

Relative Pairwise Relationship Constrained Non-Negative Matrix Factorisation

Shuai Jiang^{1b}, Kan Li^{1b}, and Richard Yi Da Xu^{1b}

Abstract—Non-negative Matrix Factorisation (NMF) has been extensively used in machine learning and data analytics applications. Most existing variations of NMF only consider how each row/column vector of factorised matrices should be shaped, and ignore the relationship among pairwise rows or columns. In many cases, such pairwise relationship enables better factorisation, for example, image clustering and recommender systems. In this paper, we propose an algorithm named, Relative Pairwise Relationship constrained Non-negative Matrix Factorisation (RPR-NMF), which places constraints over relative pairwise distances amongst features by imposing penalties in a triplet form. Two distance measures, squared Euclidean distance and Symmetric divergence, are used, and exponential and hinge loss penalties are adopted for the two measures, respectively. It is well known that the so-called “multiplicative update rules” result in a much faster convergence than gradient descend for matrix factorisation. However, applying such update rules to RPR-NMF and also proving its convergence is not straightforward. Thus, we use reasonable approximations to relax the complexity brought by the penalties, which are practically verified. Experiments on both synthetic datasets and real datasets demonstrate that our algorithms have advantages on gaining close approximation, satisfying a high proportion of expected constraints, and achieving superior performance compared with other algorithms.

Index Terms—Non-negative matrix factorisation, multiplicative update rules, clustering, recommender systems

1 INTRODUCTION

COMPARED to conventional dimensionality reduction methods, such as Singular Value Decomposition (SVD), low rank Non-negative Matrix Factorisation (NMF), mostly solving an optimisation task, converges much faster when it comes down to large real-world data sets [1], [2], [3]. Thus NMF has been widely used in many applications [4], [5], and algorithms of this kind have been the research foci in many communities, such as image processing and recommender systems [5], [6], [7], [8], [9].

A seminal approach in NMF is the so-called “multiplicative update rules” which guarantees both the convergence of the algorithm and the non-negativity of factorised matrices [10]. Though the “multiplicative update rules” were proved not converging to a stationary point numerically [11], and they are not strictly well-defined because of possible zero entries [12], it practically produces satisfactory results, especially for large scale data, which makes it a popular solution for NMF. However, the original NMF only imposes the non-negativity constraints on both of the factorising and factorised matrices, which in practice may not be enough to satisfy additional

requirements. Thus, researchers in this area have been proposing new algorithms under this framework to cater for incremental improvements [13], [14], [15], variations [16], [17], [18], and/or application oriented constraints [19], [20].

A main sub category of NMF is Constrained Non-negative Matrix Factorisation (CNMF), which imposes constraints based on variables as regularisation terms [21]. The most commonly used regularisations for NMF are L1 norm and L2 norm, the former increases the sparseness of the factorised matrices, while the latter makes the results smooth to prevent overfitting. However, these constraints are only imposed on each of the rows or columns and has not considered the relationship among pairwise rows or columns (such as to specify constraints on relative distance between row vectors A and B and that of A and C). Such pairwise relationships exists wildly in many systems, especially in those systems where the factorised matrices are representing features. A typical instance is the matrix factorisation technique used in recommender systems.

In a recommender system, the factorising matrix is usually the rating matrix whose entries denote the ratings given by the corresponding user (row) to the corresponding item (column), and the factorised matrices are usually regarded as the user feature matrix (the left factorised matrix) and the item feature matrix (the right factorised matrix). Fig. 1 shows an example of a simple movie recommender system. In this example, after factorisation by NMF, the distance between “Star Wars” and “Titanic” becomes less than the distance between “Star Wars” and “Star Trek”. However, as we all know, the movie “Star Wars” should be closer to the movie “Star Trek”—both are within the scientific fiction genre, unlike “Titanic”, which is a love story. Thus intuitively, it will result in a better factorisation and make better recommendations if we can incorporate such human-aware relative relationships into the NMF model.

- S. Jiang is with the School of Computer Science, Beijing Institute of Technology (BIT), Beijing 100081, China, and the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Ultimo, NSW 2007, Australia. E-mail: jiangshuai@bit.edu.cn.
- K. Li is with the School of Computer Science, Beijing Institute of Technology (BIT), Beijing 100081, China. E-mail: likan@bit.edu.cn.
- R.Y.D. Xu is with the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Ultimo, NSW 2007, Australia. E-mail: yida.xu@uts.edu.au.

Manuscript received 13 Jan. 2018; revised 3 July 2018; accepted 15 July 2018.
Date of publication 24 July 2018; date of current version 3 July 2019.


(Corresponding author: Kan Li.)

Recommended for acceptance by D. Cai.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TKDE.2018.2859223

Movie User ID	Titanic	Star Wars	Star Trek	The Matrix	Avatar
0001	3.5 (3.5)	0 (3.0)	3.8 (3.8)	0 (1.4)	0 (2.8)
0002	0 (4.0)	3.7 (3.7)	0 (4.3)	0 (2.3)	0 (3.6)
0003	0 (3.8)	0 (4.0)	2.4 (2.4)	0 (1.5)	3.3 (3.3)
0004	2.8 (2.8)	3.1 (3.1)	0 (3.0)	3.2 (3.2)	0 (3.5)
0005	0 (3.1)	0 (3.4)	0 (2.9)	0 (3.0)	3.6 (3.6)



0001	0.34	0.51	1.87
0002	1.32	0.51	1.64
0003	0.29	1.94	0.73
0004	3.07	0.12	0.06
0005	2.71	0.52	0.12

Titanic	Star Wars	Star Trek	The Matrix	Avatar
0.83	0.93	0.92	1.02	1.06
1.34	1.51	0.44	0.45	1.17
1.36	1.03	1.75	0.41	1.00

dis(Star Wars, Titanic)	0.38	dis(Star Wars, Star Trek)	1.29
-------------------------	------	---------------------------	------

Fig. 1. A simple movie recommender system. The factorisation algorithm used here is NMF with Euclidean measure proposed in [10]. Original missing ratings are denoted by 0 and recovered ratings are showed in brackets. As shown under the right factorised matrix, the distance between feature vectors of movies “Star Wars” and “Titanic” is less than the distance between movies “Star Wars” and “Star Trek”.

There have been a few studies that attempt to consider the above pairwise relationship in the area of matrix factorisation. Although none of these have properly addressed our problem, two existing methods are still worth noting here: one is Graph Regularised Non-negative Matrix Factorisation (GNMF) [22], the other is Label Constrained Non-negative Matrix Factorisation (LCNMF) [23].

GNMF constructs a weight matrix of the graph from the observed data, and then applies the weights (similarities) on the factorised low-dimensional data representation as regularisations. It works well on image clustering applications where the data points in different classes are distinctively different. However, in many other cases, such as where the data points are not spread, GNMF cannot guarantee the relative relationship denoted by similarities retained as expected after factorisation. Besides, the setting of similarities is sensitive to the factorisation results, especially when the similarities cannot be simply set zeros and ones. LCNMF was proposed to cater for scenarios where partially labeled grouping data were made available. If two feature vectors are labeled into the same class, they are assumed to have the same feature representation in the latent space. This approach has addressed the need of applications on image clustering, however, such a setting is far too restrictive in general: “Star Wars” and “Star Trek” could be very similar to each other, but setting their features identical is unacceptable and impractical.

In this paper, we propose a novel matrix factorisation algorithm, called RPR-NMF. Rather than using explicit similarities or previously known labels, RPR-NMF imposes penalties for relative pairwise relationships (RPRs) in a triplet form. The penalties are not limited to be within $[0, 1]$ as for similarities or to be binary values as for labels. To make our algorithm generic and flexible, we impose penalties on both factorised matrices, and in a setting where only constraining one matrix is needed, we can simply “turn-off” the constraints of the second. Both of the squared Euclidean distance and the symmetric divergence measure are used in the objective of RPR-NMF, and the penalties are in exponential and hinge loss forms respectively. The update rules for RPR-NMF conform to the well-known “multiplicative update rules” in which the proofs of convergence are essential for the algorithm. Due to the complexity of proof brought by the imposed penalties, we approximate partial terms in the proofs and have verified its practical benefits through numerous experiments.

Compared with the existing methods GNMF and LCNMF, RPR-NMF can guarantee more pairwise relationships retained after factorisation. Fig. 2 gives a demonstration of the RPRs among four data points after running GNMF, LCNMF, and our proposed algorithm RPR-NMF using Euclidean measure respectively. In this example, GNMF failed on retaining one RPR (points ② and ④ should be closer than

points ③ and ④), LCNMF projected all points onto one, while RPR-NMF retains all RPRs after factorisation.

The main contributions of this paper are:

1. We propose an algorithm named RPR-NMF, which utilises relative pairwise relationship among rows or columns of factorised matrices to achieve a better factorisation with high constraint satisfied rate and close approximation simultaneously. Different from GNMF and LCNMF, RPR-NMF can guarantee the expected RPRs retained after factorisation and does not limit the feature vectors to be identical.
2. In the method section, we use different forms of penalties for Euclidean measure and Divergence measure based on our observations on method convergence through numerous experiments. For the Euclidean measure, we incorporate the RPRs in natural exponential functions, while for the Divergence measure, we use hinge loss function.
3. The solution of RPR-NMF conforms to that of “multiplicative rules”, in which we provide complete and sufficient proofs for both of the distance measures with the help of relaxation on partial terms. Such relaxation has been verified reasonable and practical through our experiments.
4. The complexity analysis shows that RPR-NMF does not increase much of processing time by introducing penalty terms when comparing to NMF. Synthetic and real datasets experiments both demonstrate that RPR-NMF have advantages on close approximation, high constraint satisfied rate and outstanding application performance.

The rest of this paper is organised as follows: in the next section, we review related literature to our work. The Method section contains the details of our algorithm and the proof of convergence, followed by the Experiments section in which we evaluate our algorithm as well as the Conclusions.

2 RELATED WORK

NMF was first proposed to solve the following optimisation problem: given a non-negative matrix V , find non-negative matrix factors W and H such that $V \approx WH$. Since the seminal work of [10] which has proposed the so-called, “multiplicative update rules” for non-negative matrix factorisation, a number of approaches followed suit dealing with various NMF issues from different aspects. Our work falls under the category of the Constrained Non-negative Matrix Factorisation (CNMF) which was first proposed in [21]. In general, it has the following representation:

$$\min_{W, H} \{\|V - WH\|_F^2 + \alpha J_1(W) + \beta J_2(H)\}, \quad (1)$$

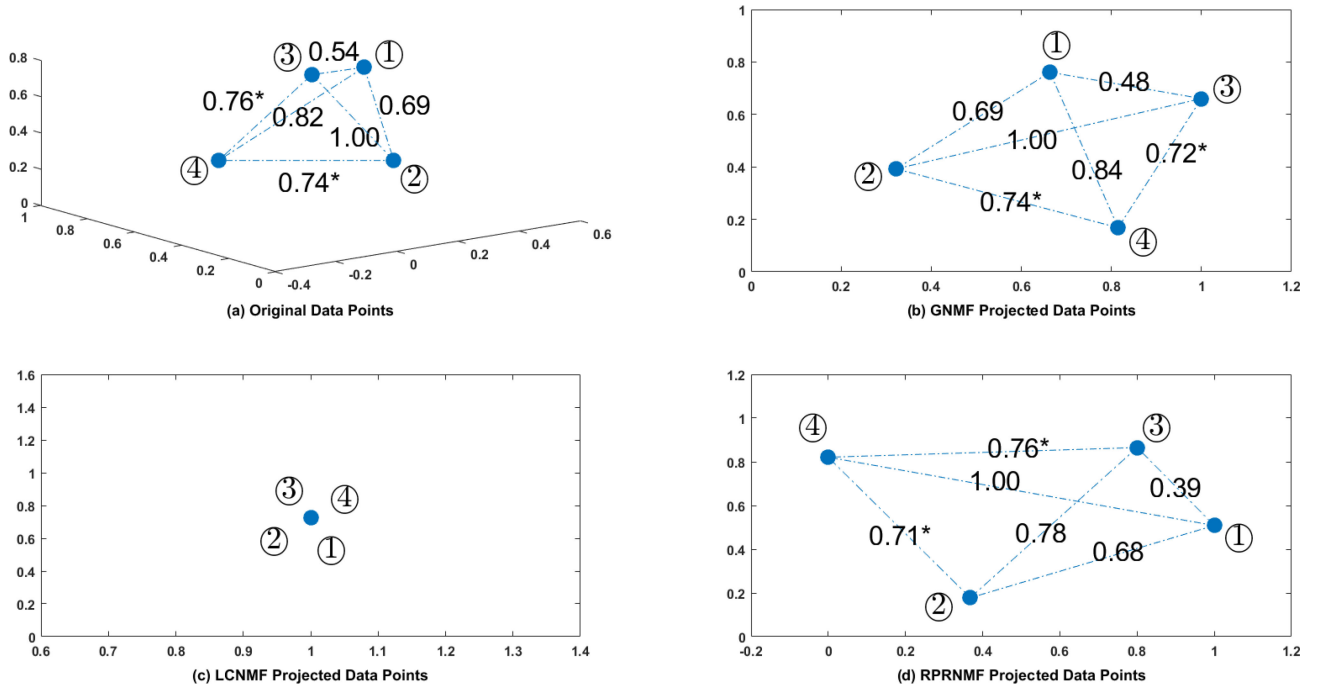


Fig. 2. An example of projecting four 3D data points into 2D space. Points are ordered by circled numbers, and the value near each dotted line is the normalised Euclidean distance. (a) The original data space. The distance between point ② and point ④ is less than the distance between point ③ and point ④ (marked by asterisk sign). (b) Projected data points by GNMF using Euclidean distance between each pair of original data points as dissimilarity matrix. The distance between point ② and point ④ becomes bigger than the distance between point ③ and point ④. (c) LCNMF projects all four points onto one when considering all RPRs. (d) RPR-NMF retains all the RPRs after factorisation.

for $W \geq 0$ and $H \geq 0$, where $\|\cdot\|_F$ is the Frobenius norm, α and β are regularisation coefficients. The functions of $J_1(W)$ and $J_2(H)$ are penalty terms used to enforce certain constraints on the solution of Eq. (1). In their work, the penalties are set $J_1(W) = \|W\|_F^2$ and $J_2(H) = \|H\|_F^2$ in order to enforce the smoothness in W and H respectively.

Many studies followed the above CNMF framework. Jia and Qian [24] used an adaptive potential function as penalties to characterise the piecewise smoothness of spectral data. Li et al. [25] imposed three additional constraints on the NMF basis to reveal local features. Shen and Si [26] imposed both L1 and L2 norms to control the shape of base matrix and increase the sparseness of the coefficient matrix so as to enhance the clustering performance on multiple manifolds.

Besides the above studies based on CNMF that consider how to constrain the value of each vector in factorised matrices, there are two algorithms that take the relationship among vectors of factorised matrix into consideration: Graph Regularised Non-negative Matrix Factorisation (GNMF) proposed by [22], and Labelled Constrained Non-negative Matrix Factorisation (LCNMF) proposed by [23].

GNMF, also a CNMF algorithm, is to utilise the relationship among factorised rows or columns for a dimensionality reduction issue. It is in an effort to ensure that similarities between data points at the original space are also retained after they are transformed to the low dimensional subspace through factorisation. Its objective function with Euclidean measure is as following:

$$\mathcal{F} = \|V - WH\|_F^2 + \lambda \text{Tr}(HLH^T), \quad (2)$$

where L is a graph Laplacian matrix obtained from the similarity matrix and λ is a regularisation coefficient. As showed in the objective function, GNMF tries to minimise two parts simultaneously: the squared errors between the product of factorised matrices and the factorising matrix, and the similarity matrix. The ideal solution is to minimise them at the same time. However, it often happens that if GNMF minimised the squared error, the RPRs implied by similarities might not be satisfied. As shown in Fig. 2, the RPR among points ②, ③ and ④ was opposite to that when they were in the original 3D space. Our proposed RPR-NMF works in a different way: it imposes penalties with respect to the expected RPRs, which forces the factorised feature vectors to keep as many of the expected RPRs as possible.

LCNMF also utilises the relationship among factorised vectors. Instead of imposing regularised constraints, it represents the relationship by altering the factorising structure. Thus it is not a CNMF method. LCNMF uses partial label information as hard constraints and turns the original NMF task into a semi-supervised problem: they represent the right factorised matrix by a product of a class matrix and a reduced feature matrix where the class matrix contains binary entries to divide data into a predefined number of classes. Its objective function with Euclidean measure is as

$$\mathcal{F} = \|V - WYB\|_F^2, \quad (3)$$

where Y is the reduced feature matrix and B is the class matrix. The method however, assumes that if two data points have the same label, their corresponding feature vectors must be identical, as showed in Fig. 2 where all four data points are labelled the same and are projected onto one point after factorisation. Such constraints are too restrictive

under many general settings. For example, if two movies are by the same director, in the same genre and even feature the same actors, setting their features identical is to ignore any difference between them, which is what our method aims to mitigate.

3 METHOD

In this section, we introduce a new factorisation algorithm when RPR constraints are in place, called RPR-NMF. Note that both GNMF and LCNMF only impose constraints on the right factorised matrix because they were proposed as data dimensionality reduction methods. For generality, RPR-NMF imposes constraints on both factorised matrices, and it is trivial to only impose constraints on one factorised matrix.

Consider a dataset represented by a non-negative $N \times M$ matrix V . This matrix is then approximately factorised into an $N \times K$ matrix W and a $K \times M$ matrix H , where K is usually set to be smaller than both N and M , and is commonly referred to as the latent dimension. The RPR constraints placed on the factorised matrices can be defined as two sets of integer indexed triples

$$L_W \subseteq \{(q, r, s) | q, r, s \in \mathbb{N}^+, q, r, s \leq N, q \neq r \neq s\}, \quad (4)$$

$$L_H \subseteq \{(q, r, s) | q, r, s \in \mathbb{N}^+, q, r, s \leq M, q \neq r \neq s\}. \quad (5)$$

Specifically, each triple represents the relative relationship between two pairs of vectors with one sharing vector. In our work, if W was the matrix in question and l th triple specified the distance between vector q and r to be less than the distance between vectors q and s , the relationship could be denoted as $dis(W_{q^l}, W_{r^l}) < dis(W_{q^l}, W_{s^l})$ where W_{q^l} is the q th row vector of matrix W , and $dis(x, y)$ measures the distance between vectors x and y . We follow the most commonly used two distance measures in our work, which are the squared Euclidean distance

$$E(x, y) = \|x - y\|^2, \quad (6)$$

and the Divergence (when the variables are two unit vectors/distributions, it becomes KL-Divergence)

$$D(x||y) = \sum_{i=1}^K x_i \log \frac{x_i}{y_i} - x_i + y_i. \quad (7)$$

Since the Divergence of two vectors is not symmetric ($D(x||y) \neq D(y||x)$), when characterising the imposed RPR constraints, we use the Symmetric Divergence defined as

$$\begin{aligned} SD(x, y) &= \frac{1}{2}(D(x||y) + D(y||x)) \\ &= \frac{1}{2} \sum_{i=1}^K (x_i - y_i) \log \frac{x_i}{y_i}. \end{aligned} \quad (8)$$

Then the constraints are incorporated as penalty terms in the objective function. In our work, the penalty format for Euclidean measure is of an addition of natural exponential functions, while that for Divergence measure is of a hinge loss function. The reason for not using the same format of penalties is that the Divergence measure with exponential penalties cannot guarantee a high proportion

of satisfied constraints after factorisation, and that the Euclidean measure with hinge loss penalties cannot steadily converge. Thus with two independent coefficients λ_W and λ_H , we define the objective function using Euclidean distance as

$$\begin{aligned} \mathcal{F}_1 &= \|V - WH\|_F^2 \\ &+ \lambda_W \sum_{l=1}^{l_W} [\exp(E(W_{q^l}, W_{r^l})) + \exp(-E(W_{q^l}, W_{s^l}))] \\ &+ \lambda_H \sum_{l=1}^{l_H} [\exp(E(H_{:q^l}, H_{:r^l})) + \exp(-E(H_{:q^l}, H_{:s^l}))] \\ s.t. \quad &\lambda_W \geq 0, \lambda_H \geq 0, \forall i, j, W_{ij} \geq 0, H_{ij} \geq 0, \end{aligned} \quad (9)$$

and the objective function using Divergence as

$$\begin{aligned} \mathcal{F}_2 &= D(V||WH) \\ &+ \lambda_W \sum_{l=1}^{l_W} \max(0, SD(W_{q^l}, W_{r^l}) - SD(W_{q^l}, W_{s^l})) \\ &+ \lambda_H \sum_{l=1}^{l_H} \max(0, SD(H_{:q^l}, H_{:r^l}) - SD(H_{:q^l}, H_{:s^l})) \\ s.t. \quad &\lambda_W \geq 0, \lambda_H \geq 0, \forall i, j, W_{ij} \geq 0, H_{ij} \geq 0, \end{aligned} \quad (10)$$

where l_W and l_H are the numbers of constraints.

3.1 Solving Objective Functions

To solve the above objective functions, we need to derive the update rules for W and H . As a matter of fact, the penalties are not convex even when fixing one of the matrix factors. However, we found it is still feasible to obtain the update rules by constructing and solving the corresponding Lagrange functions. Once we obtained the updating rules, we could minimise the objective functions by iteratively updating W and H .

3.1.1 Updating Rules for Euclidean Measure

As for the objective function in Eq. (9), we first construct a Lagrange function with non-negative constraints $W_{ij} \geq 0$ and $H_{ij} \geq 0$

$$\begin{aligned} \mathcal{L}_1 &= \|V - WH\|_F^2 + \sum_{ij} \alpha_{ij} W_{ij} + \sum_{ij} \beta_{ij} H_{ij} \\ &+ \lambda_W \sum_{l=1}^{l_W} [\exp(E(W_{q^l}, W_{r^l})) + \exp(-E(W_{q^l}, W_{s^l}))] \\ &+ \lambda_H \sum_{l=1}^{l_H} [\exp(E(H_{:q^l}, H_{:r^l})) + \exp(-E(H_{:q^l}, H_{:s^l}))]. \end{aligned} \quad (11)$$

The partial derivative with respect to W_{ab} is

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial W_{ab}} &= 2[-(VH^T)_{ab} + (WHH^T)_{ab} \\ &+ \lambda_W C_{row}(W_{ab})] + \alpha_{ab}, \end{aligned} \quad (12)$$

where

$$\begin{aligned}
 C_{row}(W_{ab}) &= \sum_{l=1}^{l_W} \left\{ \exp(E(W_{q^l}, W_{r^l})) \left[\sum_{q^l=a} (W_{q^l b} - W_{r^l b}) + \sum_{r^l=a} (W_{r^l b} - W_{q^l b}) \right] \right. \\
 &\quad \left. - \exp(-E(W_{q^l}, W_{s^l})) \left[\sum_{q^l=a} (W_{q^l b} - W_{s^l b}) + \sum_{s^l=a} (W_{s^l b} - W_{q^l b}) \right] \right\} \\
 &= \sum_{l=1}^{l_W} \left(\exp(E(W_{q^l}, W_{r^l})) \left(\sum_{q^l=a} W_{q^l b} + \sum_{r^l=a} W_{r^l b} \right) \right. \\
 &\quad \left. + \exp(-E(W_{q^l}, W_{s^l})) \left(\sum_{q^l=a} W_{s^l b} + \sum_{s^l=a} W_{q^l b} \right) \right) \\
 &\quad - \sum_{l=1}^{l_W} \left(\exp(E(W_{q^l}, W_{r^l})) \left(\sum_{q^l=a} W_{r^l b} + \sum_{r^l=a} W_{q^l b} \right) \right. \\
 &\quad \left. + \exp(-E(W_{q^l}, W_{s^l})) \left(\sum_{q^l=a} W_{q^l b} + \sum_{s^l=a} W_{s^l b} \right) \right) \\
 &= C_{row}^+(W_{ab}) - C_{row}^-(W_{ab}).
 \end{aligned} \tag{13}$$

Let the partial derivative vanish and considering the non-negative constraints ($\alpha_{ab} W_{ab} = 0$ under K.K.T conditions), we obtain

$$\begin{aligned}
 ((VH^T)_{ab} + \lambda_W C_{row}^-(W_{ab})) W_{ab} \\
 - ((WHH^T)_{ab} + \lambda_W C_{row}^+(W_{ab})) W_{ab} = 0,
 \end{aligned} \tag{14}$$

thus the update rule for W_{ab} is formulated as following:

$$W_{ab} \leftarrow W_{ab} \frac{(VH^T)_{ab} + \lambda_W C_{row}^-(W_{ab})}{(WHH^T)_{ab} + \lambda_W C_{row}^+(W_{ab})}. \tag{15}$$

Similarly, we have the update rule for H_{ab} as

$$H_{ab} \leftarrow H_{ab} \frac{(W^T V)_{ab} + \lambda_H C_{col}^-(H_{ab})}{(W^T W H)_{ab} + \lambda_H C_{col}^+(H_{ab})}. \tag{16}$$

As for the update rules, we have the following theorem:

Theorem 1. *The objective function \mathcal{F}_1 in Eq. (9) is non-increasing under the update rules in Eqs. (15) and (16) with appropriate penalty coefficients λ_W and λ_H .*

We provide the prove of the convergence for the above updating rules and Theorem 1 in Section 3.2.1.

3.1.2 Updating Rules for Divergence Measure

As for the objective function in Eq. (10), we first construct a Lagrange function with non-negative constraints $W_{ij} \geq 0$ and $H_{ij} \geq 0$

$$\begin{aligned}
 \mathcal{L}_2 &= D(V||WH) + \sum_{ij} \alpha_{ij} W_{ij} + \sum_{ij} \beta_{ij} H_{ij} \\
 &\quad + \lambda_W \sum_{l=1}^{l_W} \max(0, SD(W_{q^l}, W_{r^l}) - SD(W_{q^l}, W_{s^l})) \\
 &\quad + \lambda_H \sum_{l=1}^{l_H} \max(0, SD(H_{:q^l}, H_{:r^l}) - SD(H_{:q^l}, H_{:s^l})).
 \end{aligned} \tag{17}$$

The partial derivative with respect to W_{ab} is

$$\begin{aligned}
 \frac{\partial \mathcal{L}_2}{\partial W_{ab}} &= \sum_j \left(H_{bj} - \frac{V_{aj} H_{bj}}{(WH)_{aj}} \right) \\
 &\quad + \frac{1}{2} \lambda_W P_{row}(W_{ab}) + \alpha_{ab},
 \end{aligned} \tag{18}$$

where

$$\begin{aligned}
 P_{row}(W_{ab}) &= \sum_{l=1}^{l_W} \left\{ \sum_{q^l=a} [g(W_{q^l b}^+, W_{r^l b}^+) + g(W_{q^l b}^+, W_{s^l b}^+)] \right. \\
 &\quad \left. + \sum_{r^l=a} g(W_{r^l b}^+, W_{q^l b}^+) - \sum_{s^l=a} g(W_{s^l b}^+, W_{q^l b}^+) \right\},
 \end{aligned} \tag{19}$$

$$W_{q^l b}^+ = \begin{cases} 0, & \text{if the constraint } l \text{ is satisfied} \\ W_{q^l b} & \text{otherwise} \end{cases} \tag{20}$$

$$g(x, y) = \log \frac{x}{y} + (x - y) \frac{1}{x}. \tag{21}$$

Let the partial derivative equal to zero as well as considering the non-negative constraints ($\alpha_{ab} W_{ab} = 0$ under K.K.T conditions), we obtain

$$-W_{ab} \sum_j \frac{V_{aj} H_{bj}}{(WH)_{aj}} + W_{ab} \left(\frac{1}{2} \lambda_W P_{row}(W_{ab}) + \sum_j H_{bj} \right) = 0, \tag{22}$$

thus the update rule for W_{ab} is formulated as following:

$$W_{ab} \leftarrow W_{ab} \frac{\sum_j V_{aj} H_{bj} / (WH)_{aj}}{\frac{1}{2} \lambda_W P_{row}(W_{ab}) + \sum_j H_{bj}}. \tag{23}$$

Similarly, we have the update rule for H_{ab} as

$$H_{ab} \leftarrow H_{ab} \frac{\sum_i V_{ib} W_{ia} / (WH)_{ib}}{\frac{1}{2} \lambda_H P_{col}(H_{ab}) + \sum_i W_{ia}}. \tag{24}$$

Notice that the value of $P_{row}(W_{ab})$ and $P_{col}(H_{ab})$ may be negative during updating. Thus if the denominator in an iteration is less than zero, then we abandon the penalty parts. Besides, we dynamically change the penalty coefficients to ensure the convergence since the update rules are obtained by approximation (see proof of convergence in Section 3.2.2).

As for the update rules, we have the following theorem:

Theorem 2. *The objective function \mathcal{F}_2 in Eq. (10) is non-increasing under the update rules in Eqs. (23) and (24) with appropriate penalty coefficients λ_W and λ_H .*

We provide the prove of the convergence for the above updating rules and Theorem 2 in Section 3.2.2.

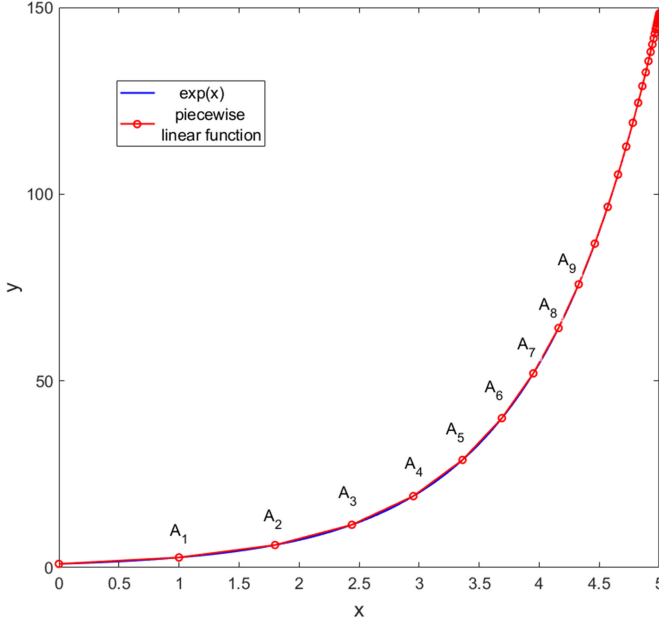


Fig. 3. Using a piecewise linear function to approximate e^x .

3.2 Proofs of Convergence and Theorems

We construct an auxiliary function $\mathcal{G}(x, x')$ to help prove Theorems 1 and 2, which satisfies the conditions $\mathcal{G}(x, x') \geq \mathcal{F}(x)$ and $\mathcal{G}(x, x) = \mathcal{F}(x)$, and guarantees $\mathcal{F}(x)$ to be non-increasing under the following update:

$$x^{t+1} = \arg \min_x \mathcal{G}(x, x'). \quad (25)$$

We illustrate our proofs for W_{ab} and the proofs for H_{ab} can be derived in a similar fashion. Let $\mathcal{F}_{ab}(W)$ denote part of $\mathcal{F}(W)$ concerning W_{ab} and so as $\mathcal{G}_{ab}(W, W^t)$.

3.2.1 Convergence of Euclidean Updating Rules and Theorem 1

As for updating rules in Eqs. (15) and (16), we have

Lemma 3. *Function*

$$\begin{aligned} \mathcal{G}_{ab}(W, W^t) &= \mathcal{F}_{ab}(W^t) + (W_{ab} - W_{ab}^t) \mathcal{F}'_{ab}(W^t) \\ &+ (W_{ab} - W_{ab}^t)^2 \frac{(W^t H H^T)_{ab} + \lambda_W C_{row}^+(W_{ab}^t)}{W_{ab}^t}, \end{aligned} \quad (26)$$

is an auxiliary function for $\mathcal{F}_{ab}(W)$.

Proof. $\mathcal{G}_{ab}(W, W) = \mathcal{F}_{ab}(W)$ is obvious. Applying Taylor Expansion to $\mathcal{F}_{ab}(W)$ on the point W_{ab}^t , we obtain

$$\begin{aligned} \mathcal{F}_{ab}(W) &\approx \mathcal{F}_{ab}(W^t) + (W_{ab} - W_{ab}^t) \mathcal{F}'_{ab}(W^t) \\ &+ \frac{1}{2} (W_{ab} - W_{ab}^t)^2 \mathcal{F}''_{ab}(W^t). \end{aligned} \quad (27)$$

Note that $\mathcal{O}((W_{ab}^t)^3)$ is omitted because we use a linear piecewise function to approximate the exponential penalties (see proofs below). Comparing Eqs. (26) and (27), in order to prove $\mathcal{G}_{ab}(W, W^t) \geq \mathcal{F}_{ab}(W)$, we only need to prove

$$\frac{2[(W^t H H^T)_{ab} + \lambda_W C_{row}^+(W_{ab}^t)]}{W_{ab}^t} \geq \mathcal{F}''_{ab}(W^t). \quad (28)$$

Rewrite $\mathcal{F}(W)$ as an addition of two functions (omitting the penalties for H since it is irrelevant with W)

$$\begin{aligned} \mathcal{F}(W) &= \|V - WH\|_F^2 \\ &+ \lambda_W \sum_{l=1}^L [\exp(E(W_{q^l}, W_{r^l})) + \exp(-E(W_{q^l}, W_{s^l}))] \\ &= \mathcal{U}(W) + \lambda_W \mathcal{V}(W). \end{aligned} \quad (29)$$

Then Eq. (28) becomes

$$\frac{\mathcal{U}'_{ab}(W^t) + \lambda_W \mathcal{V}'_{ab}(W^t)}{W_{ab}^t} \geq \mathcal{U}''_{ab}(W^t) + \lambda_W \mathcal{V}''_{ab}(W^t), \quad (30)$$

where $\mathcal{V}'_{ab}(W^t)$ is the positive part of $\mathcal{V}'_{ab}(W^t)$. Since

$$\begin{aligned} \frac{\mathcal{U}'_{ab}(W^t)}{W_{ab}^t} - \mathcal{U}''_{ab}(W^t) &= \frac{(W^t H H^T)_{ab}}{W_{ab}^t} - (H H^T)_{bb} \\ &= \frac{\sum_{k=1}^K W_{ak}^t (H H^T)_{kb}}{W_{ab}^t} - (H H^T)_{bb} \\ &\geq \frac{W_{ab}^t (H H^T)_{bb}}{W_{ab}^t} - (H H^T)_{bb} \\ &\geq 0, \end{aligned} \quad (31)$$

the only remaining work is to prove

$$\frac{\mathcal{V}'_{ab}(W^t)}{W_{ab}^t} \geq \mathcal{V}''_{ab}(W^t). \quad (32)$$

Recall that the function $\mathcal{V}(W)$ is a summation of additions of two natural exponential functions, and the exponents are quadratic functions of W . Calculation of their derivatives is intricate and causes difficulties to prove the above inequality. Here we give a proof by applying approximations to the natural exponential functions.

For the general natural exponential function $f(x) = e^x$ with support $[0, +\infty)$ under partition A , we can use a piecewise function to approximate it

$$e^x \approx \begin{cases} \frac{(e^{A_1}-1)}{A_1} x + 1, & x \in [0, A_1) \\ \frac{(e^{A_2}-e^{A_1})}{A_2-A_1} x + \frac{A_2 e^{A_1}-A_1 e^{A_2}}{A_2-A_1}, & x \in [A_1, A_2) \\ \vdots \\ \frac{(e^{A_n}-e^{A_{n-1}})}{A_n-A_{n-1}} x + \frac{A_n e^{A_{n-1}}-A_{n-1} e^{A_n}}{A_n-A_{n-1}}, & x \in [A_{n-1}, A_n), \\ \vdots \end{cases} \quad (33)$$

where each sub-function is a linear function of x . The relationship between $f(x)$ and the piecewise function is demonstrated in Fig. 3. Note that A should have smaller partitions as the curve increasing. In our implementation, we also normalise the factorising matrix to make the exponential penalties as small as possible. Similarly, we can approximate $g(x) = e^{-x}$ with a piecewise function

whose n th sub-function is

$$\frac{(e^{-A_{n-1}} - e^{-A_n})}{A_{n-1} - A_n}x + \frac{A_{n-1}e^{-A_n} - A_n e^{-A_{n-1}}}{A_{n-1} - A_n}, x \in [A_{n-1}, A_n]. \quad (34)$$

Now we can approximate $\mathcal{V}(W)$ using the above two piecewise functions. The n th part of it is

$$\begin{aligned} \mathcal{V}(W)^{(n)} \approx & \sum_{l=1}^{l_W} \frac{1}{A_n - A_{n-1}} \left\{ (e^{A_n} - e^{A_{n-1}})E(W_{q^l}, W_{r^l}) \right. \\ & + (e^{-A_n} - e^{-A_{n-1}})E(W_{q^l}, W_{s^l}) + A_n(e^{A_{n-1}} + e^{-A_{n-1}}) \\ & \left. - A_{n-1}(e^{A_n} + e^{-A_n}) \right\}. \end{aligned} \quad (35)$$

Its first and second order derivatives w.r.t. W_{ab} are

$$\begin{aligned} \mathcal{V}'_{ab}(W)^{(n)} \approx & \sum_{l=1}^{l_W} \frac{1}{A_n - A_{n-1}} \left\{ (e^{A_n} - e^{A_{n-1}}) \right. \\ & \left[\sum_{q^l=a} (W_{q^l b} - W_{r^l b}) + \sum_{r^l=a} (W_{r^l b} - W_{q^l b}) \right] + (e^{-A_n} - e^{-A_{n-1}}) \\ & \left[\sum_{q^l=a} (W_{q^l b} - W_{s^l b}) + \sum_{s^l=a} (W_{s^l b} - W_{q^l b}) \right] \left. \right\}, \end{aligned} \quad (36)$$

$$\begin{aligned} \mathcal{V}''_{ab}(W)^{(n)} \approx & \sum_{l=1}^{l_W} \frac{1}{A_n - A_{n-1}} \left\{ (e^{A_n} - e^{A_{n-1}}) \left(\sum_{q^l=a} 1 + \sum_{r^l=a} 1 \right) \right. \\ & \left. + (e^{-A_n} - e^{-A_{n-1}}) \left(\sum_{q^l=a} 1 + \sum_{s^l=a} 1 \right) \right\}. \end{aligned} \quad (37)$$

And the positive part of the first derivative is

$$\begin{aligned} \mathcal{V}^+_{ab}(W)^{(n)} \approx & \sum_{l=1}^{l_W} \frac{1}{A_n - A_{n-1}} \left\{ (e^{A_n} - e^{A_{n-1}}) \right. \\ & \left(\sum_{q^l=a} W_{q^l b} + \sum_{r^l=a} W_{r^l b} \right) - (e^{-A_n} - e^{-A_{n-1}}) \\ & \left(\sum_{q^l=a} W_{s^l b} + \sum_{s^l=a} W_{q^l b} \right) \left. \right\}. \end{aligned} \quad (38)$$

Then we have

$$\begin{aligned} \frac{\mathcal{V}^+_{ab}(W^t)^{(n)}}{W^t_{ab}} - \mathcal{V}''_{ab}(W^t)^{(n)} &= \sum_{l=1}^{l_W} \frac{1}{A_n - A_{n-1}} \left\{ \right. \\ & (e^{-A_{n-1}} - e^{-A_n}) \left(\sum_{q^l=a} \left(\frac{W^t_{s^l b}}{W^t_{q^l b}} + 1 \right) + \sum_{s^l=a} \left(\frac{W^t_{q^l b}}{W^t_{s^l b}} + 1 \right) \right) \left. \right\} \quad (39) \\ &\geq 0. \end{aligned}$$

Thus Eq. (28) holds. \square

Now we can prove the convergence of Theorem 1:

Proof of Theorem 1. According to Eqs. (25) and (26), we get

$$\begin{aligned} W^{t+1}_{ab} &= \arg \min_{W_{ab}} \mathcal{G}_{ab}(W, W^t) \\ &= W^t_{ab} \frac{(V H^T)_{ab} + \lambda_W C_{row}^-(W_{ab})}{(W H H^T)_{ab} + \lambda_W C_{row}^+(W_{ab})}. \end{aligned} \quad (40)$$

As $\mathcal{G}_{ab}(W, W^t)$ is an auxiliary function, $\mathcal{F}_{ab}(W)$ is non-increasing under this update rule. \square

3.2.2 Convergence of Divergence Updating Rules and Theorem 2

As for updating rules in Eqs. (23) and (24), we have

Lemma 4. Function

$$\begin{aligned} \mathcal{G}_{ab}(W, W^t) &= \sum_j (V_{aj} \log V_{aj} - V_{aj}) + \sum_j (W H)_{aj} \\ &\quad - \sum_{jk} V_{aj} \frac{W^t_{ak} H_{kj}}{(W^t H)_{aj}} \left(\log W_{ak} H_{kj} - \log \frac{W^t_{ak} H_{kj}}{(W^t H)_{aj}} \right) \\ &\quad + \lambda_W \sum_{l=1}^{l_W} \max(0, SD(W_{q^l}, W_{r^l}) - SD(W_{q^l}, W_{s^l})) \\ &\quad + \lambda_H \sum_{l=1}^{l_H} \max(0, SD(H_{:q^l}, H_{:r^l}) - SD(H_{:q^l}, H_{:s^l})). \end{aligned} \quad (41)$$

This is an auxiliary function for $\mathcal{F}_{ab}(W)$.

Proof. $\mathcal{G}_{ab}(W, W) = \mathcal{F}_{ab}(W)$ is obvious. To prove that $\mathcal{G}_{ab}(W, W^t) \geq \mathcal{F}_{ab}(W)$, we utilize the Jensen Inequality to obtain

$$-\log \sum_k W_{ak} H_{kj} \leq -\sum_k \alpha_k \log \frac{W_{ak} H_{kj}}{\alpha_k}, \quad (42)$$

which holds for all non-negative α_k that sum to unity. Setting

$$\alpha_k = \frac{W^t_{ak} H_{kj}}{(W^t H)_{aj}}, \quad (43)$$

we obtain

$$\begin{aligned} & -\sum_j V_{aj} \log \sum_k W_{ak} H_{kj} \\ & \leq -\sum_{jk} V_{aj} \frac{W^t_{ak} H_{kj}}{(W^t H)_{aj}} \left(\log W_{ak} H_{kj} - \log \frac{W^t_{ak} H_{kj}}{(W^t H)_{aj}} \right). \end{aligned} \quad (44)$$

This is the only different part between $\mathcal{F}_{ab}(W)$ and $\mathcal{G}_{ab}(W, W^t)$. Thus, $\mathcal{F}_{ab}(W) \leq \mathcal{G}_{ab}(W, W^t)$ holds. \square

Now we can prove the convergence of Theorem 2:

Proof of Theorem 2. According to Eqs. (25) and (41), we get

$$\begin{aligned} W^{t+1}_{ab} &= \arg \min_{W_{ab}} \mathcal{G}_{ab}(W, W^t) \\ &\approx W^t_{ab} \frac{\sum_j V_{aj} H_{bj} / (W H)_{aj}}{\frac{1}{2} \lambda_W P_{row}(W_{ab}) + \sum_j H_{bj}}. \end{aligned} \quad (45)$$

Notice that W also exists in the penalty regularisation terms, and it is difficult to calculate their corresponding derivatives with respect to W . For simplicity and efficiency, we substitute W in penalties with W^t so that they

become irrelevant to W . This approximation of the local optima of $\mathcal{G}_{ab}(W, W^t)$ is numerically proved feasible through our experiments.

As $\mathcal{G}_{ab}(W, W^t)$ is an auxiliary function, $\mathcal{F}_{ab}(W)$ is non-increasing under this update rule. \square

The complete algorithms of RPR-NMF are demonstrated in Algorithms 1 and 2.

Algorithm 1. Relative Pairwise Relationship Constrained Non-Negative Matrix Factorisation (RPR-NMF) Using Euclidean Measure

Input: $V \in \mathbb{R}_{\geq 0}^{N \times M}$ (factorising matrix),
 $K \in \mathbb{N}^+$ (latent dimension),
 $L_W \in \mathbb{N}^{I_W \times 3}$ (pairwise relationship constraints on W),
 $L_H \in \mathbb{N}^{I_H \times 3}$ (pairwise relationship constraints on H),
 $\lambda_W \in \mathbb{R}^+$, $\lambda_H \in \mathbb{R}^+$ (penalty coefficients)

Output: $W \in \mathbb{R}_{\geq 0}^{N \times K}$ (the left factorised matrix),
 $H \in \mathbb{R}_{\geq 0}^{K \times M}$ (the right factorised matrix)

- 1: Initialise W and H as non-negative random matrices
- 2: **while** Terminal conditions not satisfied **do**
- 3: **for** $k = 1$ **to** K **do**
- 4: // Update W
- 5: **for** $a = 1$ **to** N **do**
- 6: Calculate $C_{row}^+(W_{ab})$ and $C_{row}^-(W_{ab})$ as Eq. (13)
- 7: $W_{ak} \leftarrow W_{ak} \frac{(VH)_{ak} + \lambda_W C_{row}^-(W_{ab})}{(WHH^T)_{ak} + \lambda_W C_{row}^+(W_{ab})}$
- 8: **end for**
- 9: // Update H
- 10: **for** $b = 1$ **to** M **do**
- 11: Calculate $C_{col}^+(H_{ab})$ and $C_{col}^-(H_{ab})$ as Eq. (13)
- 12: $H_{kb} \leftarrow H_{kb} \frac{(W^T V)_{kb} + \lambda_H C_{col}^-(H_{ab})}{(W^T WH)_{kb} + \lambda_H C_{col}^+(H_{ab})}$
- 13: **end for**
- 14: **end for**
- 15: **end while**

4 EXPERIMENTS

In this section, we conducted experiments on both synthetic datasets and real datasets to demonstrate the performance of our proposed algorithm RPR-NMF. The baseline algorithms we chose for comparison are:

- Fast and robust recursive algorithms for separable non-negative matrix factorisation (FSNMF) [14]. It was proposed to solve NMF with separability assumption.
- Alternating Non-negative Least Squared (ANLS) method for non-negative matrix factorisation [27]. This is a generic approach for NMF by solving multiple non-negative least squared tasks when fixing one of the factorised matrices.
- Non-negative matrix factorisation using multiplicative updating rules (NMF) [10]. It is the basic framework solving NMF by multiplicative rules.
- Graph regularised non-negative matrix factorisation (GNMF) [22]. It imposes similarity constraints as penalty regularisation to maintain the proximity among data.
- Label constrained non-negative matrix factorisation (LCNMF) [23]. This method uses partial labels of data to ensure the corresponding columns in the right factorised matrix to be identical.

Algorithm 2. Relative Pairwise Relationship Constrained Non-Negative Matrix Factorisation (RPR-NMF) Using Divergence Measure

Input: $V \in \mathbb{R}_{\geq 0}^{N \times M}$ (factorising matrix),
 $K \in \mathbb{N}$ (latent dimension),
 $L_W \in \mathbb{N}^{I_W \times 3}$ (pairwise relationship constraints on W),
 $L_H \in \mathbb{N}^{I_H \times 3}$ (pairwise relationship constraints on H),
 $\lambda_W \in \mathbb{R}^+$, $\lambda_H \in \mathbb{R}^+$ (penalty coefficients)

Output: $W \in \mathbb{R}_{\geq 0}^{N \times K}$ (the left factorised matrix),
 $H \in \mathbb{R}_{\geq 0}^{K \times M}$ (the right factorised matrix)

- 1: Initialise W and H as non-negative random matrices
- 2: **while** Terminal conditions not satisfied **do**
- 3: **for** $k = 1$ **to** K **do**
- 4: // Update W
- 5: **for** $a = 1$ **to** N **do**
- 6: Calculate $P_{row}(W_{ab})$ as Eq. (19)
- 7: **if** $\frac{1}{2} \lambda_W P_{row}(W_{ab}) + \sum_j H_{kj} < 0$ **then**
- 8: $W_{ak} \leftarrow W_{ak} \frac{\sum_j V_{aj} H_{kj} / (WH)_{aj}}{\sum_j H_{kj}}$
- 9: **else**
- 10: $W_{ak} \leftarrow W_{ak} \frac{\sum_j V_{aj} H_{kj} / (WH)_{aj}}{\frac{1}{2} \lambda_W P_{row}(W_{ab}) + \sum_j H_{kj}}$
- 11: **end if**
- 12: **end for**
- 13: // Update H
- 14: **for** $b = 1$ **to** M **do**
- 15: Calculate $P_{col}(H_{ab})$ as Eq. (19)
- 16: **if** $\frac{1}{2} \lambda_H P_{col}(H_{ab}) + \sum_i W_{ik} < 0$ **then**
- 17: $H_{kb} \leftarrow H_{kb} \frac{\sum_i V_{ib} W_{ik} / (WH)_{ib}}{\sum_i W_{ik}}$
- 18: **else**
- 19: $H_{kb} \leftarrow H_{kb} \frac{\sum_i V_{ib} W_{ik} / (WH)_{ib}}{\frac{1}{2} \lambda_H P_{col}(H_{ab}) + \sum_i W_{ik}}$
- 20: **end if**
- 21: **end for**
- 22: **end for**
- 23: **if** Objective function value increases **then**
- 24: $\lambda_W \leftarrow 1/2 \cdot \lambda_W$, $\lambda_H \leftarrow 1/2 \cdot \lambda_H$
- 25: Roll back
- 26: **else**
- 27: $\lambda_W \leftarrow 1.01 \cdot \lambda_W$, $\lambda_H \leftarrow 1.01 \cdot \lambda_H$
- 28: **end if**
- 29: **end while**

FSNMF and ANLS only use Euclidean measure, while each of the other algorithms have two versions: one with Euclidean measure (denoted by suffix “_eu” in figures and tables), the other with Divergence measure (denoted by suffix “_div”).

Note that GNMF and LCNMF only have one regularisation term (they are both designed as dimensionality reduction methods), the weight matrix for GNMF only performs as constraints imposed on the right factorised matrix and the label matrix for LCNMF also only affects the right factorised matrix. Thus for fair comparison, RPR-NMF only has RPR constraints on the right factorised matrix in the experiments on synthetic datasets as well as datasets for image clustering. However, when it comes down to recommender systems, we impose constraints on both factorised matrices which are considered as users’ and items’ features.

The metrics we used to evaluate the performance of algorithms are:

1. *Mean Squared Loss (MSL)* for algorithms using Euclidean measure and *Mean Divergence (MD)* for algorithms using Divergence measure, which evaluate the approximation performance and are defined as

$$MSL = \frac{1}{NM} \|V - W^* H^*\|_F^2, \quad (46)$$

$$MD = \frac{1}{NM} D(V \| W^* H^*), \quad (47)$$

where W^* and H^* are the final factorised matrices.

2. *Constraint Satisfied Rate (CSR)*, which presents how well the RPR constraints are satisfied and is defined as

$$CSR = \frac{1}{2} (l_W^*/l_W + l_H^*/l_H), \quad (48)$$

where l_W^* and l_H^* are the numbers of satisfied constraints.

3. *Clustering Accuracy (ACC)*, which we calculate by constructing a cost matrix, solving the matrix by Munkres Assign Algorithm [28], and mapping one clustering to the other; and *Normalised Mutual Information (NMI)* [29], which is to calculate the entropy information correlation of clusterings. They are used to evaluate the performance of the clustering results in image clustering experiments.
4. *Root Squared Error (RMSE)* [30], which evaluates the difference between recovering ratings and original ratings (the latter are removed before factorisation in cross validation); and *F1 score* [31], which tells how much proportion of correct recommendations the recovered rating matrix gives. They are used to evaluate the performance on recommendation systems and defined as

$$RMSE = \left(\sum_{ij} M_{ij} \right)^{-1/2} \|M * (V - W^* H^*)\|_F, \quad (49)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (50)$$

where M is a binary matrix in which ones denote the chosen entries and zeros denote not chosen entries in a cross validation experiment. As for each user, we use its average rating as a threshold. Ratings greater than the threshold suggest the corresponding items are recommended. Thus we obtain two recommendation binary vectors before and after factorisation, which can be used to calculate the value of Precision and Recall.

We first introduce two algorithms used to convert the additional information. Then we validate the effectiveness of RPR-NMF and analyse the computing complexity through two synthetic experiments, followed by experiments on real datasets for applications of image clustering and recommender systems. The statistics of the datasets we used are presented in Table 1.

All the code of RPR-NMF as well as preprocessed datasets can be downloaded from: <https://github.com/shawn-jiang/RPRNMF>.

TABLE 1
Statistics of Datasets Used for Experiments

dataset	rows	columns	range	density
Synthetic 1	500	500	[0, 10]	1
Synthetic 2	50-10,000	50-10,000	[0, 10]	1
AT&T ORL	1,024	400	[0, 255]	1
CMU PIE	1,024	2,856	[0, 255]	1
MovieLens 1M	6,040	3,706	[1, 5]	0.0447

4.1 Additional Information Conversion

Since GNMF, LCNMF and RPR-NMF all need additional information besides the factorising matrix itself, it is necessary to make it fair for all the three algorithms to equally access available additional information.

GNMF needs a weight matrix which describes how close the data vectors should be, LCNMF requires which data vectors should have the same label, while RPR-NMF demands a list of RPR constraints among factorised vectors. These three additional information can be converted from each to the others. However, their intensities are different: labels of LCNMF are the strongest, followed by the similarities of GNMF, and the RPR constraints of RPR-NMF are the weakest. It will cause loss of information if we convert stronger constraints to weaker ones, but not if it is done the other way around. Thus here we introduce two algorithms which convert the RPR constraints for RPR-NMF to the weight matrix for GNMF (Algorithm 3) and the label matrix for LCNMF (Algorithm 4) respectively.

Algorithm 3. Constructing Weight Matrix for GNMF from Pairwise Relationship Constraints

Input: $M \in \mathbb{N}^+$ (the column dimension of H),
 $L_H \in \mathbb{N}^{l_H \times 3}$ (pairwise relationship constraints on H),
 $mins, maxs \in \mathbb{R}$ (minimal and maximal weight to set)
Output: $S_H \in \mathbb{R}_{\geq 0}^{M \times M}$ (weight matrix for GNMF)

- 1: $S_H \leftarrow I_M$
- 2: Construct a network initialised with zero node
- 3: **for** $l = 1$ **to** l_H **do**
- 4: $node_1 \leftarrow (L_H(l, 1), L_H(l, 2))$
- 5: $node_2 \leftarrow (L_H(l, 1), L_H(l, 3))$
- 6: Put the two nodes into the network (if they have not been there) with an directed edge from $node_1$ to $node_2$
- 7: **end for**
- 8: Recursively calculate the depth of each node (nodes with no outgoing edges are set depth 1, their direct parents are set depth 2, and so on).
- 9: $t \leftarrow (maxs - mins) / (\max(depth) - 1)$
- 10: **for** each node n **do**
- 11: $i, j \leftarrow$ two points of n
- 12: $S_H(i, j) \leftarrow mins + (depth(n) - 1) * t$
- 13: **end for**

4.2 Effectiveness Validation & Complexity Analysis

We conducted two experiments on synthetic datasets to verify the effectiveness and study the computing complexity of RPR-NMF comparing to the baseline algorithms. These experiments are conducted on a computer with i7-6900K CPU, 16G RAM and Windows 10.

It is worth noting that, the RPR constraints in the experiments are randomly generated but there are chains among

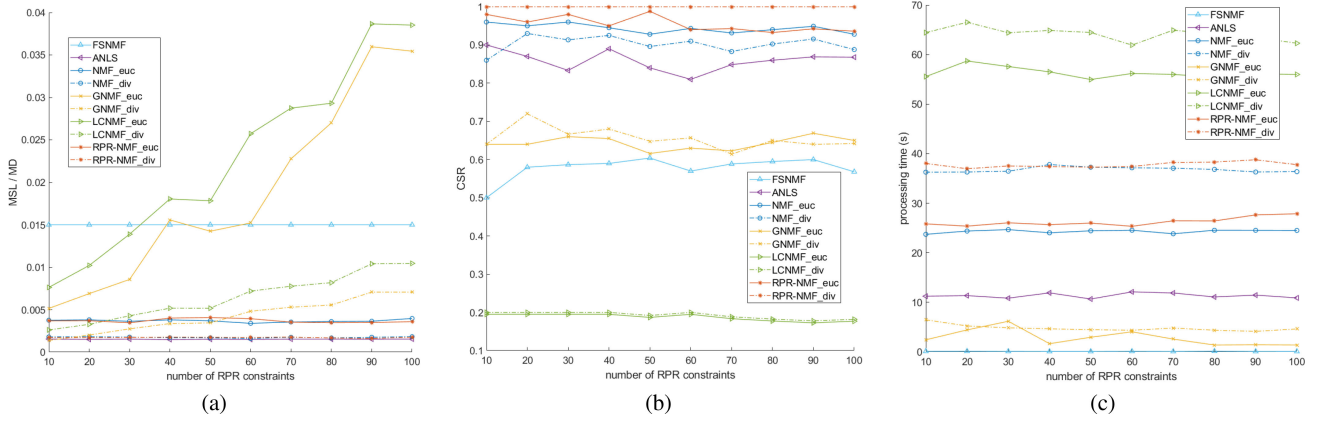


Fig. 4. Performance with respect to the number of pairwise relationship constraints. (a) MSL/MD, (b) CSR, and (c) Processing time.

them. A chain of constraints is like $dis(a, b) < dis(b, c) < dis(c, d)$, which is a 3-chain of constraints. As we are going to show in the results, RPR-NMF can satisfy chain constraints while others are incapable.

Algorithm 4. Constructing Label Matrix for LCNMF from Pairwise Relationship Constraints

Input: $M \in \mathbb{N}^+$ (the column dimension of H),
 $L_H \in \mathbb{N}^{l_H \times 3}$ (pairwise relationship constraints on H)
Output: B (binary matrix whose size is at most $M \times M$ but not determined)

- 1: $B \leftarrow 0_M$
- 2: **for** $l = 1$ **to** l_H **do**
- 3: $q \leftarrow L_H(l, 1)$ $r \leftarrow L_H(l, 2)$
- 4: **if** $B_{:,q} \neq 0 \wedge B_{:,r} \neq 0$ **then**
- 5: $B_{:,q} \leftarrow B_{:,q} \vee B_{:,r}$
- 6: $B_{:,r} \leftarrow 0$
- 7: **else if** $B_{:,q} \neq 0$ **then**
- 8: $label(r) \leftarrow label(q)$
- 9: **else if** $B_{:,r} \neq 0$ **then**
- 10: $label(q) \leftarrow label(r)$
- 11: **else**
- 12: $q, r \leftarrow \text{new label}$
- 13: **end if**
- 14: **end for**
- 15: Delete rows containing all zeros from B

In the first experiment, we explored the influence of the number of RPR constraints with respect to MSL/MD, CSR and processing time. First, we randomly generated a 500×20 matrix W_0 and a 20×500 matrix H_0 , and got the factorising matrix V by rescaling their product to be within $[0, 10]$. Then we generate 2 to 20 (with step 2) groups of 5-chain constraints (i.e., 10 to 100 constraints) from H_0 . K is set 20 and λ_H is set 1. For each experiment with different number of constraints, we repeated it 5 times and present the average results in Fig. 4.

As the number of constraints increases, (a) ANLS, NMF and RPR-NMF algorithms obtain very close and low errors, while the errors of GNMF and LCNMF are much higher and increasing as well, and the performance of FSNMF is quite stable; (b) RPR-NMF using Divergence measure achieves the highest CSRs (average 100.00 percent) followed by its Euclidean version (average 95.52 percent), ANLS and FSNMF obtain 85.89 and 57.82 percent respectively, while NMF, GNMF, LCNMF obtain 94.35/90.23 percent, 64.28/65.58 percent, 19.23/19.23 percent average CSR for Euclidean and

Divergence measures respectively. (c) the processing time of FSNMF is the lowest, while both LCNMF algorithms spend the longest time, and the time used by RPR-NMF is close to that used by NMF with slight increases. This demonstrates that comparing with GNMF and LCNMF, our proposed RPR-NMF not only maintains as low error and low time complexity as NMF, but also satisfies the most expected relationship constraints after factorisation.

The second experiment focused on how the size of factorising matrix affects the performance of different algorithms. The size of the factorising matrix V varies from 50×50 to $10,000 \times 10,000$. Each V is also obtained by the rescaled ($[0, 10]$) product of randomly generated W_0 and H_0 . For each matrix size, the latent dimension K is set 20, and the number of constraints is set $N/10$. The constraints are also of 5-chains. λ_H is set 1. The average results of 5 repeated experiments are presented in Fig. 5.

As the size of factorising matrix increases, (a) NMF and RPR-NMF algorithms obtain close and low errors again, while the errors of FSNMF and LCNMF using Euclidean measure are decreasing but still much higher than those of others; (b) RPR-NMF using Divergence measure achieves the highest score on CSR (100.00 percent on all cases), while FSNMF, ANLS, NMF and GNMF obtain higher CSR; (c) the processing time of all algorithms increases, and FSNMF is the fastest while LCNMF algorithms are the slowest. RPR-NMF spent nearly the same time as unconstrained NMF, despite the imposed RPR constraints. We do notice that FSNMF is significantly faster than all other algorithms because of its separability assumption. However, this assumption is too strong in many settings and does not generate good factorisation (see algorithm performance in Section 4.4).

The results on CSR also suggest that the weight matrix of GNMF and the labels of LCNMF are not helpful on satisfying such RPR constraints. When there are only independent constraints, GNMF and LCNMF can satisfy them by setting proper weights and labels (e.g., for a constraint $dis(a, b) < dis(b, c)$, GNMF sets weight 1 for (a, b) and 0 for (b, c) , and LCNMF sets (a, b) the same label). However, when the constraints are not independent (such as in chains), GNMF cannot simply set the weights 1s and 0s, and LCNMF can only guarantee satisfying one constraint of a group of chain constraints (e.g., for a 3-chain of constraints $dis(a, b) < dis(b, c) < dis(c, d)$, the weight of (b, c) in GNMF has to be greater than 0 and less than 1, while in LCNMF, setting (a, b) the same label can satisfy the first constraint, but the only way to satisfy the second constraint is to

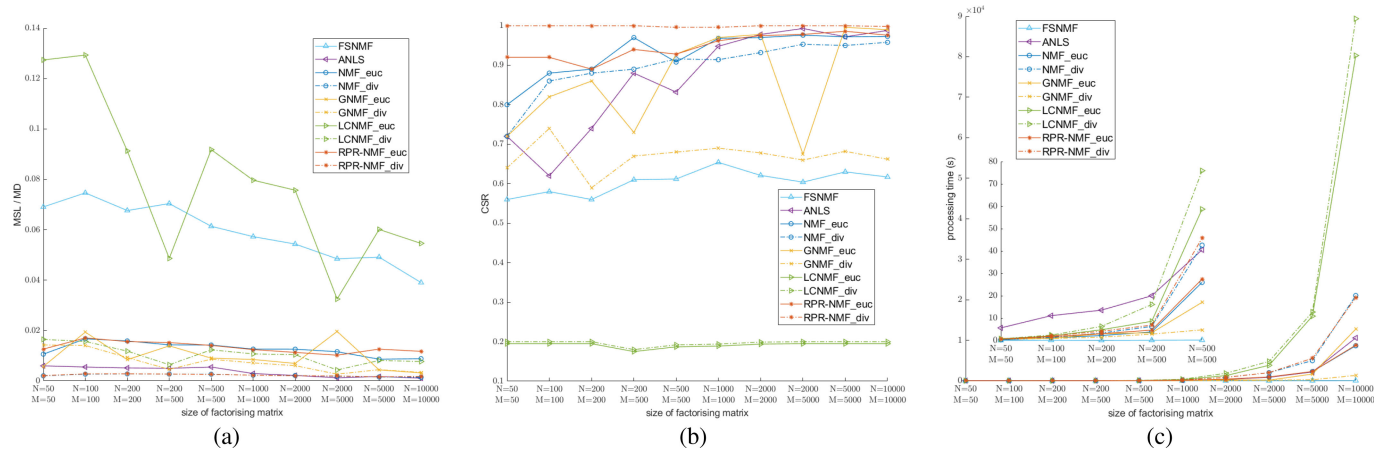


Fig. 5. Performance with respect to the size of factorising matrix. (a) MSL/MD, (b) CSR, and (c) Processing time.

set (b,c) the same label, which will contradict with the first constraint). Thus, GNMF and LCNMF cannot guarantee most of the constraints are satisfied after factorisation (recall the example in Fig. 2).

The above two synthetic experiments both demonstrate that RPR-NMF algorithms have the advantages of getting accurate factorisation and satisfying relative constraints comparing with other algorithms.

4.3 Parameter Selection

The parameters introduced by RPR-NMF are the penalty coefficients. We conducted experiments with varying values of these parameters under the following settings: $N = 100, M = 100, K = 20, l_W = l_H = N/10$. λ_W and λ_H vary from 0.4 to 4 with step 0.4 and from 20 to 100 with step 20. The average results of 10 repetitions are presented in Fig. 6.

As showed in the figure, the coefficients have nearly no influence on the CSR for both algorithms. However, the MSL of RPR-NMF using Euclidean measure increases when its parameters become larger, while the MD of the Divergence version is stable all the time. This suggests that RPR-NMF using Divergence measure is not sensitive to its parameters which can be set without much effort, but using smaller parameters is better when working with Euclidean measure.

4.4 Performance Analysis in Image Clustering

We validate the performance of algorithms in image clustering on two public datasets with ground truth: AT&T ORL and CMU PIE.

4.4.1 AT&T ORL Dataset

The AT&T ORL database consists of 400 images for 40 classes with 10 different facial images in each class [32]. The images were taken at different times, lighting and facial expressions. The faces are in an upright position in frontal view, with a slight left-right rotation. Each image is pre-processed into a 32×32 matrix with 256 grey levels [23]. Thus the size of factorising matrix is $1,024 \times 400$.

We adopt the following steps in this experiment:

- i. Randomly choose K classes and mix up images from these classes to form the factorising matrix;

- ii. In the K classes, randomly select 2 images in each class and set them more similar to images chosen in other classes, which forms the list of RPR constraints for RPR-NMF; transform the constraints into a weight matrix and a label matrix; the latent dimension is set as the number of chosen classes K ;
- iii. Run algorithms to obtain the right factorised matrix H ; utilise K-means method on H to get the clustering results;
- iv. Calculate the ACC and NMI for each algorithm.

The number of class K varies from 2 to 10 with step 2. The penalty coefficients for RPR-NMF using Euclidean measure are set 20 and for its Divergence version are set 2. The results are showed in Table 2.

From the table, RPR-NMF using Euclidean measure achieves the best average ACC (74.57 percent) and the best average NMI (81.91 percent) as well. Among algorithms using Euclidean measure, RPR-NMF outperforms FSNMF, ANLS, NMF, GNMF and LCNMF by 18.24, 10.32, 8.58, 12.77, 2.94 percent on ACC and by 13.54, 6.76, 5.77, 11.75, 0.92 percent on NMI respectively; as for algorithms using Divergence measure, RPR-NMF outperforms NMF and GNMF by

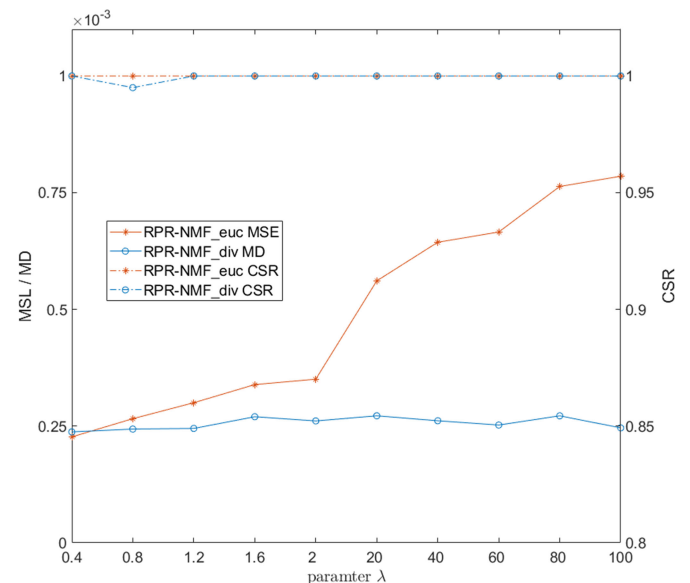


Fig. 6. MSL/MD & CSR of RPR-NMF algorithms with penalty coefficients varying from 0.4 to 2 and from 20 to 100.

TABLE 2
Clustering Performance on AT&T ORL Database (Euclidean/Divergence)

	K	l_H	FSNMF	ANLS	NMF	GNMF	LCNMF	RPR-NMF
MSL/MD	5	10	561.3	335.2	339.9/ 1.604	397.9/1.980	355.1/1.677	340.7/1.606
	10	20	432.3	252.2	263.2/1.200	304.5/1.589	288.8/1.294	261.7/ 1.197
	20	40	422.0	232.8	253.7/1.155	267.2/1.610	284.1/1.293	252.6/ 1.147
	30	60	400.2	202.8	228.6/ 1.039	233.4/1.494	265.5/1.207	228.1/1.041
	40	80	375.9	180.5	211.0/0.960	209.2/1.400	247.1/1.116	210.9/ 0.957
	Avg.		438.3	240.7	259.3/1.192	282.4/1.615	288.1/1.317	258.8/ 1.190
CSR (%)	5	10	94.00	92.00	90.00/86.00	98.00/ 100.0	100.0/100.0	98.00/96.00
	10	20	81.00	89.00	88.00/87.00	98.00/ 100.0	100.0/100.0	94.00/ 100.00
	20	40	84.50	90.50	92.00/83.50	91.00/ 100.0	100.0/100.0	91.50/ 100.00
	30	60	86.00	89.33	91.00/89.00	92.33/ 100.0	100.0/100.0	97.00/ 100.00
	40	80	85.25	90.50	91.25/89.25	93.50/ 100.0	100.0/100.0	97.00/ 100.00
	Avg.		86.15	90.27	90.45/86.95	94.57/ 100.0	100.0/100.0	95.50/99.20
ACC (%)	5	10	74.00	82.80	78.40/76.40	67.20/65.60	89.20/85.20	90.40/88.40
	10	20	58.60	65.40	64.20/68.00	63.40/55.60	75.00/71.20	78.80/71.80
	20	40	52.50	63.40	64.10/63.50	62.10/53.30	64.80/ 70.40	72.20/68.00
	30	60	50.53	56.00	62.00/60.93	59.33/48.47	66.20/63.73	67.93/64.67
	40	80	46.00	53.65	61.25/58.25	56.95/46.85	62.95/ 59.20	63.50/58.10
	Avg.		56.33	64.25	65.99/65.42	61.80/53.96	71.63/69.95	74.57/70.19
NMI (%)	5	10	70.12	80.87	73.37/79.29	59.01/51.76	85.33/ 83.03	85.35/82.05
	10	20	67.55	72.66	73.28/74.10	67.69/58.00	81.24/77.89	82.12/78.40
	20	40	67.83	75.88	77.11/76.33	73.57/60.59	79.06/ 81.79	82.44/78.19
	30	60	68.28	73.47	77.92/76.80	74.92/59.82	80.12/78.85	80.34/78.89
	40	80	68.07	72.89	79.01/77.65	75.62/60.39	79.22/ 78.01	79.29/75.71
	Avg.		68.37	75.15	76.14/76.83	70.16/58.11	80.99/ 79.91	81.91/78.65

4.77, 16.23 percent on ACC and by 1.82, 20.54 percent on NMI respectively, and is slightly better than LCNMF on ACC but has a bit lower NMI score than LCNMF. ANLS obtains the lowest factorisation errors, followed by RPR-NMF algorithms. All of the algorithms obtain very high CSR, because the images in this dataset are quite distinguishing. Besides, there is no chain among RPR constraints, thus LCNMF can satisfy all of them by setting proper labels.

4.4.2 CMU PIE Dataset

CMU PIE database was collected at Carnegie Mellon University in 2000, and it has been very influential in advancing research in face recognition across pose and illumination [33]. We followed the pre-processed PIE dataset used in [22] which contains 2,856 images for 68 different people with 42 images for each person. The images are processed into 32×32 matrices denoting the grey level of pixels. Thus the size of factorising matrix is $1,024 \times 2,856$.

We adopt similar experimental steps as we did for AT&T ORL dataset with a few changes on the extraction of RPR constraints. For this experiment, we randomly select 4 images instead of 2 in each cluster. Moreover, considering that the similarity can exist not only among intra-class images, but also inter-class images (e.g., the image of a dog is more similar to that of another dog, rather than a cat), we extract constraints in both ways. The number of class K varies from 10 to 60 with step 10. The penalty coefficients for RPR-NMF using Euclidean measure are set 20. For its Divergence version, they are set 2. The results are presented in Table 3.

According to the table, we can see that ANLS and NMF using Divergence measure achieve the best average

approximation (120.0 MSL and 1.242 MD). As for the other three evaluation metrics, RPR-NMF using Euclidean measure achieves the best average CSR (87.03 percent) and NMI (73.76 percent) while RPR-NMF using Divergence measure achieves the best average ACC (66.02 percent). Notice that the CSR of LCNMF are all zeros because inter-class and intra-class constraints lead to cyclic chain constraints.

Among the algorithms using Euclidean measure, RPR-NMF outperforms FSNMF, ANLS, NMF, GNMF and LCNMF by 38.16, 8.57, 3.86, 6.88, 8.26 percent on ACC and by 35.29, 4.80, 1.86, 3.71, 10.14 percent on NMI respectively. For the algorithms using Divergence measure, RPR-NMF outperforms NMF, GNMF and LCNMF by 4.66, 13.145, 10.42 percent on ACC. However, its performance on NMI is a bit lower than NMF, while it outperforms GNMF and LCNMF by 10.14, 9.02 percent.

4.5 Performance Analysis in Recommender Systems

As for the performance analysis in recommender systems, we compare our algorithm RPR-NMF with NMF, GNMF, and LCNMF on Movielens 1M dataset. For the reason that the rating matrices in recommender systems have missing values which are usually denoted by zeros, all the algorithms have to be modified with a MASK matrix for incomplete factorising matrix as proposed in [34]. To our best effort, we implemented the modified version for all algorithms except for the GNMF using Divergence measure. In the Appendix of [22], the authors only mentioned how to deal with incomplete factorising matrix for

TABLE 3
Clustering Performance on CMU PIE Database (Euclidean/Divergence)

	K	l_H	FSNMF	ANLS	NMF	GNMF	LCNMF	RPR-NMF
MSL/MD	10	120	548.4	191.2	203.1/ 1.788	203.8/2.916	429.8/3.234	210.1/1.852
	20	240	589.6	147.4	171.6/ 1.385	163.2/2.689	425.3/3.063	161.4/1.478
	30	360	510.6	124.1	151.4/ 1.292	140.6/2.447	397.4/2.944	141.5/1.370
	40	480	578.8	106.5	137.3/ 1.147	122.9/2.282	382.4/2.793	125.2/1.219
	50	600	592.2	98.11	130.0/ 1.088	113.7/2.217	385.8/2.796	118.1/1.139
	60	720	610.6	89.88	122.3/ 1.013	105.5/2.124	378.2/2.731	110.4/1.102
	68	816	599.1	82.74	116.9/ 0.978	98.83/2.037	370.8/2.693	103.5/0.998
	Avg.		575.6	120.0	147.5/ 1.242	135.5/2.387	395.7/2.893	138.6/1.308
CSR (%)	10	120	72.83	78.17	84.00/78.67	80.50/68.83	00.00/00.00	84.17/82.83
	20	240	72.92	78.83	82.83/ 73.25	82.25/70.58	00.00/00.00	85.42/71.75
	30	360	73.06	78.72	82.17/72.89	83.22/68.28	00.00/00.00	86.78/76.83
	40	480	74.96	79.71	83.13/76.04	84.46/70.58	00.00/00.00	88.08/77.29
	50	600	74.40	78.47	82.50/ 76.83	84.27/67.53	00.00/00.00	88.43/75.60
	60	720	72.78	79.14	81.47/74.92	83.69/67.67	00.00/00.00	87.83/75.50
	68	816	74.26	78.92	81.30/ 74.34	84.14/68.70	00.00/00.00	88.51/71.57
	Avg.		73.60	78.85	82.49/75.28	83.22/68.88	00.00/00.00	87.03/75.91
ACC (%)	10	120	32.48	55.86	64.86/63.19	51.76/53.71	60.29/57.52	66.43/66.76
	20	240	27.21	56.83	58.00/62.57	60.79/51.10	56.88/57.05	67.36/66.64
	30	360	27.06	59.41	62.49/61.89	58.21/53.51	57.87/57.16	66.32/66.73
	40	480	26.13	59.12	61.73/59.35	60.23/54.20	56.25/54.13	64.51/66.17
	50	600	25.53	53.22	60.63/61.62	58.98/52.15	56.77/56.17	64.49/66.24
	60	720	24.44	56.37	61.36/60.09	57.06/53.97	55.23/53.38	62.31/66.48
	68	816	23.82	53.03	57.75/60.78	58.60/51.51	52.72/53.79	62.40/63.10
	Avg.		26.67	56.26	60.97/61.36	57.95/52.88	56.57/55.60	64.83/66.02
NMI (%)	10	120	28.57	56.02	67.13/67.10	57.73/52.70	55.78/55.98	67.01/64.38
	20	240	34.24	67.04	68.54/ 72.07	68.63/59.70	60.88/61.87	73.61/71.00
	30	360	38.93	71.48	71.87/ 72.60	70.37/63.14	64.71/63.66	75.06/72.55
	40	480	39.19	72.44	73.15/73.01	72.66/63.26	64.40/65.35	73.94/74.63
	50	600	41.80	70.52	73.85/74.40	73.41/64.53	66.48/65.06	75.67/74.96
	60	720	43.07	73.19	75.47/74.49	73.35/66.32	66.68/66.12	75.33/75.05
	68	816	43.48	72.01	73.32/ 75.25	74.18/66.31	66.38/65.65	75.72/74.21
	Avg.		38.47	68.96	71.90/ 72.70	70.05/62.28	63.62/63.38	73.76/72.40

GNMF using Euclidean measure. Thus we did not compare GNMF using Divergence measure in this part.

Specifically, the pairwise relationship constraints in this part are extracted from meta information and are imposed on both factorised matrices: we utilised users' gender, age and occupation as well as movies' genre to obtain RPRs for the Movielens dataset.

4.5.1 Movielens 1M Dataset

The Movielens 1M dataset is a well-known stable baseline dataset in recommender systems [35]. It has 1,000,209 ratings (1 to 5) from 6,040 users on 3,883 movies. Indeed some movies have no ratings, thus after removing these movies, we derived a pre-processed $6,040 \times 3,706$ rating matrix.

Three groups of cross validation experiments are conducted on this dataset: (1) 300 constraints, $K = 20$; (2) 500 constraints and $K = 50$; (3) 1000 constraints and $K = 100$. The penalty coefficients for RPR-NMF using Euclidean measure are set 200, 10 and 1 respectively, and the coefficients for the Divergence version of RPR-NMF are set 0.1, 0.01, 0.001 respectively. For each group, we conducted 5 cross validation experiments and the average results are showed in Table 4.

As presented in the table, NMF achieves the lowest MSL and RPR-NMF achieves the lowest MD; both of the algorithms obtain close approximation errors while the other two methods, GNMF and LCNMF, have a higher rate of errors. RPR-NMF also achieves the highest CSR on both its versions. As for the recommendation evaluation, GNMF using Euclidean measure achieves the lowest RMSE while RPR-NMF using Euclidean measure achieves the highest F1 score. It is worth noting that, in this experiment, the RPRs are randomly selected through meta information, and some of the constraints may not represent the true ratings' pattern. Thus the RPR-NMF algorithms seem not greatly outperform existing methods. However, their overall performance is still better compared to other alternatives.

5 CONCLUSIONS

In this paper, we proposed a novel matrix factorisation algorithm called RPR-NMF, to effectively utilise the relative pairwise relationship among rows or columns of factorised matrices. Both of the Euclidean and Divergence measures are used in the objective function. RPR-NMF imposes penalties for each relative pairwise relationship constraint in a form of addition of natural exponential functions for

TABLE 4
Cross Validation Results on Movielens 1M Dataset (Euclidean/Divergence)

K	l_W & l_H	MSL/MD				CSR (%)			
		NMF	GNMF_euc	LCNMF	RPR-NMF	NMF	GNMF_euc	LCNMF	RPR-NMF
20	300	0.513/0.083	0.526	0.532/0.086	0.514/ 0.082	49.07/49.87	87.37	50.00/50.00	96.17/95.77
50	600	0.373/0.060	0.409	0.422/0.068	0.374/ 0.060	49.23/51.03	89.25	50.00/50.00	95.13/98.83
100	900	0.249/0.039	0.317	0.334/0.053	0.253/ 0.039	50.17/51.51	89.72	49.94/49.94	93.49/99.39
Avg.		0.378/0.061	0.417	0.429/0.069	0.380/ 0.060	49.38/50.80	88.78	49.98/49.98	94.93/98.00
K	l_W & l_H	RMSE				F1 Score (%)			
		NMF	GNMF_euc	LCNMF	RPR-NMF	NMF	GNMF_euc	LCNMF	RPR-NMF
20	300	0.959/0.975	0.929	0.991/1.004	0.928/0.974	68.75/68.67	68.29	66.39/66.23	69.25/68.71
50	600	1.027/1.045	0.961	1.044/1.068	0.979/ 1.030	67.34/ 67.20	67.45	64.78/64.80	67.51/67.19
100	900	1.100/ 1.147	1.010	1.133/1.165	1.055/1.157	65.10/65.01	65.31	62.20/62.26	65.46/65.03
Avg.		1.028/ 1.055	0.966	1.056/1.079	0.987/1.056	67.06/66.96	67.02	64.46/64.43	67.41/66.98

Euclidean measure and hinge loss for Divergence measure. Complete and sufficient proofs of convergence are also provided to ensure that RPR-NMF conforms to the “multiplicative update rules”. Numerical analysis shows that the proposed algorithm achieves superior performance on both the overall loss and the accuracy of satisfied constraints compared with the other algorithms. Experiments on synthetic and real datasets for image clustering and recommender systems demonstrate the effectiveness of RPR-NMF algorithms which outperform baseline methods on several evaluation criteria.

The experimental results presented in this work show that the selection of RPRs also plays an important role in an overall recommender system. Although outside of the scope of this paper, in our future work, we will also explore approaches to select application-oriented RPR constraints.

ACKNOWLEDGMENTS

This work was partly supported by Beijing Natural Science Foundation (4172054).

REFERENCES

- [1] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [2] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, “Metagenes and molecular pattern discovery using matrix factorization,” *Proc. Nat. Academy Sci. United States America*, vol. 101, no. 12, pp. 4164–4169, 2004.
- [3] C. Ding, T. Li, and M. I. Jordan, “Convex and semi-nonnegative matrix factorizations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.
- [4] J. Yang and J. Leskovec, “Overlapping community detection at scale: A nonnegative matrix factorization approach,” in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 587–596.
- [5] N. Mohammadiha, P. Smaragdakis, and A. Leijon, “Supervised and unsupervised speech enhancement using nonnegative matrix factorization,” *IEEE Trans. Audio Speech Language Process.*, vol. 21, no. 10, pp. 2140–2151, Oct. 2013.
- [6] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
- [7] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, “A convex model for nonnegative matrix factorization and dimensionality reduction on physical space,” *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3239–3252, Jul. 2012.
- [8] J. Liu, C. Wang, J. Gao, and J. Han, “Multi-view clustering via joint nonnegative matrix factorization,” in *Proc. SIAM Int. Conf. Data Mining*, May 2013, pp. 252–260.
- [9] H. Kim, J. Choo, J. Kim, C. K. Reddy, and H. Park, “Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization,” in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 567–576.
- [10] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 556–562.
- [11] E. F. Gonzalez and Y. Zhang, “Accelerating the Lee-Seung algorithm for non-negative matrix factorization,” Dept. Comput. Appl. Math., Rice Univ., Houston, TX, USA, Tech. Rep. TR-05-02, 2005.
- [12] C.-J. Lin, “Projected gradient methods for nonnegative matrix factorization,” *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, 2007.
- [13] C. Ding, X. He, and H. D. Simon, “On the equivalence of nonnegative matrix factorization and spectral clustering,” in *Proc. SIAM Int. Conf. Data Mining*, 2005, pp. 606–610.
- [14] N. Gillis and S. A. Vavasis, “Fast and robust recursive algorithms for separable nonnegative matrix factorization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 698–714, Apr. 2014.
- [15] K. Huang, N. D. Sidiropoulos, and A. Swami, “Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition,” *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 211–224, Jan. 2014.
- [16] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Nov. 2004.
- [17] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, “Nonsmooth nonnegative matrix factorization (nsNMF),” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 403–415, Mar. 2006.
- [18] K. Kimura, M. Kudo, and Y. Tanaka, “A column-wise update algorithm for nonnegative matrix factorization in Bregman divergence with an orthogonal constraint,” *Mach. Learn.*, vol. 103, no. 2, pp. 285–306, 2016.
- [19] A. Ozerov and C. Fevotte, “Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation,” *IEEE Trans. Audio Speech Language Process.*, vol. 18, no. 3, pp. 550–563, Mar. 2010.
- [20] R. Sandler and M. Lindenbaum, “Nonnegative matrix factorization with earth mover’s distance metric for image analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1590–1602, Aug. 2011.
- [21] V. P. Pauca, J. Piper, and R. J. Plemmons, “Nonnegative matrix factorization for spectral data analysis,” *Linear Algebra Appl.*, vol. 416, no. 1, pp. 29–47, 2006.
- [22] D. Cai, X. He, J. Han, and T. S. Huang, “Graph regularized non-negative matrix factorization for data representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.
- [23] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, “Constrained non-negative matrix factorization for image representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1299–1311, Jul. 2012.
- [24] S. Jia and Y. Qian, “Constrained nonnegative matrix factorization for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 161–173, Jan. 2009.

- [25] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. I-207–I-212.
- [26] B. Shen and L. Si, "Non-negative matrix factorization clustering on multiple manifolds," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 575–580.
- [27] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. Appl.*, vol. 30, pp. 713–730, 2008.
- [28] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.
- [29] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 267–273.
- [30] A. G. Barnston, "Correspondence among the correlation, RMSE, and heidke forecast verification measures; Refinement of the heidke score," *Weather Forecasting*, vol. 7, no. 4, pp. 699–709, 1992.
- [31] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *Bioinfo Publications*, 2011.
- [32] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. 2nd IEEE Workshop Appl. Comput. Vis.*, 1994, pp. 138–142.
- [33] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, 2010.
- [34] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2006, pp. 549–553.
- [35] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interactive Intell. Syst.*, vol. 5, no. 4, pp. 19:1–19:19, Dec. 2015.



Shuai Jiang received the bachelor's degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2013. Currently, he is working towards the dual doctoral degree at both the Beijing Institute of Technology and University of Technology Sydney. His main interests include machine learning, optimisation, and data analytics.



Kan Li is currently a professor with the School of Computer, Beijing Institute of Technology. He has published more than 50 technical papers in peer-reviewed journals and conference proceedings. His research interests include machine learning and pattern recognition.



Richard Yi Da Xu received the BEng degree in computer engineering from the University of New South Wales, Sydney, NSW, Australia, in 2001, and the PhD degree in computer sciences from the University of Technology at Sydney (UTS), Sydney, NSW, Australia, in 2006. He is currently an associate professor with the School of Electrical and Data Engineering, UTS. His current research interests include machine learning, deep learning, data analytics, and computer vision.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.