High-Order Co-Clustering via Strictly Orthogonal and Symmetric ℓ_1-Norm Nonnegative Matrix Tri-Factorization

Kai Liu and **Hua Wang**[*]
Department of Computer Science, Colorado School of Mines, Golden, CO 80401, U.S.A.
cskailiu@gmail.com, huawangcs@gmail.com

Abstract

Different to traditional clustering methods that deal with one single type of data, High-Order Co-Clustering (HOCC) aims to cluster multiple types of data simultaneously by utilizing the inter- or/and intra-type relationships across different data types. In existing HOCC methods, data points routinely enter the objective functions with squared residual errors. As a result, outlying data samples can dominate the objective functions, which may lead to incorrect clustering results. Moreover, existing methods usually suffer from soft clustering, where the probabilities to different groups can be very close. In this paper, we propose an ℓ_1-norm symmetric nonnegative matrix tri-factorization method to solve the HOCC problem. Due to the orthogonal constraints and the symmetric ℓ_1-norm formulation in our new objective, conventional auxiliary function approach no longer works. Thus we derive the solution algorithm using the alternating direction method of multipliers. Extensive experiments have been conducted on a real world data set, in which promising empirical results, including less time consumption, strictly orthogonal membership matrix, lower local minima *etc*., have demonstrated the effectiveness of our proposed method.

1 Introduction

Recent advancements of Internet and other computational technologies have brought data with much richer structures, which often convey useful information for building more effective learning models. For example, one can use different types of information to recommend movies to a user, such as the user's watching history, comments on watched movies, habits, job, age and social networks, to name a few. These different types of information are represented as different types of data that usually interrelate with each other by various means. For example, user preferences, comments and watching history can be correlated via weighted co-occurrence matrices — some movies are more favored by a user, if they appear more frequently in the user's watching history; while, if a user gave negative comments to a movie, this user is less likely to watch other movies of the same genre. Such highly heterogeneous data with inter-type relationships are called *multi-type relational data* [Long *et al.*, 2006]. The learning problems to deal with multi-type relational data under the unsupervised setting is called as *High-Order Co-Clustering (HOCC)* [Wang *et al.*, 2011a; Wang *et al.*, 2011b], which has attracted more and more attention in recent years.

Spectral clustering [Long *et al.*, 2006] was first proposed to cluster multi-type relational data, though only inter-type information was utilized. Recently, Wang *et al*. [2011b] devised an indicator Nonnegative Matrix Tri-Factorization (NMTF) method to avoid soft clustering and improve the computational speed. But since this method searches the binary space that is small, it may not always find a good optimum. It was then further developed by performing optimization in a continuous space [Wang *et al.*, 2011a]. However, the clustering performance was not improved, because the orthogonal constraints were not imposed on the factor matrices in the new objective, which, though, was very important to avoid degenerate solutions [Ding *et al.*, 2006]. Besides the limitations mentioned above, all these previous studies routinely formulated their NMTF objectives as least-square error functions, which are notoriously known to be sensitive to outliers [Kong *et al.*, 2011]. However, at the era of big data, noise and outliers are inevitable by nature due to the ever increasing data sizes. To tackle this problem, Liu and Wang [2015] improved these prior studies by using the ℓ_1-norm distances. However, because the new objective in [Liu and Wang, 2015] computes the symmetric NMTF that involves the fourth-order matrix polynomials, the factor matrices in the solution were not orthogonally constrained due to the mathematical difficulty. Moreover, same as many existing NMTF methods that derive the solutions using Multiplicative Updating Algorithm (MUA) [Lee and Seung, 2001], this method also suffers from suboptimal solutions because it is easily trapped into local minimum [Lin, 2007].

To address all above difficulties, in this paper we propose a novel HOCC method via a strictly orthogonal and symmetric ℓ_1-norm NMTF method. We derive the solution algorithm using the Alternating Direction Method of Multipliers (ADMM) [Boyd *et al.*, 2011], such that the orthogonal con-

[*] Corresponding author. This work was partially supported by NSF-IIS 1423591 and NSF-IIS 1652943.

straints are strictly enforced on the factor matrices. Extensive experiments have been conducted with promising results that validate the effectiveness of our new method on clustering multi-type relational data.

2 High-Order Co-Clustering via Graph Regularized Symmetric ℓ_1-Norm NMTF

Throughout this paper, we use $A^{(ij)}$ to denote the entry at the i-th row and j-th column of a matrix A. A K-type relational data set can be denoted as $\chi = \{\chi_1, \chi_2, \ldots, \chi_K\}$, where $\chi_k = \{x_1^k, x_2^k, \ldots, x_{n_k}^k\}$ represents the data of the k-th type. Suppose that we are given a set of relationship matrices $\{R_{kl} \in \Re^{n_k \times n_l}\}_{(1 \le k \le K, 1 \le l \le K)}$ between different types of data objects, then we assume $R_{kl} = R_{lk}^T$. Our goal is to simultaneously partition the data objects in $\chi_1, \chi_2, \ldots, \chi_K$ into $c_1, c_2, \ldots, c_K$ disjoint clusters respectively.

2.1 Co-clustering Two-Type Relational Data Using Symmetric ℓ_1-Norm NMTF

The simplest multi-type relational data involve only two types of data objects, which widely appear in many real world applications, like *words* and *documents* in document analysis, *users* and *items* in collaborative filtering, *etc*. Simultaneously clustering two-type relational data is often called *co-clustering* or *bi-clustering*. Ding *et al.* [2006] proposed to use NMTF to simultaneously cluster the rows and columns of an input nonnegative relationship matrix R_{ij} by decomposing it into three nonnegative factor matrices, which minimizes:

$$\begin{aligned} J = \|R_{12} - G_1 S_{12} G_2^T\|_F^2, \quad s.t.\ G_1 \ge 0, G_2 \ge 0, S_{12} \ge 0, \\ G_1^T G_1 = I, G_2^T G_2 = I, \qquad (1) \end{aligned}$$

where $\|\cdot\|_F^2$ denotes the squared Frobenius norm of a matrix, $G_1 \in \Re_+^{n_1 \times c_1}$ and $G_2 \in \Re_+^{n_2 \times c_2}$ are the cluster indicator matrices for χ_1 and χ_2 respectively, and $S_{12} \in \Re_+^{c_1 \times c_2}$ absorbs the different scales of R_{12}, G_1 and G_2. Simultaneous clustering of χ_1 and χ_2 is then achieved by solving Eq. (1). Because the rows of resulted G_k ($k \in \{1, 2\}$) (with normalization) can be interpreted as the posterior probability for clustering on χ_k, the cluster label of x_i^k is obtained by:

$$l(x_i^k) = \operatorname{argmax}_j G_{k(ij)}. \qquad (2)$$

As can be seen, the data points enter the objective function in Eq. (1) as squared residual errors. Thus, outlying data samples can easily dominate the objective function because of the squared errors. To solve the problem, Liu and Wang [2015] proposed to replace the traditional squared Frobenius norm into ℓ_1-norm and optimize the following objective:

$$J = \|R_{12} - G_1 S_{12} G_2^T\|_1, \quad s.t. \quad G_1 \ge 0, G_2 \ge 0, \qquad (3)$$

where $\|X\|_1$ is defined as $\sum_{i=1}^{n} \sum_{j=1}^{m} |X^{(ij)}|$ for a matrix $X \in \Re^{n \times m}$.

2.2 Simultaneously Clustering Multi-Type Relational Data Using Inter-Type Relationships

A natural way to generalize the co-clustering objective in Eq. (1) to simultaneously cluster multi-type relational data is to solve the following optimization problem:

$$\begin{aligned} \min\ J = \sum_{0 < i,j \le K} \|R_{ij} - G_i S_{ij} G_j^T\|_1, \\ s.t. \quad G_i \ge 0, G_i^T G_i = I, \ \forall\, 0 < i \le K. \end{aligned} \qquad (4)$$

Although Eq. (4) generalizes Eq. (1) to deal with multi-type relational data, it is not straightforward to solve Eq. (4) by generalizing existing NMTF algorithms. As a contribution of this paper, we will derive the solution to Eq. (4).

We first introduce the following useful lemma.

Lemma 1 *The optimization problem in Eq. (3) can be equivalently solved by the following problem:*

$$\min\ J = \|R - GSG^T\|_1, \quad s.t. \quad G \ge 0, \qquad (5)$$

in which

$$\begin{aligned} R &= \begin{bmatrix} 0^{n_1 \times n_1} & R_{12}^{n_1 \times n_2} \\ R_{21}^{n_2 \times n_1} & 0^{n_2 \times n_2} \end{bmatrix}, \\ G &= \begin{bmatrix} G_1^{n_1 \times c_1} & 0^{n_1 \times c_2} \\ 0^{n_2 \times c_1} & G_2^{n_2 \times c_2} \end{bmatrix}, \quad S = \begin{bmatrix} 0^{c_1 \times c_1} & S_{12}^{c_1 \times c_2} \\ S_{21}^{c_2 \times c_1} & 0^{c_2 \times c_2} \end{bmatrix}, \end{aligned} \qquad (6)$$

where the superscripts denote the matrix sizes, and $R_{12} = R_{21}^T$, $S_{21} = S_{12}^T$. $0^{n_1 \times n_1}$ is a matrix with all zero entries of size $n_1 \times n_1$.

Proof 1 *Following the definitions of R, G and S, we can derive*

$$\begin{aligned} \|R - GSG^T\|_1 &= \left\| \begin{bmatrix} 0 & R_{12} \\ R_{12}^T & 0 \end{bmatrix} - \begin{bmatrix} 0 & P \\ P^T & 0 \end{bmatrix} \right\|_1 \\ &= 2\|R_{12} - P\|_1, \end{aligned}$$

where $P = G_1 S_{12} G_2^T$, which proves the lemma.

Based upon Lemma 1, we have the following theorem.

Theorem 1 *It is equivalent to solve Eq. (4) and to solve*

$$\min\ J = \|R - GSG^T\|_1, \quad s.t. \quad G \ge 0, \qquad (7)$$

in which

$$\begin{aligned} R &= \begin{bmatrix} 0^{n_1 \times n_1} & R_{12}^{n_1 \times n_2} & \cdots & R_{1K}^{n_1 \times n_K} \\ R_{21}^{n_2 \times n_1} & 0^{n_2 \times n_2} & \cdots & R_{2K}^{n_2 \times n_K} \\ \vdots & \vdots & \ddots & \vdots \\ R_{K1}^{n_K \times n_1} & R_{K2}^{n_K \times n_2} & \cdots & 0^{n_K \times n_K} \end{bmatrix}, \\ G &= \begin{bmatrix} G_1^{n_1 \times c_1} & 0^{n_1 \times c_2} & \cdots & 0^{n_1 \times c_K} \\ 0^{n_2 \times c_1} & G_2^{n_2 \times c_2} & \cdots & 0^{n_2 \times c_K} \\ \vdots & \vdots & \ddots & \vdots \\ 0^{n_K \times c_1} & 0^{n_K \times c_2} & \cdots & G_K^{n_K \times c_K} \end{bmatrix}, \\ S &= \begin{bmatrix} 0^{c_1 \times c_1} & S_{12}^{c_1 \times c_2} & \cdots & S_{1K}^{c_1 \times c_K} \\ S_{21}^{c_2 \times c_1} & 0^{c_2 \times c_2} & \cdots & S_{2K}^{c_2 \times c_K} \\ \vdots & \vdots & \ddots & \vdots \\ S_{K1}^{c_K \times c_1} & S_{K2}^{c_K \times c_2} & \cdots & 0^{c_K \times c_K} \end{bmatrix}, \end{aligned} \qquad (8)$$

where $R_{ji} = R_{ij}^T$ and $S_{ij} = S_{ji}^T$.

The proof of Theorem 1 can be easily obtained by generalizing the proof of Lemma 1 to multi-type relational data.

Theorem 1 presents a general framework via ℓ_1-norm *symmetric* NMTF (S-NMTF) for simultaneously cluster multi-type relational data using the mutual relationship matrices.

2.3 Incorporating Intra-Type Information via Graph Regularization

The optimization objectives in Eq. (4) and Eq. (7) only use the inter-type relationships of a multi-type relational data set, whereas the intra-type information, though often available, is not used. We can incorporate the intra-type relationship information through Laplacian regularization. For a multi-type relational data set, given the intra-type information in the form of the pairwise affinity matrices $W_1, W_2, \dots, W_K$ respectively, we can use them as following:

$$J = \sum_{0<i,j\le K} \|R_{ij} - G_i S_{ij} G_j^T\|_1 + \lambda \sum_{0<i\le K} \mathbf{tr}\left(G_i^T L_i G_i\right), \qquad s.t. \quad G_i \ge 0, G_i^T G_i = I, \; \forall\, 0 < i \le K, \tag{9}$$

where $L_k = D_k - W_k$ is the Laplacian matrix with $D_k^{(ii)} = \sum_j W_k^{(ij)}$. Because L_k is the discrete approximation of the Laplace-Beltrami operator on the underlying data manifold, the regularization term reflects the label smoothness of the two types of data points. The smoother the data labels are with respect to the underlying data manifolds, the smaller their values will be.

Using R, S and G defined in Eq. (8), denote

$$W = \begin{bmatrix} W_1^{n_1\times n_1} & 0^{n_1\times n_2} & \cdots & 0^{n_1\times n_K} \\ 0^{n_2\times n_1} & W_2^{n_2\times n_2} & \cdots & 0^{n_2\times n_K} \\ \vdots & \vdots & \ddots & \vdots \\ 0^{n_K\times n_1} & 0^{n_K\times n_2} & \cdots & W_K^{n_K\times n_K} \end{bmatrix}, \tag{10}$$

and $L = D - W$ where $D^{(ii)} = \sum_j W^{(ij)}$, it is easy to prove

$$\mathbf{tr}\left(G^T L G\right) = \sum_{0<i\le K} \mathbf{tr}\left(G_i^T L_i G_i\right). \tag{11}$$

Combining Theorem 1, we approach simultaneous clustering of multi-type relational data with manifold in Eq. (9) by solving the following optimization problem:

$$\min \; J = \|R - GSG^T\|_1 + \lambda\, \mathbf{tr}\left(G^T L G\right), \qquad s.t. \quad G \ge 0, \quad G^T G = I. \tag{12}$$

It can be verified that with constraint $G^T G = I$, then $G_{(i)}^T G_{(i)} = I$, $\forall\, 0 < i \le K$, which means the orthogonality of G will make membership matrices in each relational data orthogonal. The benefits of orthogonality constraint on G are two-fold: 1) get unique solution; 2) avoid soft clustering.

3 Difficulty in Solving the Problem with Traditional Method

Traditional methods to solve NMTF problem are based on MUA, where an auxiliary function is introduced to give an upper bound for the objective during every iteration. However, our proposed ℓ_1-norm objective function, different from traditional squared Frobenius norm which has nice mathematical properties, is non-differentiable. As a result, we cannot apply the traditional method to solve Eq. (12).

In our previous work [Liu and Wang, 2015], we attempted to solve the ℓ_1 norm by transforming it into a trace problem as:

$$\begin{aligned} J(G) &= \|R - GSG^T\|_1 \\ &= \mathbf{tr}\left((R - GSG^T)D(R - GSG^T)^T\right) \\ &= \mathbf{tr}\left(-2RDGSG^T + GSG^T DGSG^T\right). \end{aligned} \tag{13}$$

where D is a diagonal matrix defined as:

$$D(i,i) = \frac{\sum |R - GSG^T|_i}{\|R - GSG^T\|_i^2}. \tag{14}$$

By taking the derivative of J with respect to G and S, they get the solution following the traditional method. However, the derivation is not rigorous since D is a dependent variable on G and S, while they are suppose to be independent when we take derivatives.

As a contribution of our paper, we propose an algorithm which can give the optimal solution with rigorous mathematical guarantee. Moreover, a strictly orthogonal membership matrix will be given to avoid soft clustering while the objective loss is usually lower than MUA based methods.

4 The Solution Algorithm

Before giving our algorithm, we will first introduce the ADMM, which was proposed in [Bertsekas, 1996; Boyd *et al.*, 2011] to solve convex optimization problems by breaking them into smaller pieces that are easier to handle. Specifically, given the following objective with the equality constraint:

$$\min_{x,z} f(x) + g(z), \qquad s.t. \quad h(x,z) = 0, \tag{15}$$

Algorithm 1 solves the problem by decoupling it into subproblems and optimizing each variable while fixing others [Bertsekas, 1996; Boyd *et al.*, 2011], where y is the Lagrangian multiplier to the constraint h. It is worth noting that Algorithm 1 was proved to converge Q-linearly to the optimal solution [Bertsekas, 1996].

Algorithm 1: The ADMM algorithm.

```
Set $1 < \rho < 2$ and initialize $\mu > 0$ and $y$;
while not converge do
    1. Update $x$ by solving
       $x^{k+1} = \arg\min_x (f(x) + \frac{\mu}{2}\|h(x, z^k) + \frac{y^k}{\mu}\|^2)$;
    2. Update $z$ by solving
       $z^{k+1} = \arg\min_z (g(z) + \frac{\mu}{2}\|h(x^{k+1}, z) + \frac{y^k}{\mu}\|^2)$;
    3. Update $y$ by $y^{k+1} = y^k + \mu h(x^{k+1}, z^{k+1})$;
    4. Update $\mu$ by $\mu = \rho\mu$.
end
```

Huang *et al.* [2014] used the ADMM to solve the manifold regularized NMF problem. Following the same idea, we derive the solution algorithm to our objective.

Because there are two constraints (nonnegative and orthogonal) on G in our objective in Eq. (12), which is difficult to optimize, we introduce three auxiliary variables:

$F = G; P = F; H = G$. In addition, considering that the ℓ_1-norm objective is non-differentiable, we introduce one additional variable of $E = R - FSG^T$ to solve the following new objective function:

$$\begin{aligned}\min_{G,E,F,S} \; J = & \|E\|_1 + \lambda \, \mathbf{tr}\left(G^T L F\right) + \frac{\mu}{2}\|H - G + \frac{1}{\mu}\Omega\|_F^2 \\ & + \frac{\mu}{2}\|E - R + FSG^T + \frac{1}{\mu}\Lambda\|_F^2 \qquad (16) \\ & + \frac{\mu}{2}\|F - G + \frac{1}{\mu}\Sigma\|_F^2 + \frac{\mu}{2}\|P - F + \frac{1}{\mu}\Delta\|_F^2.\end{aligned}$$

The constraints in Eq. (12) now become: $G^T G = I, F^T F = I, H \geq 0, P \geq 0$, where each variable has only one constraint which is easier to be optimized.

Eq. (16) can be reduced to several manageable subproblems, where each subproblem yields a closed-form solution as following. We repeat the optimizing procedure until convergence, which is summarized in Algorithm 2.

Step 1. Solving S when fixing other variables:

$$\min_S \left\|\tilde{R} - FSG^T\right\|_{\mathrm{F}}^2, \qquad (17)$$

where we write $\tilde{R} = R - E - \frac{1}{\mu}\Lambda$ for brevity. Taking derivative of Eq. (17) with respect to S and setting it as 0, we can easily obtain the solution to this optimization problem as $S = \left(F^T F\right)^{-1} F^T \tilde{R} G \left(G^T G\right)^{-1}$.

Step 2. Solving E when fixing other variables:

$$\min_E \|E\|_{1,1} + \frac{\mu}{2}\|E - Z\|_{\mathrm{F}}^2, \qquad (18)$$

where we write $Z = R - FSG^T - \frac{1}{\mu}\lambda$ for brevity.

The optimization problem in Eq. (22) can be decoupled to solve the following problem for every entry of E:

$$\min_{e_{ij}} \frac{1}{2}\left(e_{ij} - z_{ij}\right)^2 + \frac{1}{\mu}|e_{ij}|. \qquad (19)$$

Taking derivative of Eq. (19) with respect to e_{ij} and setting it as 0, we can solve Eq. (19) as follows:

$$e_{ij} = \begin{cases} z_{ij} - \frac{1}{\mu} & \text{if} \quad z_{ij} > \frac{1}{\mu}; \\ z_{ij} + \frac{1}{\mu} & \text{if} \quad z_{ij} < -\frac{1}{\mu}; \\ 0 & \text{else.} \end{cases} \qquad (20)$$

Step 3. Solving **G** when fixing other variables:

$$\begin{aligned}\min_G \; & \frac{\mu}{2}\left\|E - R + FSG^T + \frac{1}{\mu}\Lambda\right\|_{\mathrm{F}}^2 + \frac{\mu}{2}\|H - G + \frac{1}{\mu}\Omega\|_F^2 \\ & + \frac{\mu}{2}\left\|F - G + \frac{1}{\mu}\Sigma\right\|_{\mathrm{F}}^2 + \lambda \, \mathbf{tr}\left(G^T L F\right) \\ & s.t. \quad G^T G = I. \qquad (21)\end{aligned}$$

By mathematical derivation, the problem in Eq. (21) can be rewritten as follows:

$$\max_G \mathbf{tr}\left(G^T M\right), \quad s.t. \quad G^T G = I, \qquad (22)$$

where we write $M = \left(R - E - \frac{1}{\mu}\Lambda\right)^T FS + \left(F + \frac{1}{\mu}\Sigma + H + \frac{1}{\mu}\Omega - \frac{\lambda}{\mu}LF\right)$ for brevity.

According to [Wang *et al.*, 2013], the problem in Eq. (22) can be solved by computing the SVD of M: if $svd(M) = UAV^T$, the solution of Eq. (22) is given by UV^T.

Step 4. Solving F when fixing other variables, similar to optimize G, the subproblem is equal to:

$$\max_F \mathbf{tr}\left(F^T M\right), \quad s.t. \quad F^T F = I, \qquad (23)$$

where we write $M = \left(R - E - \frac{1}{\mu}\Lambda\right) GS^T + \left(G - \frac{1}{\mu}\Sigma + P + \frac{1}{\mu}\Delta - \frac{\lambda}{\mu}LG\right)$ for brevity. By computing the SVD of M: $svd(M) = UAV^T$, the solution of F is given by UV^T.

Step 5. Solving H when fixing other variables:

$$\min_H \left\|H - G + \frac{1}{\mu}\Omega\right\|_{\mathrm{F}}^2, \quad s.t. \quad H \geq 0. \qquad (24)$$

and the solution is $H = \max\left(G - \frac{1}{\mu}\Omega, 0\right)$.

Step 6. Solving P when fixing other variables, similar to optimize H, $P = \max\left(F - \frac{1}{\mu}\Delta, 0\right)$.

Step 7. Update Lagrangian Multipliers $\Lambda, \Sigma, \Delta, \Omega$.

Step 8. Update μ.

Algorithm 2: The solution algorithm to our objective.

```
Data: Multi-relational data: ${R^1, R^2, ..., R^{n_K}}$,
      Number of Clusters $K$ and set $1 \leq \rho \leq 2$.
Result: Factor matrices: $G$.
1. Construct $R$ and $W$ with ${R^1, R^2, ..., R^{n_K}}$;
2. Initialize $G, \Lambda, \Sigma, \Delta, \Omega$;
while not converge do
    3. Optimize $G$, $S$, $F$, $E$, $P$, $H$ as Eq. (17–24);
    4. Update Lagrangian Multipliers $\Lambda, \Sigma, \Delta, \Omega$;
    5. Update $\mu = \rho\mu$.
end
6. Get the clustering result from $G$.
```

We can get the corresponding solution by changing ℓ_1 norm to traditional squared Frobenius norm. The only difference is to update E. When it is squared Frobenius norm, we can solve the subproblem at every entry of E as following:

$$\min_{e_{ij}} \frac{1}{2}\left(e_{ij} - z_{ij}\right)^2 + \frac{1}{\mu}\left(e_{ij}\right)^2. \qquad (25)$$

Taking derivative of Eq. (25) with respect to e_{ij} and setting it as 0, we can solve Eq. (25) as follows:

$$e_{ij} = \frac{\mu}{2 + \mu} z_{ij}. \qquad (26)$$

Due to space limit, we skipped some details in above derivations, which will be supplied in our extended journal version of this paper.

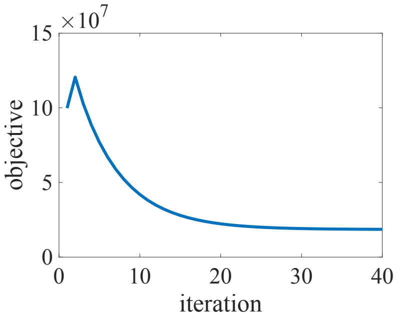

Figure 1: Objective value with updates.

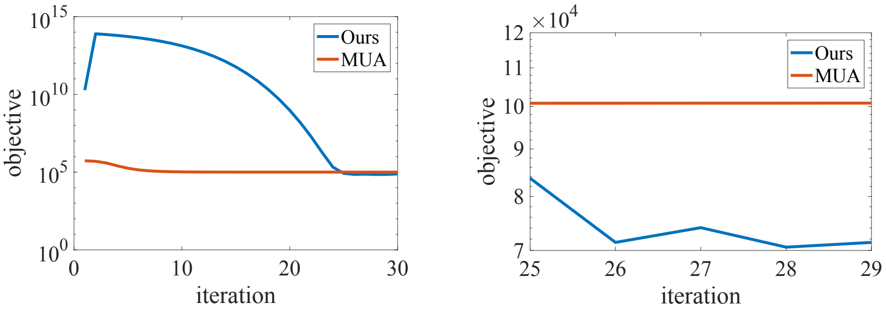

Figure 2: Left: Objective comparison between our proposed method and MUA based algorithm. Right: Zoom in iteration 25 to 29, showing ours can eventually get a lower local minimum.

5 Experiments

Although multi-relational data can help to improve clustering, there is no such existing real-world data set which can be directly used for experiments. For our experiments, we created a new multi-relational information from *ACL-IMDB* data set.

5.1 Multi-Relational Data Set and Its Collection

In the *ACL-IMDB* [Maas *et al.*, 2011] data set, there are 25000 reviews for highly popular movies, in which positive and negative comments come up with one half (12500) each. In our experiments, we build three relational matrices: comment-word, user-comment and user-word respectively.

Comment-word matrix. Since the comments are scored from 1 to 10 points (positive from 7 and 10 while negative from 1 to 4) with detailed content (word as the unit or feature), we can build the comment-word matrix along with its corresponding labels (ground truth), which will be used for sentiment analysis (comment clustering) of our proposed algorithm. We first rule out the stopwords (stopwords, such as *the*, *here*, *etc*., are words which are common across documents and contribute little to the content of the document), and then get the top 700 most commonly used words such as *cool*, *great*, *etc*. as the features.

User-comment matrix. Comments from the same person by using the same words are more likely to belong to the same cluster (positive or negative). Since each comment in the data set has its corresponding URL, of which we can make use to identify the author. Therefore we can build the user-comment matrix.

User-word matrix. Given the comment-word and user-comment matrix respectively, the user-word matrix can be obtained by multiplying the two matrices described above.

The three matrices create the multi-relational data, which are expected help improve the clustering performance. It is worth noting that the generated comment-word matrix is noisy due to some inevitable post typos. To make our experiments more convincing, we add some noise to the three relationship matrices with a ratio up to 25% (up to 20% in amplitude). By randomly choosing 500 authors from the top 1500 who post most comments, we can generate sub-data sets to conduct our experiments.

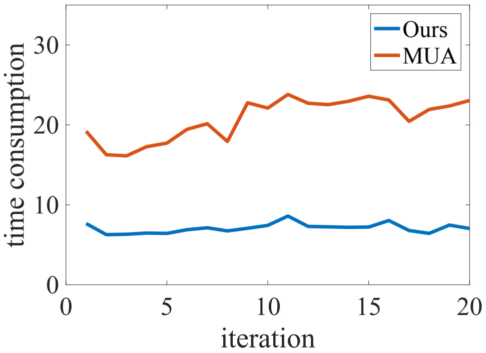

Figure 3: Time consumption in each iteration: comparison between our proposed method and ℓ_1-norm symmetric NMF based on MUA.

5.2 Convergence of the Algorithm

Because our solution algorithm employs an iterative method, we first study its convergence property. From Fig. 1 we can see that, though there is an increase in the first update, it decreases sharply later. In our experiment, we initialize μ as 0.02 and keep updating it by $\mu = \rho\mu$ during the iterations to accelerate the decrease. we find that if μ is fixed to be very small such as 0.001, the decrease will be slow; while if it is large such as 0.5, the decrease will be insignificant, trapped into a poor local minimum.

In addition, we compare clustering method using the squared Frobenius norm loss function (solved by the ADMM method by updating E using Eq. (26)) against the same objective solved by the MUA based method in [Wang *et al.*, 2011a]. As demonstrated in Fig. 2, our proposed method can get a lower minimum that may lead to a better clustering result.

5.3 Time Consumption

We also compare the time consumption between our proposed ℓ_1-norm algorithm against that in [Liu and Wang, 2015] (even though the proof is not rigorous and the loss does not decrease monotonically as expected) . We find that our algorithm is faster during each iteration, and it only takes 1/3 to 1/2 time as the counterpart does. We experiment with different sizes of matrices, and find our method is always faster as Fig. 3 shows.

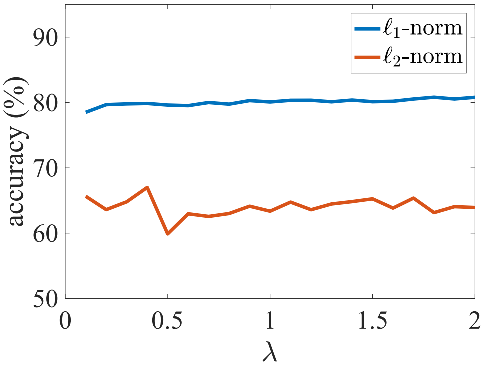

Figure 4: Clustering accuracies comparison of ℓ_1-norm objective with ℓ_2-norm based on our proposed ADMM method with regularization parameter changes.

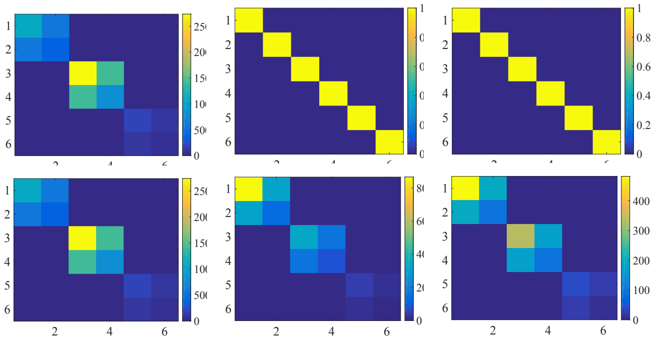

Figure 5: Heatmap of G^TG. Top row: our proposed method; Bottom row: MUA-based algorithm. The two methods are given the same initialized G as the left column images show, and during the update demonstrated as central and right column images, our method always yield strict orthogonal clustering matrix G.

5.4 Parameter Study and Robustness of ℓ_1-norm Objective Function

We incorporate the intra-type data through manifold regularization. We study the influence of regularization item to clustering accuracy by changing λ from 0.1 to 2 with a step size of 0.1, and find that ℓ_1-norm objective function remains robust with parameter changes, and its accuracy is constantly higher than the counterpart as Fig. 4 demonstrates. In our new method, we strictly enforce the orthogonal constraints onto the clustering membership matrix G, which plays an important role in getting rid of soft clustering (weights to different clusters are close) may lead to more distinguishable in clustering result. As we can see from Fig. 5, our solution of G satisfies the orthogonality constraint. In contrast, methods based on traditional MUA all fail to yield a strict orthogonal clustering matrix, which results in many *soft clustering* cases.

5.5 Clustering Result

We compare our proposed methods with some typical algorithms w.r.t. Accuracy (Acc), Normalized Mutual Information (NMI) [Xu *et al.*, 2003] and Adjusted Rand Index [Yeung and Ruzzo, 2001]:

K-means: We conduct K-means on comment-word and user-comment matrix respectively and take the best result.

ONMTF: Ding *et al.* [2006] solves symmetric matrix tri-factorization with orthogonal constraint while ignoring the intra-type data.

GNMF: By making use of graph, Cai *et al.* [2011] imposes the graph regularization to improve clustering.

HOCC: We only utilize two types of relational data (comment-word matrix) as Eq. (9) (where $K = 2$) for clustering, while ignoring other relational type data.

ℓ_2-MUA: Wang *et al.* [2011a] solves ℓ_2-norm HOCC with traditional MUA based method.

ℓ_1-MUA: Liu *et al.* [2015] solves ℓ_1-norm HOCC which is supposed to be robust with noise and outliers.

ℓ_2-ADMM: Our proposed ℓ_2-norm objective algorithm which is sensitive to outliers.

ℓ_1-ADMM: Our proposed ℓ_1-norm objective algorithm.

In Table 1, we change the number of users to generate symmetric matrices with different dimensions, and run the algorithms for 20 times and take the average for comparison. We also add some artificial noise to the data as mentioned in Section 5.1. In Table 2, the top section are clustering accuracies and NMI with different matrix sizes without noise while the middle and bottom sections show the clustering results before and after the noise added. We see that, in most cases, our method can get the best clustering result.

Due to space limit, we cannot report the experimental results on two-type relational data, which will be supplied in our extended journal version of this paper.

6 Conclusion

In this paper, we explored the high-order co-clustering problem and solved with the symmetric nonnegative matrix tri-factorization method. Different from traditional squared error loss, we proposed a non-differentiable ℓ_1-norm objective to stay robust with both sample and feature outliers. Instead of following traditional MUA method, we propose an ADMM-based algorithm and compare our method with other methods on real-world multi-relational data set we collect. All experiment results demonstrate that our methods are superior in terms of less time consumption, lower local minimum, higher clustering accuracy, NMI and Adjusted Rand Index, as well as strictly orthogonal membership matrix to get rid of *soft clustering*. Our method is flexible and could also be a new framework to solve non-smooth, non-convex problems with promising preliminary experimental results.

References

[Bertsekas, 1996] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 1996.

[Boyd *et al.*, 2011] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[Cai *et al.*, 2011] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative ma-

#users	#comments	K-means	ONMTF	GNMF	HOCC	ℓ_2-MUA	ℓ_1-MUA	ℓ_2-ADMM	ℓ_1-ADMM
50	315	0.0038	0.0053	0.0130	0.8401	0.8940	0.8949	0.8998	**0.9100**
80	387	-0.0025	0.0002	0.0034	0.8030	0.8688	0.8692	0.8707	**0.8889**
120	569	-0.0014	-0.0007	0.0058	0.7412	0.7958	0.7970	0.8007	**0.8258**
200	1061	-0.0002	0.0036	0.0073	0.5942	0.6217	0.6274	0.6359	**0.6551**
300	1609	-0.0035	-0.0022	0.0015	0.4848	0.5050	0.5358	0.5473	**0.5647**
400	2120	-0.0024	0.0001	0.0071	0.4023	0.4125	0.4246	0.4794	**0.5001**
500	2569	0.0088	0.0123	0.0264	0.3388	0.3617	0.3863	0.3912	**0.4164**

Table 1: Adjusted Rand Index of different algorithms on different size subsets.

K-means		ONMTF		GNMF		HOCC		ℓ_2-MUA		ℓ_1-MUA		ℓ_2-ADMM		ℓ_1-ADMM	
Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI	Acc	NMI
0.5029	0.008	0.5118	0.011	0.5203	0.012	0.6152	0.272	0.7061	0.5226	0.7166	0.5784	0.6679	0.5613	**0.8207**	**0.6526**
0.5152	0.006	0.5198	0.017	0.5186	0.011	0.6111	0.249	0.6952	0.5334	0.7152	0.5861	0.6594	0.5557	**0.8107**	**0.6494**
0.5495	0.021	0.5489	0.018	0.5452	0.015	0.6488	0.267	0.6549	0.5600	0.6878	0.6199	0.6603	0.5279	**0.8043**	**0.6477**
0.5769	0.025	0.5789	0.019	0.5450	0.010	0.6556	0.274	0.6714	0.5621	0.7002	0.6493	0.6622	0.5334	**0.7929**	**0.6451**
0.5789	0.030	0.5833	0.021	0.5232	0.012	0.5748	0.273	0.6748	0.5638	**0.7957**	0.6291	0.6374	0.5481	0.7895	**0.6445**
0.5708	0.026	0.5769	0.020	0.5178	0.011	0.5607	0.270	0.6474	0.5630	0.6881	0.6148	0.6218	0.5275	**0.7854**	**0.6422**
0.6087	0.031	0.6126	0.025	0.5627	0.013	0.6338	0.276	0.6306	0.5301	0.7628	**0.6495**	0.6515	0.5435	**0.8064**	0.6426
0.5838	0.002	0.6061	0.022	0.5623	0.013	0.6242	0.271	0.6296	0.5275	0.7616	0.6444	0.6335	0.5318	**0.8006**	**0.6483**

Table 2: The clustering performance of different algorithms on noiseless testing data set (top) and noise data set (the rest).

trix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

[Ding *et al.*, 2006] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[Huang *et al.*, 2014] Jin Huang, Feiping Nie, Heng Huang, and Chris Ding. Robust manifold nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3):11, 2014.

[Kong *et al.*, 2011] Deguang Kong, Chris Ding, and Heng Huang. Robust nonnegative matrix factorization using l21-norm. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 673–682. ACM, 2011.

[Lee and Seung, 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[Lin, 2007] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[Liu and Wang, 2015] Kai Liu and Hua Wang. Robust multi-relational clustering via l1-norm symmetric nonnegative matrix factorization. In *ACL (2)*, pages 397–401, 2015.

[Long *et al.*, 2006] Bo Long, Zhongfei Mark Zhang, Xiaoyun Wu, and Philip S Yu. Spectral clustering for multi-type relational data. In *Proceedings of the 23rd international conference on Machine learning*, pages 585–592. ACM, 2006.

[Maas *et al.*, 2011] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[Wang *et al.*, 2011a] Hua Wang, Heng Huang, and Chris Ding. Simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (ACM CIKM 2011)*, pages 279–284. ACM, 2011.

[Wang *et al.*, 2011b] Hua Wang, Feiping Nie, Heng Huang, and Chris Ding. Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation. In *The 11th IEEE International Conference on Data Mining series (ICDM 2011)*, pages 774–783. IEEE, 2011.

[Wang *et al.*, 2013] Hua Wang, Feiping Nie, and Heng Huang. Multi-view clustering and feature learning via structured sparsity. In *International Conference on Machine Learning (ICML 2013)*, pages 352–360, 2013.

[Xu *et al.*, 2003] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

[Yeung and Ruzzo, 2001] Ka Yee Yeung and Walter L Ruzzo. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.