Robust Nonnegative Matrix Factorization with Ordered Structure Constraints

Jing Wang[1], Feng Tian[1], Chang Hong Liu[2], Hongchuan Yu[3], Xiao Wang[4,*], Xianchao Tang[5]
[1]Faculty of Science and Technology, Bournemouth University, UK
[2]Department of Psychology, Bournemouth University, UK
[3]National Centre for Computer Animation, Bournemouth University, UK
[4]Department of Computer Science and Technology, Tsinghua University, China
[5]School of Computer Science and Technology, Tianjin University, China
[*]Email:wangxiao007@mail.tsinghua.edu.cn

***Abstract*—Nonnegative matrix factorization (NMF) as a popular technique to find parts-based representations of nonnegative data has been widely used in real-world applications. Often the data which these applications process, such as motion sequences and video clips, are with ordered structure, i.e., consecutive neighbouring data samples are very likely share similar features unless a sudden change occurs. Therefore, traditional NMF assumes the data samples and features to be independently distributed, making it not proper for the analysis of such data. In this paper, we propose an ordered robust NMF (ORNMF) by capturing the embedded ordered structure to improve the accuracy of data representation. With a novel neighbour penalty term, ORNMF enforces the similarity of neighbouring data. ORNMF also adopts the $L_{2,1}$-norm based loss function to improve its robustness against noises and outliers. A new iterative updating optimization algorithm is derived to solve ORNMF's objective function. The proofs of the convergence and correctness of the scheme are also presented. Experiments on both synthetic and real-world datasets have demonstrated the effectiveness of ORNMF.**

I. Introduction

Finding an optimal data representation is a fundamental problem in many data analysis tasks [1], [2]. A good data representation can typically reveal the latent structure of data and facilitate further processes such as clustering, classification and recognition. Nonnegative matrix factorization (NMF) [3] as a fundamental approach for such data representation has attracted great attentions.

NMF based approaches have been widely used in the fields of machine learning and computer vision such as motion segmentation [4], [5], human activity recognition [6] and face recognition [7]. In these applications, data such as a video clip, a sequence of a subjects images taken under changing illuminations are often with ordered structure, i.e., consecutive neighbouring data samples are very likely share similar features unless a sudden change occurs. This ordered structure provides valuable information about the relationship between data [8], [9], [10], [11]. However, to our best knowledge, this ordered structure has not been given enough or specific attentions by existing NMF based approaches. As a result, it is unlikely or extremely challenging to find optimal representations of sequential data. For example, to cluster frames of a video clip into scenes they belong to, the representations of the frames in the same scene could be quite different, due to the fact that only the characteristic features of frames such as illumination or perspective are utilized by the existing approaches. Instead, if the ordered structure is incorporated as a constraint, these differences will be reduced because the representations of every two neighbouring frames can be enforced to be similar. This will improve the clustering accuracy. Thus, exploiting the ordered structure with NMF holds a great potential for seeking for optimal representations.

Also, real-world data usually come with noise and outliers. The standard NMF uses the least square error function which is unstable with respect to noise and outliers [12], because a few noisy features or with large errors will dominate objective function. Thus, more practical NMF approaches are required to tackle the issue of noises or outliers [13], [14].

In this paper, we take factors above into consideration and propose a novel method, named as ordered robust nonnegative matrix factorization (ORNMF). A novel neighbour penalty term is constructed to enforce the similarity of the consecutive data representations to preserve the ordered structure of data. A $L_{2,1}$-norm loss function is used to improve the robustness so that ORNMF is insensitive to the data outliers and applicable to applications with noisy data. An efficient and elegant iterative updating rule is derived and analyzed theoretically to demonstrate its correctness and convergence. The experiments on one synthetic and three real datasets, in comparison with both baselines and state-of-the-art methods, have demonstrated the superiority of ORNMF in terms of accuracy and normalized mutual information.

II. Related Work

In different circumstances, various variants of NMF have been proposed to seek for effective representations of data. In particular, to deal with the issues caused by outliers and noises, Kong *et al*. [13] proposed a robust formulation of NMF (RNMF), where the errors are measured by $L_{2,1}$-norm rather than the conventional least square function. Based on the assumption that the data points nearby have more similar data representations than those far away, a graph regularized NMF (GNMF) [15] was proposed to model the local manifold structure. These NMF methods, which are referred as unsupervised learning methods, are not optimal to many real-world problems where limited knowledge (such as label information) from domain experts is available. To address this limitation, Liu *et al*. [2] extended NMF to the semi-supervised setting and proposed the constrained NMF (CNMF). It takes the

label information as hard constraints by enforcing data with the same label to have the same new representations, thus, the obtained representations may have more discriminating power. By far, all the methods mentioned above are developed for dealing with data of single view (feature) only. In order to integrate multiple features for more comprehensive understanding of data, a multi-view NMF (MultiNMF) [16] was proposed. It aims to obtain a common consensus data representation matrix with combination of multiple features together. However, all these NMF methods deal with features only, and are not able to utilize the ordered structure of the data as conditional constraints to improve the discriminative ability of data representation.

Recently, several approaches [17], [18], [19] have been proposed for temporally changing data streams. However, these methods are mainly developed in the online learning setting, i.e., how to effectively learn the representation of new data rather than preserving the ordered structure of all the data. Slow features NMF (SFNMF) [20] focuses on capturing the transitions between the temporal phases of facial action units by considering the principles of temporal slowness in NMF. However, it does not fully consider enhancing the similarity of neighbouring data. And moreover, SFNMF uses least square loss function which is not resilience to large noises and outliers. NMF with interpolated coefficients (NMFi) [21] considers the relationship between neighbouring data to smooth the representations of consecutive data samples. It modifies the algorithm of NMF [3] by simply setting every other representations of data samples be the average of the previous and following representations. However, this algorithm is not normally derived directly from NMF, which will lead to a suboptimal solution. More importantly, the convergence and correctness of the algorithm cannot be guaranteed.

III. A Brief Review of NMF

Given a nonnegative data matrix $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_n] \in \mathbb{R}^{m \times n}$, of which each column represents a data point. NMF [3] aims to decompose $\mathbf{V}$ into a nonnegative basis matrix $\mathbf{W} \in \mathbb{R}^{m \times k}$ and a representation matrix $\mathbf{H} \in \mathbb{R}^{k \times n}$, where k denotes the number of bases. Mathematically, NMF solves the following optimization problem:

$$\min_{\mathbf{W} \geq 0, \mathbf{H} \geq 0} \|\mathbf{V} - \mathbf{W}\mathbf{H}\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ is Frobenius norm and defined as $\|\mathbf{V} - \mathbf{W}\mathbf{H}\|_F^2 = \sum_{i=1}^{n} \|\mathbf{V}_i - \mathbf{W}\mathbf{H}_i\|^2$. The optimization problem is usually solved by an multiplicative updating rule proposed in [3] due to its computational efficiency compared to traditional gradient descent algorithms. With the updating rule, the objective function (1) is solved iteratively and the solution converges to a local minimum.

IV. Ordered Robust Nonnegative Matrix Factorization (ORNMF)

ORNMF is proposed in this study to enforce the similarity between representations of neighbouring data. The inspiration behind ORNMF is that the changes between neighbouring data are usually very subtle, so the representations of these data should be similar to each other. Taken a video sequence for an example, since the scenes in the sequence normally change much less frequently than the frame rate, it is safe to assume that a high similarity exists among consecutive frames, except when two neighbouring frames are from different scenes.

To achieve the optimal data representations by incorporating this ordered structure, a novel regularization term is incorporated to (1) in two steps. First, we construct the following matrix $\mathbf{R} \in \mathbb{R}^{n \times (n-1)}$, which is a lower triangular matrix with -1 on the diagonal and 1 on the second diagonal:

$$\mathbf{R} = \begin{bmatrix} -1 & 0 & 0 & \dots & 0 \\ 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & -1 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Multiplying $\mathbf{H}$ by $\mathbf{R}$ gives $\mathbf{HR} = [\mathbf{H}_2 - \mathbf{H}_1, \mathbf{H}_3 - \mathbf{H}_2, \mathbf{H}_4 - \mathbf{H}_3 \dots \mathbf{H}_n - \mathbf{H}_{n-1}]$. If the columns of $\mathbf{HR}$ are or nearly equal to zero vectors, i.e. $\mathbf{H}_i - \mathbf{H}_{i-1} \approx 0$, data must be from the same subject/scene because they are similar, or a boundary or sudden change exists inbetween. Given k subjects, ideally, only $k-1$ non-zero columns should $\mathbf{HR}$ have. To guarantee $k-1$ non-zeros columns, we introduce a $L_{2,0}$-norm, $\|\cdot\|_{2,0}$, to penalise each column directly and maintain the sparsity of $\mathbf{HR}$. The quasi-norm $L_{2,0}$-norm is defined as the number of non-zero columns. We thereby propose an objective function as

$$\min_{\mathbf{W} \geq 0, \mathbf{H} \geq 0} J = \|\mathbf{V} - \mathbf{W}\mathbf{H}\|_F^2 + \alpha \|\mathbf{HR}\|_{2,0}, \tag{2}$$

where α is a trade-off parameter that controls the weight of the regularization term.

However, solving the problem (2) is NP-hard because of the $L_{2,0}$-norm [22]. According to [22], the $L_{2,1}$-norm of a given matrix $\mathbf{X}$, i.e., $\|\mathbf{X}\|_{2,1}$, is the minimum convex hull of $\|\mathbf{X}\|_{2,0}$. When $\mathbf{X}$ is column-sparse enough, namely, many zero columns are involved, minimize $\|\mathbf{X}\|_{2,1}$ is always equivalent to minimize $\|\mathbf{X}\|_{2,0}$. Therefore, we can relax the objective function (2) as:

$$\min_{\mathbf{W} \geq 0, \mathbf{H} \geq 0} J = \|\mathbf{V} - \mathbf{W}\mathbf{H}\|_F^2 + \alpha \|\mathbf{HR}\|_{2,1}. \tag{3}$$

Since the error, i.e. the first term of (3) is squared, a few big ones due to outliers or noises may dominate the objective function. As in [13], we then propose a more robust function as the following:

$$\min_{\mathbf{W} \geq 0, \mathbf{H} \geq 0} J = \|\mathbf{V} - \mathbf{W}\mathbf{H}\|_{2,1} + \alpha \|\mathbf{HR}\|_{2,1}, \tag{4}$$

where the $L_{2,1}$-norm is applied to the loss function and defined as $\|\mathbf{V} - \mathbf{W}\mathbf{H}\|_{2,1} = \sum_{i=1}^{n} \|\mathbf{V}_i - \mathbf{W}\mathbf{H}_i\|$. With the error for each data not being squared, the impact of large errors is reduced significantly.

A. Optimization Algorithm

Since the optimization problem in (4) is not convex in both variables $\mathbf{W}$ and $\mathbf{H}$, it is infeasible to find the global minimum. In addition, as the matrix $\mathbf{R}$ contains negative values, it is technically challenging to solve (4) directly. Here we propose an algorithm that iteratively updates $\mathbf{H}$ with $\mathbf{W}$ fixed and then $\mathbf{W}$ with $\mathbf{H}$ fixed, which guarantees the objective function values do not increase with iterations.

Update for H: To update $\mathbf{H}$ with $\mathbf{W}$ fixed, we need to solve the following problem:

$$\min_{\mathbf{H}\geq 0} J(\mathbf{H}) = \|\mathbf{V} - \mathbf{W}\mathbf{H}\|_{2,1} + \alpha\|\mathbf{H}\mathbf{R}\|_{2,1}. \tag{5}$$

We introduce a Lagrange multiplier matrix $\boldsymbol{\eta} = [\eta_{ij}] \in \mathbb{R}^{k\times n}$ for the constraint $\mathbf{H} \geq 0$, then we have the following equivalent objective function:

$$J(\mathbf{H}) = tr(\mathbf{V}\mathbf{D}_1\mathbf{V}^T - 2\mathbf{V}\mathbf{D}_1\mathbf{H}^T\mathbf{W}^T + \mathbf{W}\mathbf{H}\mathbf{D}_1\mathbf{H}^T\mathbf{W}^T) + \alpha tr(\mathbf{H}\mathbf{R}\mathbf{D}_2\mathbf{R}^T\mathbf{H}^T). \tag{6}$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are diagonal matrices with the diagonal elements being

$$(\mathbf{D_1})_{ii} = \frac{1}{\|\mathbf{V}_i - \mathbf{W}\mathbf{H}_i\|}, i = 1, 2..., n. \tag{7}$$

$$(\mathbf{D_2})_{ii} = \frac{1}{\|(\mathbf{H}\mathbf{R})_i\|}, i = 1, 2..., n-1. \tag{8}$$

Setting the derivative of $J(\mathbf{H})$ to be 0 with respect to $\mathbf{H}$, we have

$$\boldsymbol{\eta} = 2\mathbf{W}^T\mathbf{V}\mathbf{D}_1 - 2\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 - 2\alpha\mathbf{H}\mathbf{R}\mathbf{D}_2\mathbf{R}^T, \tag{9}$$

Following the Karush-Kuhn-Tucker (KKT) condition [23] $\eta_{ij}\mathbf{H}_{ij} = 0$, we have

$$(\mathbf{W}^T\mathbf{V}\mathbf{D}_1 - \mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 - \alpha\mathbf{H}\mathbf{R}\mathbf{D}_2\mathbf{R}^T)_{ij}\mathbf{H}_{ij} = 0. \tag{10}$$

Because $\mathbf{R}$ contains negative values, we decompose $\mathbf{R}$ into two nonnegative parts for ensuring $\mathbf{H} \geq 0$ in each iteration:

$$\mathbf{R} = \mathbf{R}^+ - \mathbf{R}^-, \tag{11}$$

where $\mathbf{R}^+_{ij} = (|\mathbf{R}_{ij}| + \mathbf{R}_{ij})/2$ and $\mathbf{R}^-_{ij} = (|\mathbf{R}_{ij}| - \mathbf{R}_{ij})/2$. Substituting (11) into (10), we obtain

$$(\mathbf{W}^T\mathbf{V}\mathbf{D}_1 - \mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}^+\mathbf{D}_2\mathbf{R}^{-T} + \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{+T}) - \alpha\mathbf{H}(\mathbf{R}^+\mathbf{D}_2\mathbf{R}^{+T} + \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{-T}))_{ij}\mathbf{H}_{ij} = 0. \tag{12}$$

Denoting $\mathbf{R}_a = \mathbf{R}^+\mathbf{D}_2(\mathbf{R}^-)^T$, $\mathbf{R}_b = \mathbf{R}^-\mathbf{D}_2(\mathbf{R}^+)^T$, $\mathbf{R}_c = \mathbf{R}^+\mathbf{D}_2\mathbf{R}^{+T}$, $\mathbf{R}_d = \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{-T}$, we then have the following successive update of $\mathbf{H}$ with an initial value of $\mathbf{H}$.

$$\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij}\sqrt{\frac{(\mathbf{W}^T\mathbf{V}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}_a + \mathbf{R}_b))_{ij}}{(\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}_c + \mathbf{R}_d))_{ij}}}. \tag{13}$$

When (13) converges, its solution satisfies (12).

This updating rule of $\mathbf{H}$ satisfies the following theorem, which guarantees the correctness of the rule.

Theorem 1. If the updating rule of $\mathbf{H}$ converges, then the final solution satisfies the KKT optimality condition.

Proof of Theorem 1. At convergence, $\mathbf{H}^\infty = \mathbf{H}^{t+1} = \mathbf{H}^t = \mathbf{H}$, where t denotes the t-th iteration, i.e.,

$$\mathbf{H}_{ij} = \mathbf{H}_{ij}\sqrt{\frac{(\mathbf{W}^T\mathbf{V}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}_a + \mathbf{R}_b))_{ij}}{(\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}_c + \mathbf{R}_d))_{ij}}}. \tag{14}$$

This is the same as

$$(\mathbf{W}^T\mathbf{V}\mathbf{D}_1 - \mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}^+\mathbf{D}_2\mathbf{R}^{-T} + \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{+T}) - \alpha\mathbf{H}(\mathbf{R}^+\mathbf{D}_2\mathbf{R}^{+T} + \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{-T}))_{ij}\mathbf{H}^2_{ij} = 0. \tag{15}$$

which is equivalent to (12). □

We now prove the convergence of the updating rule. To achieve this goal, following [24], we use an auxiliary function as following .

Definition 1 [24] A function $G(\mathbf{H}, \mathbf{H}')$ is an auxiliary function of the function $J(\mathbf{H})$ if $G(\mathbf{H}, \mathbf{H}') \geq J(\mathbf{H})$ and $G(\mathbf{H}, \mathbf{H}) = J(\mathbf{H})$ for any $\mathbf{H}$ and a constant matrix $\mathbf{H}'$.

The auxiliary function helps because of the following lemma:

Lemma 1 [24] If G is an auxiliary function of J, then J is non-increasing under the updating rule $\mathbf{H}^{t+1} = \arg\min_{\mathbf{H}} G(\mathbf{H}, \mathbf{H}^t)$.

Proof. $J(\mathbf{H}^{t+1}) \leq G(\mathbf{H}^{t+1}, \mathbf{H}^t) \leq G(\mathbf{H}^t, \mathbf{H}^t) = J(\mathbf{H}^t)$

Now we have the specific form of the auxiliary function $G(\mathbf{H}, \mathbf{H}')$ for the objective function $J(\mathbf{H})$ in the problem (5), based on the following lemma.

Lemma 2 The function

$$\begin{aligned} G(\mathbf{H}, \mathbf{H}') = &-2\sum_{ij}(\mathbf{W}^T\mathbf{V}\mathbf{D}_1)_{ij}\mathbf{H}'_{ij}(1 + \log\frac{\mathbf{H}_{ij}}{\mathbf{H}'_{ij}}) \\ &+ \sum_{ij}\frac{(\mathbf{W}^T\mathbf{W}\mathbf{H}'\mathbf{D}_1)_{ij}\mathbf{H}^2_{ij}}{\mathbf{H}'_{ij}} \\ &- \sum_{ijk}((\mathbf{R}_a + \mathbf{R}_b)_{jk})\mathbf{H}'_{ij}\mathbf{H}'_{ik}(1 + \log\frac{\mathbf{H}_{ij}\mathbf{H}_{ik}}{\mathbf{H}'_{ij}\mathbf{H}'_{ik}}) \\ &+ \sum_{ij}\frac{(\mathbf{H}'(\mathbf{R}_c + \mathbf{R}_d))_{ij}\mathbf{H}^2_{ij}}{\mathbf{H}'_{ij}} \end{aligned} \tag{16}$$

is an auxiliary function for $J(\mathbf{H})$ in problem (5).

Proof of Lemma 2. We find upper bounds for each of the two positive terms by the following lemma,

Lemma 3 [25]. For any nonnegative matrices $\mathbf{S} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{g\times g}$, $\mathbf{F} \in \mathbb{R}^{n\times g}$ and $\mathbf{F}' \in \mathbb{R}^{n\times g}$, with $\mathbf{S}$ and $\mathbf{B}$ are symmetric, then the following inequality holds

$$tr(\mathbf{F}^T\mathbf{S}\mathbf{F}\mathbf{B}) \leq \sum_{i=1}^{n}\sum_{p=1}^{g}(\mathbf{S}\mathbf{F}'\mathbf{B})\frac{\mathbf{F}^2{}_{ip}}{\mathbf{F}'{}_{ip}}. \tag{17}$$

Then, we have following inequations:

$$tr(\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1\mathbf{H}^T) \leq \sum_{ij}\frac{(\mathbf{W}^T\mathbf{W}\mathbf{H}'\mathbf{D}_1)_{ij}\mathbf{H}^2_{ij}}{\mathbf{H}'_{ij}}, \tag{18}$$

$$tr(\mathbf{H}(\mathbf{R}_c + \mathbf{R}_d)\mathbf{H}^T) \leq \sum_{ij}\frac{(\mathbf{H}'(\mathbf{R}_c + \mathbf{R}_d))_{ij}\mathbf{H}^2_{ij}}{\mathbf{H}'_{ij}}. \tag{19}$$

To obtain lower bounds for the remaining terms, we use the inequality $z > 1 + \log z, \forall z > 0$ [25] and have

$$tr(\mathbf{W}^T\mathbf{V}\mathbf{D}_1\mathbf{H}^T) \geq \sum_{ij}(\mathbf{W}^T\mathbf{V}\mathbf{D}_1)_{ij}\mathbf{H}'_{ij}(1+\log\frac{\mathbf{H}_{ij}}{\mathbf{H}'_{ij}}), \tag{20}$$

$$tr(\mathbf{H}(\mathbf{R}_a+\mathbf{R}_b)\mathbf{H}^T) \geq \sum_{ijk}(\mathbf{R}_a+\mathbf{R}_b)_{jk}\mathbf{H}'_{ij}\mathbf{H}'_{ik}(1+\log\frac{\mathbf{H}_{ij}\mathbf{H}_{ik}}{\mathbf{H}'_{ij}\mathbf{H}'_{ik}}). \tag{21}$$

Collecting all bounds, we have the final auxiliary function in **Lemma 2**. □

Based on Lemmas 1 and 2, we can show the convergence of the updating rule (13).

Theorem 2. The problem (5) is non-increasing under the iterative updating rule (13).

Proof of Theorem 2. **Lemma 2** provides a specific form $G(\mathbf{H}, \mathbf{H}')$ of the auxiliary function for $J(\mathbf{H})$ in problem (5). We can have the solution for $\min_{\mathbf{H}} G(\mathbf{H}, \mathbf{H}')$ by the following KKT condition

$$\frac{\partial G(\mathbf{H},\mathbf{H}')}{\partial \mathbf{H}_{ij}} = -2(\mathbf{W}^T\mathbf{V}\mathbf{D}_1)_{ij}\frac{\mathbf{H}'_{ij}}{\mathbf{H}_{ij}} + 2\frac{(\mathbf{W}^T\mathbf{W}\mathbf{H}'\mathbf{D}_1)_{ij}\mathbf{H}_{ij}}{\mathbf{H}'_{ij}} - 2\frac{(\mathbf{H}'(\mathbf{R}_a+\mathbf{R}_b))_{ij}\mathbf{H}'_{ij}}{\mathbf{H}_{ij}} + 2\frac{(\mathbf{H}'(\mathbf{R}_c+\mathbf{R}_d))_{ij}\mathbf{H}_{ij}}{\mathbf{H}'_{ij}} = 0, \tag{22}$$

which gives rise to the updating rule in (13). Following **Lemma 1**, under this updating rule the objective function values of $J(\mathbf{H})$ in (5) will be non-increasing. □

Update for W: To update $\mathbf{W}$ with $\mathbf{H}$ fixed, we need to solve the following problem:

$$\min_{\mathbf{W}\geq 0} J(\mathbf{W}) = \|\mathbf{V} - \mathbf{W}\mathbf{H}\|_{2,1} \tag{23}$$

This is exactly same as that in [13]. So we have the following updating rule for (23).

$$\mathbf{W}_{di} \leftarrow \mathbf{W}_{di}\frac{(\mathbf{V}\mathbf{D}_1\mathbf{H}^T)_{di}}{(\mathbf{W}\mathbf{H}\mathbf{D}_1\mathbf{H}^T)_{di}}. \tag{24}$$

More details on the correctness analysis and convergence proof of (24) can be found in [13].

The details of the algorithm is described in Algorithm 1.

B. Complexity analysis

Based on (13) and (24), we estimate the number of operations for each iteration. When we update $\mathbf{H}$, the cost of multiplications for $\mathbf{W}^T\mathbf{V}\mathbf{D}_1$, $\mathbf{H}(\mathbf{R}_a+\mathbf{R}_b)$, $\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1$ and $\mathbf{H}(\mathbf{R}_c+\mathbf{R}_d)$ are $\mathcal{O}(kmn+kn^2)$, $\mathcal{O}(kn^2)$, $\mathcal{O}(mk^2+nk^2+kn^2)$ and $\mathcal{O}(kn^2)$, respectively. And $\mathbf{R}_a$, $\mathbf{R}_b$, $\mathbf{R}_c$ and $\mathbf{R}_d$ have computational complexity of $\mathcal{O}(n^3)$ each. So the overall cost for $\mathbf{H}$ is $\mathcal{O}(n^3 + kmn)$ as we usually set $k \ll \min(m, n)$; similarly, the cost for $\mathbf{W}$ is $\mathcal{O}(kn^2 + mnk)$. Nevertheless, $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{R}_a$, $\mathbf{R}_b$, $\mathbf{R}_c$ and $\mathbf{R}_d$ are sparse matrices. The overall complexity for $\mathbf{H}$ and $\mathbf{W}$ can be greatly reduced with sparse matrices multiplication. Besides, many optimized libraries for matrix multiplication[1], such as OpenBLAS[2], are currently available to further speed up the computation.

Algorithm 1 The algorithm of ORNMF

Input:
The sequential data matrix $\mathbf{V}$
The constructed matrix $\mathbf{R}$
The parameter α
Output:
The data representation matrix $\mathbf{H}$
1: Initialize $\mathbf{W}$ and $\mathbf{H}$
2: **while** not converges **do**
3: Decompose $\mathbf{R}$ into two nonnegative parts by (11)
4: Calculate the diagonal matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ by (7) and (8)
5: Fixing $\mathbf{W}$, update $\mathbf{H}$ by (13)
6: Fixing $\mathbf{H}$, update $\mathbf{W}$ by (24)
7: **end while**

V. Experiments and Analysis

We conduct experiments on four datasets including one synthetic dataset and three real-world datasets to demonstrate ORNMF's performance and compare it with a few state-of-the-art approaches. The synthetic data is used to present and validate the ordered data representations with ORNMF. The Yale dataset[3] is to test ORNMF's performances against benchmark data with quasi sequential nature. The video sequence dataset [10] that consists of two short videos is to evaluate ORNMF's effectiveness on handling the sequential data. For each experiment, the parameter α of ORNMF in (4) is tuned within [0.1, 0.7]. The corresponding parameters of all competing methods (as listed below) are tuned for their best performances. k-means is applied on the obtained new data representation matrix $\mathbf{H}$ and repeated 20 times to produce the average performances.

A. Baselines for comparison

1) Standard normalized cut (Ncut) in [26].
2) Nonnegative Matrix Factorization minimizing F-norm cost [3].
3) Robust Nonnegative Matrix Factorization (RNMF) [13]: This is a robust formulation of NMF which adopts $L_{2,1}$-norm loss function to alleviate the noise problem.
4) Graph Regularized Nonnegative Matrix Factorization (GNMF) [15] which encodes the geometrical information of the data space into matrix factorization. It has two versions: GNMF minimizing F-norm cost and GNMF$_{KL}$ minimizing KL-divergence cost.
5) Optimal Mean Robust Principal Component Analysis (OMPCA) [27] which can correctly calculate the euclidean distance based mean of robust PCA. It has two implementations: OMPCA and OMCPCA.

[1]https://github.com/attractivechaos/matmul
[2]http://www.openblas.net/
[3]http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

TABLE I: Comparison of Clustering Results (%) on Synthetic Data

	Noises	Ncut	NMF	RNMF	GNMF	GNMF$_{KL}$	OM-RPCA	OM-CRPCA	NMFi	ORNMF
AC	0%	100	84.38	86.25	100	100	100	100	96.25	**100**
	20%	100	100	83.57	83.13	85.00	93.50	100	94.38	**100**
	50%	96.06	96.88	96.25	100	81.25	96.25	96.37	98.13	**100**
NMI	0%	100	91.67	91.67	100	100	100	100	92.75	**100**
	20%	100	100	91.67	91.67	91.67	96.67	100	89.96	**100**
	50%	95.65	95.66	95.18	100	91.67	98.33	98.67	96.36	**100**

6) Nonnegative Matrix Factorization with Interpolated Coefficients (NMFi) [21] which incorporates temporal constraint by adding a simple smoothness on the update rules of NMF.
7) Our proposed Ordered Robust Nonnegative Matrix Factorization(ORNMF).

B. Evaluation metrics

Two metrics, the accuracy (AC) and the normalized mutual information metric (NMI) are used to measure the clustering performance [13]. For both metrics, a higher value indicates better clustering quality. These measurements are widely used by comparing the obtained label of each sample with ground truth in different clustering approaches.

Clustering accuracy (AC) is used to measure the percentage of correct labels obtained. Given a data set containing n images, let l_i and r_i be the the obtained cluster label and label provided from each sample images, respectively. The AC is defined as follows,

$$AC = \frac{\sum_{i=1}^{n} \delta(r_i, map(l_i))}{n} \tag{25}$$

where $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(l_i)$ is the permutation mapping function that maps each cluster label l_i to the equivalent label r_i from the data set. The best mapping can be found by using the Kuhn-Munkres algorithm [28].

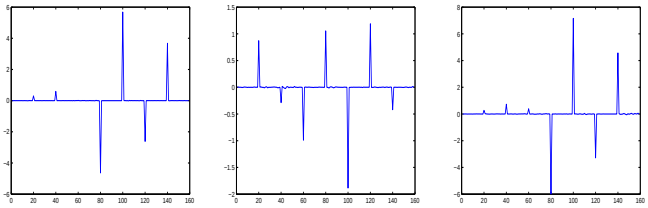
Normalized mutual information (NMI) is used to measure the similarity between the cluster assignments and the pre-existing input labeling of the classes by normalization on the mutual information between them. The normalization used is the average of the entropy of the cluster assignment and that of pre-existing input labeling. Let C and S denote the set of clusters obtained from the ground truth and obtained from our algorithm, respectively, their NMI is defined as follows,

$$NMI = \frac{I(S, C)}{(H(S) + H(C))/2}, \tag{26}$$

where $I(S, C)$ is the mutual information of clustering assignment with pre-existing class labels, and $H(S)$ is the entropy for the clustering assignment.

C. Experiment on synthetic dataset

To build the dataset we first construct a data matrix $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_8] \in \mathbb{R}^{400\times 8}$, in which each element of the data vector $\mathbf{A}_i, i \in \{1, 2, \ldots 8\}$ is a random number between 0 and 1, i.e., $\mathbf{A}_{ji} = [0, 1], j \in \{1, 2, \ldots 400\}$. Multiplying $\mathbf{A}$ with a uniform random weights $s_i \in \mathbb{R}^8$ forms a single synthetic data vector $\mathbf{V}_i$ (=$\mathbf{A}s_i$). We then duplicate $\mathbf{V}_i$ 20 times to construct $\mathbf{V}^i = [\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_{20}] \in \mathbb{R}^{400\times 20}$. Repeating the progress for $\mathbf{V}^i$ 8 times with $\mathbf{A}$ being an invariant and combining all $\mathbf{V}^i$, we finally build our artificial data matrix $\mathbf{V} = [\mathbf{V}^1, \mathbf{V}^2, \ldots, \mathbf{V}^8] \in \mathbb{R}^{400\times 160}$. The experiment is expected to group $\mathbf{V}$ into 8 clusters.

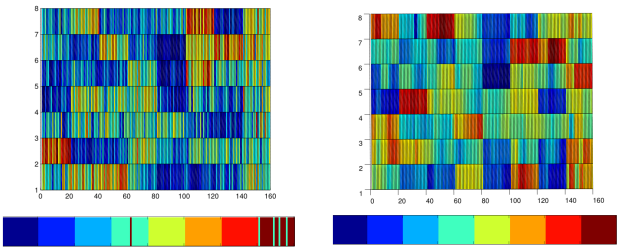


(a) NMF (k=8) (b) ORNMF (k=8) (c) ORNMF (k=50)

Fig. 1: Comparison on inferring the number of clusters.

When data are clean, ORNMF is able to detect the cluster boundaries and infer the number of clusters, which can not be achieved by most NMF based methods. To demonstrate this, we calculate $\mathbf{HR} = [\mathbf{H}_2 - \mathbf{H}_1, \mathbf{H}_3 - \mathbf{H}_2, \ldots, \mathbf{H}_{160} - \mathbf{H}_{159}]$ after obtaining $\mathbf{H}$, and sum the values of each column of $\mathbf{HR}$ to find the peak values. The visualization results of NMF and ORNMF with clean data are shown in Figure 1. It can be seen that NMF in (a) achieves 6 peak values indicating 7 clusters, which is incorrect as the predefined number of clusters is 8. On the contrary, ORNMF finds 8 clusters according to the number of significant peak values as shown in (b), since all the columns in $\mathbf{HR}$ are nearly zeros but the boundaries. To demonstrate the robustness of ORNMF to k, we then randomly chose $k = 50$ and reported result in (c). As we can see, ORNMF can also find 8 clusters. As a result, ORNMF can correctly find the cluster boundaries and get the number of clusters regardless of the value of k. Nevertheless, in case the number of clusters is known beforehand or data is noisy, k-means is still a good option to cluster the data.

According to [9], to further test the robustness of ORNMF, we add 20% and 50% level of Gaussian noise with zero mean and unit variance onto $\mathbf{V}$ and then normalize the corresponding contaminated $\mathbf{V}$ between 0 and 1 to evaluate the performances. As shown in Table I, although all methods have obtained promising results, only ORNMF achieves the perfect performances in all three cases.



(a) NMF (b) ORNMF

Fig. 2: Top figures in (a) and (b) represent the data representation matrix $\mathbf{H}$. The horizontal is the number of data and the vertical represents the reduced dimensionality of each data, k. Every consecutive 20 data belong to one subject. Each bottom figure displays the clustering results, where different colors represent different clusters.

In order to present the performances visually, Figure 2 illustrates the data representation matrix $\mathbf{H}$ and the corresponding clustering results of NMF and ORNMF when data come with 50% level of Gaussian noise. The data representations within each cluster of $\mathbf{H}$ in ORNMF are smooth, which implies that they are of high similarity despite of being contaminated by noises. This is inline with the expectation behind our proposed ORNMF. Hence $\mathbf{H}$ in ORNMF captures the ordered structure effectively, leading to the perfect segmentation result which NMF fails to achieve as shown in the bottom figures.

D. Face clustering

This experiment is to group a set of face images in the Yale dataset into different clusters. The dataset consists of 11 facial images of 15 subjects/clusters - total 165 grayscale images. Each image comes with different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, rightlight, sad, sleepy, surprised, and wink. Before clustering, images are preprocessed. First, we normalize the images in scale and orientation such that eyes are all aligned at the same position horizontally. Then, the facial areas were cropped into the final images for clustering. To

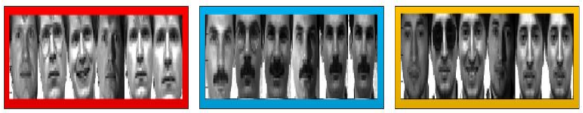
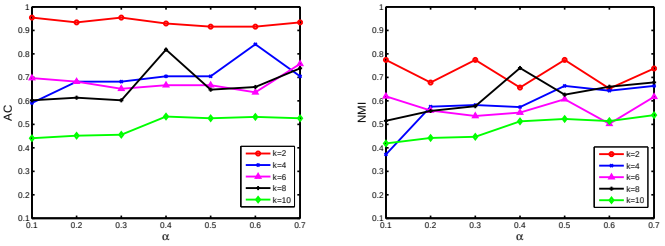


Fig. 3: Samples of Yale Dataset. Different color indicates different clusters.

reduce the computational cost and the memory requirements, all face images are downsized to 32×32 pixels with 256 gray levels per pixel as shown in Figure 3 for example. Thus, each image is represented by a data vector $\mathbf{V}_i \in \mathbb{R}^{1024}$ and we concatenate all these data vectors in order. Strictly speaking, these data are not sequential. However, since the similarities among images of the same subject are much stronger than those from different subjects, the dataset can be regarded as exhibiting a quasi sequential nature.

Similar to the experimental setting in [2], we conduct the experiments for each method on the different number of clusters from 2 to 10 to make a thorough comparison. For a fixed cluster number k, we randomly choose k categories from the dataset, and mix the images of these k categories as the collection $\mathbf{V}$ for clustering.

The clustering results of each k and the overall average performances on all cases are reported in Table II, in which it can be clearly seen that ORNMF significantly outperforms other methods in most cases. Specifically, for average results, compared to the second best method, ORNMF achieves 3.19% improvements in AC and a bigger margin of 7.45% in NMI. We also test the effect of the parameter α, which is first



(a) NMF (b) ORNMF

Fig. 4: **Left**: Comparison of AC w.r.t α. **Right**: Comparison of NMI w.r.t α.

selected in a wide range and then changes within a relative robust range, i.e, from 0.1 to 0.7 with an increment of 0.1. For a clear presentation, Figure 4 illustrates the performances with even k numbers only. It is easy to see that ORNMF produces excellent and relatively stable results, which demonstrates ORNMF is insensitive to α.

E. Video scene segmentation

We extract video sequences from two short animations available free from Internet, same as that in [10]. The videos 1 and 2 contain 19 and 24 sequences, respectively. Each sequence is about 10 s (approximately 300 frames), containing three scenes (that to be segmented). Those frames in which the scene changes are annotated manually and used as our ground truth data. Each sequence is then converted from color to grayscale and resized to a resolution of 129×96. The frames are vectorized to $\mathbf{V}_i \in \mathbb{R}^{12384}$ and concatenated in order to form $\mathbf{V}$ for segmentation. Figure 5 is an example of sequences. This experiment aims to cluster frames into the scene they belong to.

The experimental results on the two videos are shown in Table III. ORNMF outperforms other methods consistently in both videos 1 and 2. For example, the improvements against RNMF are 1.51% and 4.1% in terms of AC and NMI in video 1; 6.68% AC and 5.93% NMI in video 2. This is due to the effectiveness of ORNMF in utilizing the ordered structure of

TABLE II: Comparison of Clustering Results (%) on Yale Dataset

	k	Ncut	NMF	RNMF	GNMF	GNMF$_{KL}$	OM-RPCA	OM-CRPCA	NMFi	ORNMF
AC	2	71.82	78.64	90.91	86.36	86.36	86.36	90.91	86.36	**91.59**
	3	57.27	66.36	66.97	60.61	60.61	75.76	**75.76**	63.64	69.70
	4	52.73	63.18	68.18	65.91	65.91	68.18	63.64	63.64	**70.45**
	5	51.27	58.19	68.00	65.45	67.27	61.82	67.27	69.09	**74.55**
	6	49.84	49.09	57.12	57.58	53.03	53.03	54.55	59.09	**63.64**
	7	39.39	44.25	52.07	50.89	50.05	44.27	57.65	50.65	**52.92**
	8	44.66	45.45	54.55	61.36	46.59	54.55	56.82	52.27	**64.77**
	9	43.78	43.34	49.44	**57.11**	48.28	35.54	44.29	54.55	51.16
	10	36.91	48.36	48.18	48.18	44.55	39.09	41.82	50.91	**52.59**
	Avg.	49.37	56.30	63.05	60.35	59.41	59.19	63.46	61.13	**66.65**
NMI	2	33.97	40.76	56.05	41.27	43.23	43.23	56.05	52.30	**68.65**
	3	32.75	37.69	40.32	37.76	37.76	43.30	43.30	41.25	**52.60**
	4	41.23	43.50	57.51	49.14	47.47	43.75	43.28	51.04	**66.38**
	5	43.74	42.39	52.89	39.14	49.36	44.25	52.08	62.66	**62.94**
	6	44.93	36.07	43.95	40.96	39.31	39.54	48.46	47.74	**52.49**
	7	39.39	44.25	52.07	50.89	50.05	44.27	57.65	44.55	**52.92**
	8	45.51	40.59	46.38	38.14	42.72	46.91	54.16	48.47	**62.64**
	9	43.78	43.34	49.44	**57.11**	48.28	35.54	44.29	53.07	51.16
	10	43.04	46.22	50.69	41.91	46.39	37.72	40.26	53.93	**52.31**
	Avg.	40.93	41.64	49.92	44.04	44.95	42.06	48.84	50.56	**58.01**

TABLE III: Comparison of Clustering Results (%) on Video Sequences Dataset

		Ncut	NMF	RNMF	GNMF	GNMF$_{KL}$	OM-RPCA	OM-CRPCA	NMFi	ORNMF
Video 1	AC	73.37	77.78	77.57	74.46	77.49	77.72	75.97	78.29	**79.08**
	NMI	60.96	66.65	65.33	63.48	67.60	69.40	66.98	66.29	**69.43**
Video 2	AC	79.86	84.41	85.16	78.69	82.12	80.53	82.45	86.51	**91.84**
	NMI	70.31	76.76	77.95	63.21	76.12	73.44	72.68	76.83	**83.88**

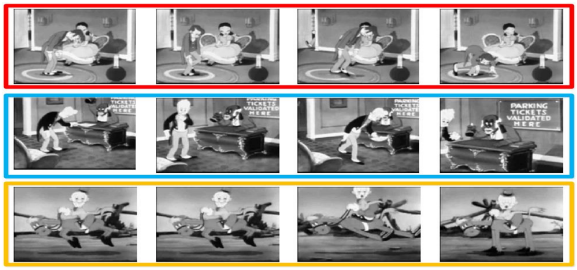
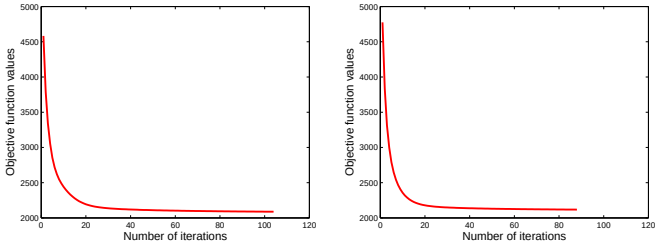


Fig. 5: A sequence with three scenes from the video 1 marked by coloured borders.

video sequences. Because we use multiplicative updating rules to obtain the local optimum, it is important to analyze the convergence. Here we choose a sequence from the video 2 and compare the convergence speed of ORNMF and RNMF. The convergence criteria is $\frac{J_{t+1}-J_t}{J_t} < 10^{-4}$, where J_t is the objective function value in tth iteration. The comparison in Figure 6 shows that the objective function values of ORNMF drop sharply in about 20 iterations and are non-increasing in the whole iterative procedure. And ORNMF takes about 90 iterations to finish the computation, which is 20 iterations less than RNMF. This demonstrates ORNMF converges effectively.



(a) RNMF (b) ORNMF

Fig. 6: Comparison on convergence speed.

F. Human Activity Segmentation

The aim of this experiment is to segment activities in a sequence from the HDM05 Motion Capture Database [29]. The motion sequences were performed by five actors according to the guidelines specified in a script. The script consists of five parts, where each part is subdivided into several scenes. For this experiment we choose the scene 1-1 which contains 9842 frames and 14 activities. However, there is no frame by frame ground truth provided. We assembled the ground truth

TABLE IV: Comparison of Clustering Results (%) on HDM05 dataset

	Ncut	NMF	RNMF	GNMF	GNMF$_{KL}$	OM-RPCA	OM-CRPCA	NMFi	ORNMF
AC	42.13	60.72	58.21	61.14	61.84	58.86	58.86	60.92	**71.00**
NMI	51.14	68.78	65.16	71.93	71.03	72.16	69.89	71.62	**74.15**

by watching the replay of the activities and manually labelling the activities using the activity list provided by [29].

We report clustering performances for this experiment in Table IV. It is clear to see that Ncut performs worst with 42.13% accuracy only, and all the other existing approaches achieve around 60% accuracies, while ORNMF gets more than 70% rate which outperforms other methods with a large margin. This well demonstrates the effectiveness of ORNMF.

VI. Conclusion and future work

We have presented a novel ordered robust nonnegative matrix factorization (ORNMF), which exploits the ordered nature of sequential data. With a neighbour penalty term to enforce the similarity of data presentations, ORNMF has achieved more discriminative and explicit data representations. Using $L_{2,1}$-norm based loss function, ORNMF has effectively dealt with noisy data. A new iterative updating optimization scheme has been derived to solve ORNMF's objective function. In comparison to baselines (NMF, Ncut) and state-of-art approaches (RNMF, GNMF, OM-PCA), ORNMF has achieved the superior performances on both synthetic data, the benchmark dataset (Yale), video sequences and human activities (HDM05) in accuracy and normalized mutual information. Further work includes extending ORNMF into multi-view setting with considering that ordered structure among the data are consistent with different views, and incorporating discriminative information into the framework.

References

[1] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 260–274, 2009.

[2] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1299–1311, 2012.

[3] 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.

[4] A. M. Cheriyadat and R. J. Radke, "Non-negative matrix factorization of partial track data for motion segmentation," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 865–872.

[5] Q. Mo and B. A. Draper, "Semi-nonnegative matrix factorization for motion segmentation with missing data," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 402–415.

[6] N. Guan, D. Tao, L. Lan, Z. Luo, and X. Yang, "Activity recognition in still images with transductive non-negative matrix factorization," in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014, pp. 802–817.

[7] X. Long, H. Lu, Y. Peng, and W. Li, "Graph regularized discriminative non-negative matrix factorization for face recognition," *Multimedia Tools and Applications*, vol. 72, no. 3, pp. 2679–2699, 2014.

[8] B. Ni, P. Moulin, and S. Yan, "Order-preserving sparse coding for sequence classification," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 173–187.

[9] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1019–1026.

[10] F. Wu, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Ordered subspace clustering with block-diagonal priors," *Cybernetics, IEEE Transactions on*, 2015.

[11] Y. Guo, J. Gao, F. Li, S. Tierney, and M. Yin, "Low rank sequential subspace clustering," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–8.

[12] W. Liu, N. Zheng, and Q. You, "Nonnegative matrix factorization and its applications in pattern recognition," *Chinese Science Bulletin*, vol. 51, no. 1, pp. 7–18, 2006.

[13] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using l21-norm," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 673–682.

[14] J. Wang, F. Tian, C. H. Liu, and X. Wang, "Robust semi-supervised non-negative matrix factorization," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[15] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.

[16] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proc. of SDM*, vol. 13. SIAM, 2013, pp. 252–260.

[17] H. Van Hamme, "An on-line nmf model for temporal pattern learning: theory with application to automatic speech recognition," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2012, pp. 306–313.

[18] X.-Y. Huang, W. Li, K. Chen, X.-H. Xiang, R. Pan, L. Li, and W.-X. Cai, "Multi-matrices factorization with application to missing sensor data imputation," *Sensors*, vol. 13, no. 11, pp. 15 172–15 186, 2013.

[19] B. Ju, Y. Qian, M. Ye, R. Ni, and C. Zhu, "Using dynamic multi-task non-negative matrix factorization to detect the evolution of user preferences in collaborative filtering," *PloS one*, vol. 10, no. 8, p. e0135090, 2015.

[20] L. Zafeiriou, S. Nikitidis, S. Zafeiriou, and M. Pantic, "Slow features nonnegative matrix factorization for temporal data decomposition," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 1430–1434.

[21] V. C. Cheung, K. Devarajan, G. Severini, A. Turolla, and P. Bonato, "Decomposing time series data by a non-negative matrix factorization algorithm with temporally constrained coefficients," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 3496–3499.

[22] F. Nie, H. Wang, H. Huang, and C. Ding, "Early active learning via robust representation and structured sparsity," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1572–1578.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[25] C. Ding, T. Li, M. Jordan *et al.*, "Convex and semi-nonnegative matrix factorizations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 45–55, 2010.

[26] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[27] F. Nie, J. Yuan, and H. Huang, "Optimal mean robust principal component analysis," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1062–1070.

[28] M. D. Plummer and L. Lovász, *Matching theory*. Elsevier, 1986.

[29] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05," 2007.