theta}_1 \mathbf{G}_1) + tr(\mathbf{G}_2^T \mathbf{\Theta}_2 \mathbf{G}_2) \end{aligned} \tag{3.19}$$

Then the correctness of the algorithm in table 3 is guaranteed by the following theorem.

THEOREM 3.1. *If the update rule of* $\mathbf{G}_1, \mathbf{S}$ *and* $\mathbf{G}_2$ *in table 3 converges, then the final solution satisfies the KKT optimality condition.*

Proof. The same as in section 2.3, we introduce the Lagrangian multipliers $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, and construct the following Lagrange function

$$L = J - tr(\boldsymbol{\beta}_1 \mathbf{G}_1^T) - tr(\boldsymbol{\beta}_2 \mathbf{G}_2^T) \tag{3.20}$$

Then

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{S}} &= -2\mathbf{G}_1^T\mathbf{R}_{12}\mathbf{G}_2 + 2\mathbf{G}_1\mathbf{G}_1^T\mathbf{S}\mathbf{G}_2^T\mathbf{G}_2 \\
\frac{\partial L}{\partial \mathbf{G}_1} &= -2\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T + 2\mathbf{G}_1\mathbf{S}\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}^T + 2\boldsymbol{\Theta}_1\mathbf{G}_1 - \boldsymbol{\beta}_1 \\
\frac{\partial L}{\partial \mathbf{G}_2} &= -2\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S} + 2\mathbf{G}_2\mathbf{S}^T\mathbf{G}_1^T\mathbf{G}_1\mathbf{S} + 2\boldsymbol{\Theta}_2\mathbf{G}_2 - \boldsymbol{\beta}_2
\end{aligned}$$

Fixing $\mathbf{G}_1,\ \mathbf{G}_2$, we can update $\mathbf{S}$ as

$$\mathbf{S} \longleftarrow (\mathbf{G}_1^T\mathbf{G}_1)^{-1}\mathbf{G}_1^T\mathbf{R}_{12}\mathbf{G}_2(\mathbf{G}_2^T\mathbf{G}_2)^{-1} \tag{3.21}$$

Fixing $\mathbf{S},\ \mathbf{G}_2$, we can get that the KKT complementary condition for the nonnegativity of $\mathbf{G}_1$ is

$$(-2\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T + 2\mathbf{G}_1\mathbf{S}\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}^T + 2\boldsymbol{\Theta}_1\mathbf{G}_1)_{ij}\mathbf{G}_{1ij} = 0 \tag{3.22}$$

Then given an initial guess of $\mathbf{G}_1$, we can successively update $\mathbf{G}_{1ij}$ by

$$\mathbf{G}_{1ij} \leftarrow \mathbf{G}_{1ij}\sqrt{\frac{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T)_{ij}^+ + [\mathbf{G}_1(\mathbf{S}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S})^-]_{ij} + (\boldsymbol{\Theta}_1^-\mathbf{G}_1)_{ij}}{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T)_{ij}^- + [\mathbf{G}_1(\mathbf{S}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S})^+]_{ij} + (\boldsymbol{\Theta}_1^+\mathbf{G}_1)_{ij}}}$$

It can be easily seen that using such a rule, at convergence, $\mathbf{G}_{1ij}$ satisfies

$$(-2\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T + 2\mathbf{G}_1\mathbf{S}\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}^T + 2\boldsymbol{\Theta}_1\mathbf{G}_1)_{ij}\mathbf{G}_{1ij}^2 = 0 \tag{3.23}$$

which is equivalent to Eq.(3.22).

Fixing $\mathbf{S},\ \mathbf{G}_1$, we can get that the KKT complementary condition for the nonnegativity of $\mathbf{G}_2$ is

$$(-2\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S} + 2\mathbf{G}_2\mathbf{S}^T\mathbf{G}_1^T\mathbf{G}_1\mathbf{S} + 2\boldsymbol{\Theta}_2\mathbf{G}_2)_{ij}\mathbf{G}_{2ij} = 0 \tag{3.24}$$

Then given an initial guess of $\mathbf{G}_2$, we can successively update $\mathbf{G}_{2ij}$ by

$$\mathbf{G}_{2ij} \leftarrow \mathbf{G}_{2ij}\sqrt{\frac{(\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S})_{ij}^+ + [\mathbf{G}_2(\mathbf{S}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}^T)^-]_{ij} + (\boldsymbol{\Theta}_2^-\mathbf{G}_2)_{ij}}{(\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S})_{ij}^- + [\mathbf{G}_2(\mathbf{S}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}^T)^+]_{ij} + (\boldsymbol{\Theta}_2^+\mathbf{G}_2)_{ij}}}$$

We can also easily derive that at convergence, $\mathbf{G}_{2ij}$ satisfies that

$$(-2\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S} + 2\mathbf{G}_2\mathbf{S}^T\mathbf{G}_1^T\mathbf{G}_1\mathbf{S} + 2\boldsymbol{\Theta}_2\mathbf{G}_2)_{ij}\mathbf{G}_{2ij}^2 = 0 \tag{3.25}$$

which is equivalent to Eq.(3.24). □

3.2.3 Convergence of the Algorithm Now the only remaining thing is to prove the convergence of the algorithm in table 3. Similar to theorem 2.2, we have the following theorem.

THEOREM 3.2. *Updating* $\mathbf{G}_1,\ \mathbf{G}_2,\ \mathbf{S}$ *using the rules in table 3 will finally converge.*

Proof. Viewing J in Eq.(3.19) as a function of $\mathbf{G}_1$, we can construct the auxiliary function based on the theorem in appendix II by setting $\mathbf{B} = \mathbf{R}_{12}\mathbf{G}_2\mathbf{S}^T,\ \mathbf{A} = \mathbf{S}\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}^T,\ \mathbf{G} = \mathbf{G}_1,\ \boldsymbol{\Theta} = \boldsymbol{\Theta}_1$. Similarly, Viewing J in Eq.(3.19) as a function of $\mathbf{G}_2$, we can construct the auxiliary function based on the theorem in appendix II by setting $\mathbf{B} = \mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S},\ \mathbf{A} = \mathbf{S}^T\mathbf{G}_1^T\mathbf{G}_1\mathbf{S},\ \mathbf{G} = \mathbf{G}_2,\ \boldsymbol{\Theta} = \boldsymbol{\Theta}_2$. Therefore J will be monotonically decreasing using the updating rules of $\mathbf{G}_1$ and $\mathbf{G}_2$. Therefore, let $J(\mathbf{G}_1, \mathbf{S}, \mathbf{G}_2) = J$, then we have

$$J(\mathbf{G}_1^0, \mathbf{S}^0, \mathbf{G}_2^0) \geqslant J(\mathbf{G}_1^0, \mathbf{S}^1, \mathbf{G}_2^0) \geqslant J(\mathbf{G}_1^1, \mathbf{S}^1, \mathbf{G}_2^0) \geqslant \cdots$$

So $J(\mathbf{G}_1, \mathbf{S}, \mathbf{G}_2)$ is monotonically decreasing. Since $J(\mathbf{G}_1, \mathbf{S}, \mathbf{G}_2)$ is obviously bounded below, we prove the theorem. □

4 Symmetric Penalized Matrix Tri-Factorization (tri-SPMF) for Multi-Type Constrained Co-Clustering

In section 2 and section 3 we have introduced how to solve the constrained clustering problem on uni-type or dyadic data sets using the penalized matrix factorization based algorithm. A natural question is how to generalize those algorithms to the data sets containing data objects more than two types, as we have stated in the introduction that many real world data sets have multiple types of data objects. In this section we introduce a novel algorithm called *symmetric penalized matrix tri-factorization* (*tri-SPMF*) to solve such problem.

4.1 Problem Formulation We denote a K-type data set as $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_K\}$, where $\mathcal{X}_i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_{n_i}^i\}$ represent the data set of type i. Assuming we are also given a set of relation matrices $\{\mathbf{R}_{ij} \in \mathbb{R}^{n_i \times n_j}\}$ between different types of data objects with $\mathbf{R}_{ji} = \mathbf{R}_{ij}^T$. Then the goal of co-clustering on $\mathcal{X}$ is just to cluster the data objects in $\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_K$ simultaneously [13][20][21].

In constrained multi-type co-clustering, we also have some prior knowledge, *i.e.*, must-link and cannot-link constraints for each $\mathcal{X}_i$ $(1 \leqslant i \leqslant K)$. Therefore, we can construct a penalty matrix $\boldsymbol{\Theta}_i$ for each $\mathcal{X}_i$. The goal is to cluster $\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_K$ simultaneously by making use of $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \cdots, \boldsymbol{\Theta}_K$. We denote the cluster indicator for $\mathcal{X}_i$ as $\mathbf{G}_i \in \mathbb{R}_+^{n_i \times C_i}$ (C_i is the number of clusters in

$\mathcal{X}_i$). Then a natural way to generalize the penalized matrix tri-factorization for dyadic data to multi-type data is to solve the following optimization problem

(4.26)
$$\min_{\substack{\mathbf{F}_1\geqslant 0\\ \mathbf{F}_2\geqslant 0\\ \vdots \\ \mathbf{F}_K\geqslant 0}} \sum_{0<i<j\leqslant K} \|\mathbf{R}_{ij}-\mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T\|^2+\sum_i tr(\mathbf{G}_i^T\mathbf{\Theta}_i\mathbf{G}_i)$$

However, it is not direct to generalize the PMF or tri-PMF algorithm to solve the above problem. Here we first introduce two lemmas.

LEMMA 4.1. *The optimization problem*

$$\min_{\mathbf{G}_1\geqslant 0,\mathbf{G}_2\geqslant 0} \|\mathbf{R}_{12}-\mathbf{G}_1\mathbf{S}_{12}\mathbf{G}_2^T\|^2 \tag{4.27}$$

can be equivalently solved by the following symmetric matrix tri-factorization problem.

$$\min_{\mathbf{G}\geqslant 0} \|\mathbf{R}-\mathbf{G}\mathbf{S}\mathbf{G}^T\|^2, \tag{4.28}$$

where

$$\mathbf{R} = \begin{bmatrix} \mathbf{0}^{n_1\times n_1} & \mathbf{R}_{12}^{n_1\times n_2} \\ \mathbf{R}_{21}^{n_2\times n_1} & \mathbf{0}^{n_2\times n_2} \end{bmatrix} \tag{4.29}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1^{n_1\times C_1} & \mathbf{0}^{n_1\times C_2} \\ \mathbf{0}^{n_2\times C_1} & \mathbf{G}_2^{n_2\times C_2} \end{bmatrix} \tag{4.30}$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{0}^{C_1\times C_1} & \mathbf{S}_{12}^{C_1\times C_2} \\ \mathbf{S}_{21}^{C_2\times C_1} & \mathbf{0}^{C_2\times C_2} \end{bmatrix} \tag{4.31}$$

where we use superscripts to denote the sizes of the matrices, and $\mathbf{R}_{21}=\mathbf{R}_{12}^T,\ \mathbf{S}_{21}=\mathbf{S}_{12}^T$.

Proof. Following the definitions of $\mathbf{G}$ and $\mathbf{S}$, we can derive that

$$\begin{aligned}\mathbf{G}\mathbf{S}\mathbf{G}^T &= \begin{bmatrix} \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2 \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{S}_{12} \\ \mathbf{S}_{12}^T & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{G}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{G}_1\mathbf{S}_{12} \\ \mathbf{G}_2\mathbf{S}_{12}^T & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{G}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{G}_1\mathbf{S}_{12}\mathbf{G}_2^T \\ \mathbf{G}_2\mathbf{S}_{12}^T\mathbf{G}_1^T & \mathbf{0} \end{bmatrix}\end{aligned}$$

Therefore

$$\begin{aligned}&\|\mathbf{R}-\mathbf{G}\mathbf{S}\mathbf{G}^T\|^2 \\ &= \left\|\begin{bmatrix} \mathbf{0} & \mathbf{R}_{12} \\ \mathbf{R}_{12}^T & \mathbf{0} \end{bmatrix}-\begin{bmatrix} \mathbf{0} & \mathbf{G}_1\mathbf{S}_{12}\mathbf{G}_2^T \\ \mathbf{G}_2\mathbf{S}_{12}^T\mathbf{G}_1^T & \mathbf{0} \end{bmatrix}\right\|^2 \\ &= 2\|\mathbf{R}_{12}-\mathbf{G}_1\mathbf{S}_{12}\mathbf{G}_2^T\|^2\end{aligned}$$

which proves the lemma. □

LEMMA 4.2. *The solutions to Eq.(4.27) using matrix tri-factorization and Eq.(4.28) using matrix tri-factorization are the same.*

Proof. See appendix III.

Combining Lemma 4.1 and Lemma 4.2 we can draw the conclusion that it is equivalent to solve Eq.(4.27) and Eq.(4.28). Since problem Eq.(4.28) is just to factorize a symmetric matrix $\mathbf{R}$ into matrices $\mathbf{S}$ and $\mathbf{G}$, we call it *symmetric matrix tri-factorization* problem. Therefore, returning to problem (4.26), we have the following theorem.

THEOREM 4.1. *It is equivalent to solve Eq.(4.26) and to solve*

$$\min_{\mathbf{G}\geqslant 0} \left\|\mathbf{R}-\mathbf{G}\mathbf{S}\mathbf{G}^T\right\|^2+tr(\mathbf{G}^T\mathbf{\Theta}\mathbf{G}) \tag{4.32}$$

where

$$\mathbf{R} = \begin{bmatrix} \mathbf{0}^{n_1\times n_1} & \mathbf{R}_{12}^{n_1\times n_2} & \cdots & \mathbf{R}_{1K}^{n_1\times n_K} \\ \mathbf{R}_{21}^{n_2\times n_1} & \mathbf{0}^{n_2\times n_2} & \cdots & \mathbf{R}_{2K}^{n_2\times n_K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{K1}^{n_K\times n_1} & \mathbf{R}_{K2}^{n_K\times n_2} & \cdots & \mathbf{0}^{n_K\times n_K} \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1^{n_1\times C_1} & \mathbf{0}^{n_1\times C_2} & \cdots & \mathbf{0}^{n_1\times C_K} \\ \mathbf{0}^{n_2\times C_1} & \mathbf{G}_2^{n_2\times C_2} & \cdots & \mathbf{0}^{n_2\times C_K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}^{n_K\times C_1} & \mathbf{0}_{n_K\times C_2} & \cdots & \mathbf{G}_K^{n_K\times C_K} \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{0}^{C_1\times C_1} & \mathbf{S}_{12}^{C_1\times C_2} & \cdots & \mathbf{S}_{1K}^{C_1\times C_K} \\ \mathbf{S}_{21}^{C_2\times C_1} & \mathbf{0}^{C_2\times C_2} & \cdots & \mathbf{S}_{2K}^{C_2\times C_K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{K1}^{C_K\times C_1} & \mathbf{S}_{K2}^{C_K\times C_2} & \cdots & \mathbf{0}^{C_K\times C_K} \end{bmatrix}$$

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{\Theta}_1^{n_1\times n_1} & \mathbf{0}^{n_1\times n_2} & \cdots & \mathbf{0}^{n_1\times n_K} \\ \mathbf{0}^{n_2\times n_1} & \mathbf{\Theta}^{n_2\times n_2} & \cdots & \mathbf{0}^{n_2\times n_K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}^{n_K\times n_1} & \mathbf{0}^{n_K\times n_2} & \cdots & \mathbf{\Theta}_K^{n_K\times n_K} \end{bmatrix}$$

in which $\mathbf{R}_{ji}=\mathbf{R}_{ij}^T,\ \mathbf{S}_{ji}=\mathbf{S}_{ij}^T$.

Proof. The proof of the theorem is a natural generalization of the proofs of lemma 4.1 and lemma 4.2. □

4.2 The Algorithm Therefore, theorem 4.1 shows that we can solve Eq.(4.26) by equivalently solving the *symmetric penalized matrix tri-factorization* (*tri-SPMF*) problem Eq.(4.32). More concretely, we have the following theorem.

THEOREM 4.2. *Problem (4.32) can be solved via the following updating rule:*

$$\mathbf{S}\leftarrow(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{R}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1} \tag{4.33}$$

(4.34)
$$\mathbf{G}_{ij}\leftarrow\mathbf{G}_{ij}\sqrt{\frac{(\mathbf{R}\mathbf{G}\mathbf{S})_{ij}^+ + [\mathbf{G}(\mathbf{S}\mathbf{G}^T\mathbf{G}\mathbf{S})^-]_{ij} + \left(\mathbf{\Theta}^-\mathbf{G}\right)_{ij}}{(\mathbf{R}\mathbf{G}\mathbf{S})_{ij}^- + [\mathbf{G}(\mathbf{S}\mathbf{G}^T\mathbf{G}\mathbf{S})^+]_{ij} + \left(\mathbf{\Theta}^+\mathbf{G}\right)_{ij}}}$$

Table 4: Symmetric Penalized Matrix Tri-Factorization for Multi-Type Constrained Co-Clustering

Inputs: Relation matrices $\{\mathbf{R}_{ij}\}_{1\leqslant i<j\leqslant K}$;
Penalty matrices $\{\mathbf{\Theta}_i\}_{1\leqslant i\leqslant K}$
Outputs: $\mathbf{G}$, $\mathbf{S}$
1. Randomly initialize $\{\mathbf{G}_i\}_{1\leqslant i\leqslant K}$;
2. Construct $\mathbf{R}$ and $\mathbf{G}$ as their definitions in theorem 4.1;
3. Repeat the following two steps till convergence.
   (a). Updating $\mathbf{S}$ using Eq.(4.33);
   (b). Updating $\mathbf{G}$ using Eq.(4.34).

Proof. The proof can be easily derived based on the analysis of *tri-PMF* in section 3 with $\mathbf{G}_1 = \mathbf{G}_2 = \mathbf{G}$.

The basic procedure of multi-type constrained co-clustering based on the symmetric penalized matrix tri-factorization is summarized in table 4. Note that when updating $\mathbf{S}$, we can make use of the special block-diagonal structure of $\mathbf{G}$, and when updating $\mathbf{G}$, we only need to update its nonzero blocks on the diagonal line.

5 Experiments

In this section we present the experimental results of applying our penalized matrix factorization algorithm for semi-supervised (co-)clustering.

5.1 Experiments on Uni-Type Data Sets In this subsection, we conduct a set of experiments to show the effectiveness of our penalized matrix factorization algorithm (table 2) on clustering uni-type data set with constraints.

5.1.1 Data Sets The data sets used in our experiments include six UCI data sets[1]. Here are some basic information of those data sets. Table 5 summarizes the basic information of those data sets.

- **Balance**. This data set was generated to model psychological experimental results. There are totally 625 examples that can be classified as having the balance scale tip to the right, tip to the left, or be balanced.
- **Iris**. This data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
- **Ionosphere**. It is a collection of the radar signals belonging to two classes. The data set contains 351 objects in total, which are all 34-dimensional.

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html

Table 5: Descriptions of the datasets

Datasets	Sizes	Classes	Dimensions
Balance	625	3	4
Iris	150	3	4
Ionosphere	351	2	34
Soybean	562	19	35
Wine	178	3	13
Sonar	208	2	60

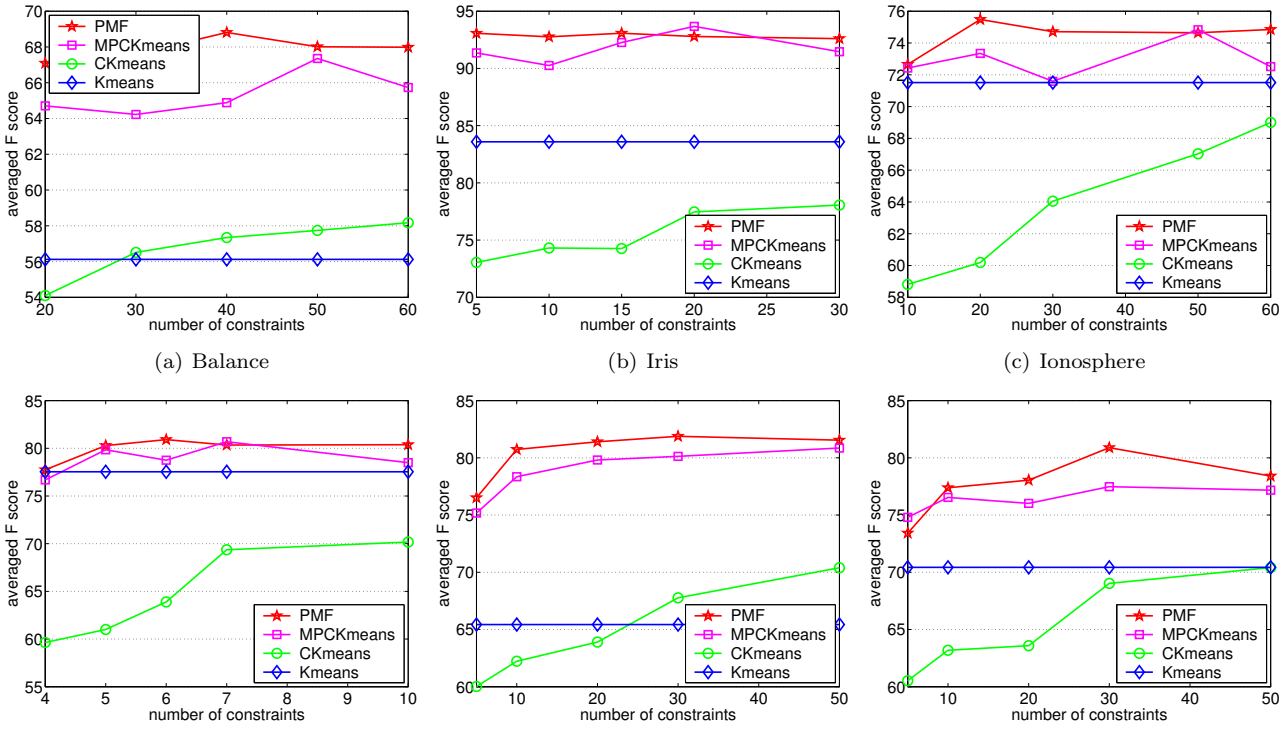
- **Soybean**. It is collected from the Michalski's famous soybean disease databases, which contains 562 instances from 19 classes.
- **Wine**. The purpose of this data set is to use chemical analysis for determining the origin of wines. It contains 178 instances from 3 classes.
- **Sonar**. This is the data set used by Gorman and Sejnowski in their study of the classification of sonar signals using a neural network, which contains 208 instances from 2 classes.

5.1.2 Experimental Settings In the experiments, the penalty of violating a must-link constraint is set to 1 and the penalty of violating a cannot-link constraint is set to 2. The constraints were generated as follows: for each constraint, we picked out one pair of data points randomly from the input data sets (the labels of which were available for evaluation purpose but unavailable for clustering). If the labels of this pair of points were the same, then we generated a must link. If the labels were different, a cannot link was generated. The amounts of constraints were determined by the size of input data. In all the experiments, the results were averaged over 50 trials to eliminate the difference caused by constraints.

In our experiments, we also carry out the original *kmeans* algorithm, the *constrained kmeans* (*CKmeans*) algorithm [22], and the *MPC-Kmeans* (*MPCKmeans*) [5] algorithm for comparison. The F-score [16] is used to evaluate the performance of each algorithm.

5.1.3 Experimental Results Figure 2 shows the F-scores (in percentages) of the four algorithms on the six UCI data sets under different amounts of constraints respectively, from which we can clearly see the superiority of our *PMF* algorithm.

5.2 Experiments on Multi-Type Data Sets In this subsection, we present the results on applying our *tri-SPMF* algorithm to co-cluster a multi-type data set with constraints.



(a) Balance (b) Iris (c) Ionosphere

(d) Soybean (e) Sonar (f) Wine

Figure 2: Experimental results of different algorithms on clustering uni-type data set with constraints.

Table 6: Data Set 1

Field Name	Board name	Notation
Computer Science	C++Builder	C_1
Computer Science	Delphi	C_2
Computer Science	Database	C_3
Sports	Baseball	C_4
Sports	Volleyball	C_5
Sports	Badminton	C_6

Table 7: Data Set 2

Field Name	Board name	Notation
Computer Science	Virus	C_7
Computer Science	Unix	C_8
Entertainment	Music	C_9
Entertainment	Dance	C_{10}
Society	Law	C_{11}
Society	Commerce	C_{12}

5.2.1 Data Set For testing the effectiveness of the *tri-SPMF* algorithm on clustering the multi-type data, we adopt a data set sampled from the *Bulletin Board Systems* (*BBS*) data in [14]. In a *BBS* system, the users first register IDs. Using their IDs, the users can read others' published messages and leave their own messages. The whole system consists of many discussion fields, each of which contains many boards with similar themes. The boards are named to reflect the contents of the articles in them [14]. Once an ID posts a new article (initial article) on one board, the others can show their opinions by replying the initial article using reply articles. The initial article and reply articles constitute a topic. Each board contains many topics. Each topic connects with several IDs through articles.

People's behaviors on the BBS usually reflect their interests. For example, people who post articles in the same topic may share similar interests, and people who are interested in the same boards or discussion fields may have something in common, *e.g*, similar background and education level. In this sense, it is meaningful to cluster the people(IDs) based on the relationships among the IDs and the topics or boards. On the other hand, the topics in the same board or being discussed by the same people may have similar contents. Clustering the topics can help us find the similar topics more quickly. At last, the clustering of the boards is also useful since it can help the administrators to rearrange the boards into appropriate discussion fields. We can find that the above clustering problems can be modeled

Table 8: Data Set 3

Field Name	Board name	Notation
Computer Science	VisualBasic	C_{13}
Computer Science	Programming	C_{14}
Entertainment	Music	C_{15}
Entertainment	Dance	C_{16}
Sports	Speed	C_{17}
Sports	Running	C_{18}
Sense	Heart	C_{19}
Sense	Memory	C_{20}

by a three-type co-clustering problem with the three data types referring to user IDs, topics and boards.

In this paper, we used three subsets from this system. In each data set, several boards were sampled from several discussion fields. In each board, 80 topics are sampled randomly. The names of the fields and boards we used are listed in table 6, 7, 8. The user IDs related to these topics and boards are found out. Then the tensor was constructed by the co-occurrence of these three data types similar to the toy problem.

5.2.2 Experimental Settings In our experiments, there are three data types: topics ($\mathcal{X}_1$), user IDs ($\mathcal{X}_2$) and boards ($\mathcal{X}_3$). The topic-user matrix($\mathbf{R}_{12}$) was constructed with the number of articles each user posted in each topics with *TF-IDF* normalization [1]. The topic-board matrix ($\mathbf{R}_{13}$) was constructed such that if a topic belongs to a board, then the corresponding entry of $\mathbf{R}_{13}$ is 1. $\mathbf{R}_{23}$ was constructed such that if the user had posted any articles on that board, then the corresponding element of $\mathbf{R}_{23}$ is set to 1. Finally the elements of $\mathbf{R}_{23}$ were also normalized using *TF-IDF* scheme. In our method, we randomly generate 500 constraints on $\mathcal{X}_2$ based on their registered profiles, 100 constraints on $\mathcal{X}_1$ based on the boards they belong to, and 10 constraints on $\mathcal{X}_3$ based on their corresponding fields. Besides our algorithm, the results of applying the *Spectral Relational Clustering* (*SRC*) method [21] and *Multiple Latent Semantic Analysis* (*MLSA*) method [24] are also included for comparison. The evaluation metric is also the F score computed based on the clustering results on topics, the ground truth of which is set to be the classes corresponding to the field names they belong to.

5.2.3 Experimental Results The experimental results are shown in table 9, in which the value of d represents the different number of clusters. From the table we can clearly see the effectiveness of our algorithm (note that the F scores of our *tri-SPMF* algorithm are the values averaged over 50 independent runs).

6 Conclusions

In this paper, we proposed a novel semi-supervised clustering algorithm based on matrix factorization. Moreover, we also extend our algorithm to cluster dyadic and multi-type data sets with constraints. The experimental results show the effectiveness of our method.

Appendix I: A Preliminary Proposition

PROPOSITION 6.1. *For any nonnegative matrices* $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{k\times k}$, $\mathbf{S} \in \mathbb{R}^{n\times k}$, $\mathbf{S}' \in \mathbb{R}^{n\times k}$, *and* $\mathbf{A}$, $\mathbf{B}$ *are symmetric, the following inequality holds*

$$\sum_{ip} \frac{(\mathbf{AS}'\mathbf{B})_{ip}\mathbf{S}_{ip}^2}{\mathbf{S}_{ip}} \geqslant tr(\mathbf{S}^T\mathbf{ASB})$$

Proof. See theorem 6 in [11].

Appendix II: A Preliminary Theorem

THEOREM 6.1. *Let*

$$J(\mathbf{G}) = tr(-2\mathbf{G}^T\mathbf{B} + \mathbf{GAG}^T) + tr(\mathbf{G}^T\mathbf{\Theta G}) \tag{6.35}$$

where $\mathbf{A}$, $\mathbf{\Theta}$ *are symmetric,* $\mathbf{G}$ *is nonnegative. Then the following function*

$$\begin{aligned} Z(\mathbf{G},\mathbf{G}') &= -2\sum_{ij}\mathbf{B}_{ij}^+\mathbf{G}'_{ij}\left(1+\log\frac{\mathbf{G}_{ij}}{\mathbf{G}'_{ij}}\right) \\ &+ \sum_{ij}\mathbf{B}_{ij}^-\frac{\mathbf{G}_{ij}^2+\mathbf{G}'^2_{ij}}{2\mathbf{G}'_{ij}} + \sum_{ij}\frac{(\mathbf{G}'\mathbf{A}^+ + \mathbf{\Theta}^+\mathbf{G}')_{ij}\mathbf{G}_{ij}^2}{\mathbf{G}'_{ij}} \\ &- \sum_{ijk}\mathbf{A}_{jk}^-\mathbf{G}'_{ij}\mathbf{G}'_{ik}\left(1+\log\frac{\mathbf{G}_{ij}\mathbf{G}_{ik}}{\mathbf{G}'_{ij}\mathbf{G}'_{ik}}\right) \\ &- \sum_{ijk}\mathbf{\Theta}_{jk}^-\mathbf{G}'_{ji}\mathbf{G}'_{ki}\left(1+\log\frac{\mathbf{G}_{ji}\mathbf{G}_{ki}}{\mathbf{G}'_{ji}\mathbf{G}'_{ki}}\right) \end{aligned}$$

is an auxiliary function for $J(\mathbf{G})$. *Furthermore, it is a convex function in* $\mathbf{G}$ *and its global minimum is*

$$\mathbf{G}_{ij} = \mathbf{G}_{ij}\sqrt{\frac{(\mathbf{B})_{ij}^+ + [\mathbf{GA}^-]_{ij} + \left(\mathbf{\Theta}^-\mathbf{G}\right)_{ij}}{(\mathbf{B})_{ij}^- + [\mathbf{GA}^+]_{ij} + \left(\mathbf{\Theta}^+\mathbf{G}\right)_{ij}}} \tag{6.36}$$

Proof. We rewritten Eq.(2.7) as

$$\begin{aligned} J(\mathbf{G}) = tr(&-2\mathbf{G}^T\mathbf{B}^+ + 2\mathbf{G}^T\mathbf{B}^- + \mathbf{GA}^+\mathbf{G}^T - \mathbf{GA}^-\mathbf{G}^T) \\ &+tr(\mathbf{G}^T\mathbf{\Theta}^+\mathbf{G} - \mathbf{G}^T\mathbf{\Theta}^-\mathbf{G}) \end{aligned}$$

by ignoring $tr(\mathbf{X}^T\mathbf{X})$. By applying the proposition in appendix I, we have

$$\begin{aligned} tr(\mathbf{G}^T\mathbf{\Theta}^+\mathbf{G}) &\leqslant \sum_{ij}\frac{(\mathbf{\Theta}^+\mathbf{G}')_{ij}\mathbf{G}_{ij}^2}{\mathbf{G}'_{ij}} \\ tr(\mathbf{GA}^+\mathbf{G}^T) &\leqslant \sum_{ij}\frac{(\mathbf{G}'\mathbf{A}^+)_{ij}\mathbf{G}_{ij}^2}{\mathbf{G}'_{ij}} \end{aligned}$$

Table 9: The F measure of three algorithms on the three data sets

Data Sets	Algorithm	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$	$d=9$	$d=10$
1	MLSA	0.7019	0.7079	0.7549	0.7541	0.7081	0.7031	0.6990	0.7066
1	SRC	0.7281	0.6878	0.6183	0.6183	0.6103	0.6029	0.4783	0.4268
1	Tri-SPMF	**0.7948**	**0.8011**	**0.8021**	**0.7993**	**0.7754**	**0.7732**	**0.7653**	**0.7590**
2	MLSA	0.7651	0.7429	0.7581	0.7309	0.7284	0.7063	0.6806	0.6856
2	SRC	0.7627	0.7226	0.7280	0.6965	0.6972	0.6620	0.6570	0.5949
2	Tri-SPMF	**0.8007**	**0.7984**	**0.7938**	**0.7896**	**0.7810**	**0.7763**	**0.7801**	**0.7754**
3	MLSA	0.6689	0.6511	0.6987	0.7301	0.7236	0.7045	0.7197	0.6953
3	SRC	0.7556	0.7666	0.7472	0.7125	0.6758	0.6887	0.6636	0.6456
3	Tri-SPMF	**0.8095**	**0.8034**	**0.7993**	**0.7874**	**0.7703**	**0.7722**	**0.7665**	**0.7492**

Moreover, by the inequality

$$a \leqslant (a^2+b^2)/2b, \ \ for \ \forall \ a,b>0$$

We have

$$tr(\mathbf{G}^T\mathbf{B}^-) = \sum_{ij}\mathbf{B}^-_{ij}\mathbf{G}_{ij} \leqslant \sum_{ij}\mathbf{B}^-_{ij}\frac{\mathbf{G}^2_{ij}+\mathbf{G}'^2_{ij}}{2\mathbf{G}'_{ij}}$$

To obtain the lower bounds for the remaining terns, we use the inequality that $z \geqslant 1+\log z$, which holds for any $z>0$, then

$$\begin{aligned}\frac{\mathbf{G}_{ij}}{\mathbf{G}'_{ij}} &\geqslant 1+\log\frac{\mathbf{G}_{ij}}{\mathbf{G}'_{ij}}\\ \frac{\mathbf{G}_{ij}\mathbf{G}_{ik}}{\mathbf{G}'_{ij}\mathbf{G}'_{ik}} &\geqslant 1+\log\frac{\mathbf{G}_{ij}\mathbf{G}_{ik}}{\mathbf{G}'_{ij}\mathbf{G}'_{ik}}\end{aligned}$$

Then

$$\begin{aligned}tr(\mathbf{G}^T\mathbf{B}^+) &\geqslant \sum_{ij}\mathbf{B}^+_{ij}\mathbf{G}'_{ij}\left(1+\log\frac{\mathbf{G}_{ij}}{\mathbf{G}'_{ij}}\right)\\ tr(\mathbf{G}^T\mathbf{\Theta}^-\mathbf{G}) &\geqslant \sum_{ijk}\mathbf{\Theta}^-_{jk}\mathbf{G}'_{ji}\mathbf{G}'_{ki}\left(1+\log\frac{\mathbf{G}_{ji}\mathbf{G}_{ki}}{\mathbf{G}'_{ji}\mathbf{G}'_{ki}}\right)\\ tr(\mathbf{G}\mathbf{A}^-\mathbf{G}^T) &\geqslant \sum_{ijk}\mathbf{A}^-_{jk}\mathbf{G}'_{ij}\mathbf{G}'_{ik}\left(1+\log\frac{\mathbf{G}_{ij}\mathbf{G}_{ik}}{\mathbf{G}'_{ij}\mathbf{G}'_{ik}}\right)\end{aligned}$$

By summing over all the bounds, we can get $Z(\mathbf{G},\mathbf{G}')$, which clearly satisfies (1) $Z(\mathbf{G},\mathbf{G}') \geqslant J(\mathbf{G})$; (2) $Z(\mathbf{G},\mathbf{G}) = J(\mathbf{G})$.

To find the minimum of $Z(\mathbf{G},\mathbf{G}')$, we take

$$\begin{aligned}\frac{\partial Z(\mathbf{G},\mathbf{G}')}{\partial \mathbf{G}_{ij}} &= -2\mathbf{B}^+_{ij}\frac{\mathbf{G}'_{ij}}{\mathbf{G}_{ij}} + 2\mathbf{B}^-_{ij}\frac{\mathbf{G}_{ij}}{\mathbf{G}'_{ij}} + \frac{2(\mathbf{G}'\mathbf{A}^+)_{ij}\mathbf{G}_{ij}}{\mathbf{G}'_{ij}}\\ &-\frac{2(\mathbf{G}'\mathbf{A}^-)_{ij}\mathbf{G}'_{ij}}{\mathbf{G}_{ij}} + \frac{2(\mathbf{\Theta}^+\mathbf{G}')_{ij}\mathbf{G}_{ij}}{\mathbf{G}'_{ij}} - \frac{2(\mathbf{\Theta}^-\mathbf{G}')_{ij}\mathbf{G}'_{ij}}{\mathbf{G}_{ij}}\end{aligned}$$

and the Hessian matrix for $Z(\mathbf{G},\mathbf{G}')$

$$\frac{\partial Z(\mathbf{G},\mathbf{G}')}{\partial \mathbf{G}_{ij}\partial \mathbf{G}_{kl}} = \delta_{ik}\delta_{jl}\mathbf{\Phi}_{ij} \tag{6.37}$$

is a diagonal matrix with positive diagonal elements

$$\begin{aligned}\mathbf{\Phi}_{ij} &= \frac{2(\mathbf{B}^+ + \mathbf{G}'\mathbf{A}^- + \mathbf{\Theta}\mathbf{G}')_{ij}\mathbf{G}'_{ij}}{\mathbf{G}^2_{ij}}\\ &+\frac{2(\mathbf{B}^- + \mathbf{G}'\mathbf{A}^+ + \mathbf{\Theta}\mathbf{G}')_{ij}}{\mathbf{G}'_{ij}}\end{aligned} \tag{6.38}$$

Thus $Z(\mathbf{G},\mathbf{G}')$ is a convex function of $\mathbf{G}$. Therefore, we can obtain the global minimum of $Z(\mathbf{G},\mathbf{G}')$ by setting $\partial Z(\mathbf{G},\mathbf{G}')/\partial\mathbf{G}_{ij}=0$ and solving for $\mathbf{G}$, from which we can get Eq.(6.36). □

Appendix III: Proof of Lemma 4.2

Proof. It can be easily inferred that (from the derivation in section 3) the updating rules for solving Eq.(4.27) are

$$\begin{aligned}\mathbf{S}_{12} &\leftarrow (\mathbf{G}_1^T\mathbf{G}_1)^{-1}\mathbf{G}_1^T\mathbf{R}_{12}\mathbf{G}_2(\mathbf{G}_2^T\mathbf{G}_2)^{-1}\\ \mathbf{G}_{1ij} &\leftarrow \mathbf{G}_{1ij}\sqrt{\frac{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}_{12}^T)^+_{ij}+[\mathbf{G}_1(\mathbf{S}_{12}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}_{12})^-]_{ij}}{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}_{12}^T)^-_{ij}+[\mathbf{G}_1(\mathbf{S}_{12}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}_{12})^+]_{ij}}}\\ \mathbf{G}_{2ij} &\leftarrow \mathbf{G}_{2ij}\sqrt{\frac{(\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S}_{12})^+_{ij}+[\mathbf{G}_2(\mathbf{S}_{12}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}_{12}^T)^-]_{ij}}{(\mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S}_{12})^-_{ij}+[\mathbf{G}_2(\mathbf{S}_{12}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}_{12}^T)^+]_{ij}}}\end{aligned}$$

Using the similar approach, we can derive the updating rules for solving Eq.(4.28) are

$$\begin{aligned}\mathbf{S} &\leftarrow (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{R}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\\ \mathbf{G}_{ij} &\leftarrow \mathbf{G}_{ij}\sqrt{\frac{(\mathbf{R}\mathbf{G}\mathbf{S})^+_{ij}+[\mathbf{G}(\mathbf{S}\mathbf{G}^T\mathbf{G}\mathbf{S})^-]_{ij}}{(\mathbf{R}\mathbf{G}\mathbf{S})^-_{ij}+[\mathbf{G}(\mathbf{S}\mathbf{G}^T\mathbf{G}\mathbf{S})^+]_{ij}}}\end{aligned}$$

Due to the special block-diagonal form of $\mathbf{R}$, $\mathbf{G}$ and $\mathbf{S}$ (see Eq.(4.29)-Eq.(4.31)), we can get that

$$\mathbf{G}^T\mathbf{G} = \begin{bmatrix}\mathbf{G}_1^T & \mathbf{0}\\ \mathbf{0} & \mathbf{G}_2^T\end{bmatrix}\begin{bmatrix}\mathbf{G}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{G}_2\end{bmatrix} = \begin{bmatrix}\mathbf{G}_1^T\mathbf{G}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{G}_2^T\mathbf{G}_2\end{bmatrix}$$

$$\mathbf{G}^T\mathbf{R}\mathbf{F} = \begin{bmatrix}\mathbf{0} & \mathbf{G}_1^T\mathbf{R}_{12}\mathbf{G}_2\\ \mathbf{G}_2^T\mathbf{R}_{12}^T\mathbf{G}_1 & \mathbf{0}\end{bmatrix}$$

Then

$$(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{R}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1} = \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{S}} \\ \tilde{\mathbf{S}}^T & \mathbf{0} \end{bmatrix}$$

where $\tilde{\mathbf{S}} = (\mathbf{G}_1^T\mathbf{G}_1)^{-1}\mathbf{G}_1^T\mathbf{R}_{12}\mathbf{G}_2(\mathbf{G}_2^T\mathbf{G}_2)^{-1}$, which shows the equivalence between the updating rules of $\mathbf{S}$ and $\mathbf{S}_{12}$. Moreover, it also suggests that when updating $\mathbf{S}$ in solving the symmetric semi-tri-NMF problem Eq.(4.28), we only need to update the corresponding $\mathbf{S}_{12}$ part in $\mathbf{S}$ (see the definition of $\mathbf{S}$ in Eq.(4.31)).

Similarly, we can get that

$$\mathbf{RGS} = \begin{bmatrix} \mathbf{R}_{12}\mathbf{G}_2\mathbf{S}_{12}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{12}^T\mathbf{G}_1\mathbf{S}_{12} \end{bmatrix}$$

$$\mathbf{SF}^T\mathbf{FS} = \begin{bmatrix} \mathbf{S}_{12}\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}_{12}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{12}^T\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}_{12} \end{bmatrix}$$

Bringing these two equations back to the updating rule of $\mathbf{G}_{ij}$, we can find that the rules for updating the corresponding part $\mathbf{G}_1$, $\mathbf{G}_2$ in $\mathbf{G}$ (see the definition of $\mathbf{G}$ in Eq.(4.30), and we do not need to updating the zero part in $\mathbf{G}$) are equivalent to the updating rules of $\mathbf{G}_1$ and $\mathbf{G}_2$ in solving Eq.(4.27). Therefore the solutions to Eq.(4.28) is equivalent to the solutions to Eq.(4.27). □

References

[1] R. Baeza- Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[2] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A Generalized Maximum Entropy Approach to Bregman Co-Clustering and Matrix Approximations. In *Proc. of SIGKDD*, 509-514, 2004.

[3] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised Clustering by Seeding. *Proc. of ICML*, 2002.

[4] S. Basu, M. Bilenko, and R. J. Mooney. A Probabilistic Framework for Semi-Supervised Clustering. In *Proc. of SIGKDD*, 59-68, 2004.

[5] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of ICML*, 2004.

[6] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra. Minimum Sum-Squared Residue Co-clustering of Gene Expression Data. In *Proc. of the SIAM SDM*, 114-125, 2004.

[7] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society. 1997.

[8] I. S. Dhillon. Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning. *Proc. of SIGKDD*, 269-274. 2001.

[9] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-Theoretic Co-clustering. *Proc. of SIGKDD*, 2003.

[10] C. Ding, X. He, and H. D. Simon. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. *Proc. SIAM SDM*, pp:606-610. 2005.

[11] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal Nonnegative Matrix Tri-factorizations for Clustering. In *Proc. of SIGKDD*, 126-135. 2006.

[12] C. Ding, T. Li, and M. I. Jordan. Convex and Semi-Nonnegative Matrix Factorizations. *LBNL Tech Report* 60428. 2006.

[13] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, Consistent Bipartite Graph Co-Partitioning for Star-Structured High-Order Heterogeneous Data Co-Clustering. In *Proc. of SIGKDD*, pp. 41-50, 2005.

[14] Z. Kou, and C. Zhang. Reply Networks on a Bulletin Board System. *Phys Rev E*, 2003, 67(3-2): 036117.

[15] B. Kulis, S. Basu, I. Dhillon, R. J. Mooney. Semi-Supervised Graph Clustering: A Kernel Approach. In *Proc. of ICML*, 457-464, 2005.

[16] B. Larsen, C. Aone. Fast and Effective Text Mining Using Linear-time Document Clustering. *In Proc. SIGKDD*, pp 16-22.

[17] D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. *NIPS 13*, 556-562, 2001.

[18] T. Li and C. Ding. Relationships Among Various Non-negative Matrix Factorization Methods for Clustering. *Proc. of ICDM*. pp.362-371. 2006.

[19] T. Li, C. Ding and M. I. Jordan. Solving Consensus and Semi-supervised Clustering Problems Using Non-negative Matrix Factorization. In *Proc. of ICDM*. 2007.

[20] B. Long, X. Wu, Z. (Mark) Zhang, and P. S. Yu. Unsupervised Learning on K-partite Graphs, In *Proc. of SIGKDD*, 317-326, 2006.

[21] B. Long, Z. (Mark) Zhang, X. Wu, and P. S. Yu, Spectral Clustering for Multi-Type Relational Data, In *Proc. of ICML*, 2006.

[22] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. of ICML*, 2001.

[23] H. Zha, C. Ding, M. Gu, X. He, H. Simon. Spectral Relaxation for K-means Clustering, *NIPS 14*. 2001.

[24] X. Wang, J. Sun, Z. Chen, and C. Zhai. Latent semantic analysis for multiple-type interrelated data objects. *SIGIR*, 2006.