

# Linear and Nonlinear Projective Nonnegative Matrix Factorization

Zhirong Yang, *Member, IEEE*, and Erkki Oja, *Fellow, IEEE*

**Abstract**—A variant of nonnegative matrix factorization (NMF) which was proposed earlier is analyzed here. It is called Projective Nonnegative Matrix Factorization (PNMF). The new method approximately factorizes a projection matrix, minimizing the reconstruction error, into a positive low-rank matrix and its transpose. The dissimilarity between the original data matrix and its approximation can be measured by the Frobenius matrix norm or the modified Kullback-Leibler divergence. Both measures are minimized by multiplicative update rules, whose convergence is proven for the first time. Enforcing orthonormality to the basic objective is shown to lead to an even more efficient update rule, which is also readily extended to nonlinear cases. The formulation of the PNMF objective is shown to be connected to a variety of existing nonnegative matrix factorization methods and clustering approaches. In addition, the derivation using Lagrangian multipliers reveals the relation between reconstruction and sparseness. For kernel principal component analysis with the binary constraint, useful in graph partitioning problems, the nonlinear kernel PNMF provides a good approximation which outperforms an existing discretization approach. Empirical study on three real-world databases shows that PNMF can achieve the best or close to the best in clustering. The proposed algorithm runs more efficiently than the compared nonnegative matrix factorization methods, especially for high-dimensional data. Moreover, contrary to the basic NMF, the trained projection matrix can be readily used for newly coming samples and demonstrates good generalization.

## I. INTRODUCTION

*Nonnegative matrix factorization* (NMF) has become an active research field, with much progress recently both in theory and in practice. The method has been successfully used as a tool in many applications such as signal processing, data mining, pattern recognition and gene expression studies [1], [2], [3], [4], [5], [6]. For an overview, see [7].

Much of this attention stems from the work by Lee and Seung [8]. Their NMF method was shown to be able to generate sparse representations of data. Later a multitude of variants have been proposed to improve NMF. A notable stream of efforts attaches various regularization terms to the original NMF objective to enforce higher sparseness [3], [4], [5], [9]. These methods introduce an additional parameter that balances the sparseness and reconstruction. However, the selection of the parameter value usually relies on exhaustive methods, which hinders their application. Recently it has been shown that the orthogonality constraint on the factorized matrices can significantly enhance the sparseness [10], [11].

Z. Yang and E. Oja are with Department of Information and Computer Science, Aalto University School of Science and Technology, P.O.Box 15400, FI-00076 Aalto, Espoo, Finland, e-mail: {zhirong.yang, erkki.oja}@tkk.fi

This work was financed by the Center of Excellence programme of the Academy of Finland.

Another remarkable finding [12], [13] connects the nonnegative matrix factorization to the classical K-means clustering objective. This in turn associates NMF with Principal Component Analysis (PCA) for effectively finding the K-means clusters [14]. The nonnegativity constraint that provides sparseness can lead to better approximations of the cluster indicators than direct optimization in the discrete space.

In [15], Yuan and Oja introduced a variant called *Projective Nonnegative Matrix Factorization* (PNMF), which approximates a data matrix by its nonnegative subspace projection. Compared with NMF, the PNMF algorithm involves considerably fewer parameters but is able to generate a much sparser factorized matrix, which is desired for both feature extraction and clustering. Its relations to sparseness and to K-means clustering have been analyzed in [16], [17], with some applications in facial image feature extraction and text document clustering. Although the success of PNMF has been empirically demonstrated, its theoretical convergence analysis has been lacking so far.

A key in the learning algorithms for variants of NMF is a multiplicative update rule that is able to maintain positivity. For PNMF, it was shown in [16], [11] that a nonnegative multiplicative version of a PCA learning rule (“Oja rule”) suggested earlier in [18] can be used for computing the PNMF.

Recently, a similar idea as PNMF, based on the Frobenius norm, was discussed in [19] as a particular case of their Convex Nonnegative Matrix Factorization. The authors called it the Cluster-NMF due to its closeness to the K-means clustering, which was also noted in [17].

After a brief review of PNMF in Section II, we extend and complete the above preliminary efforts with the following new contributions:

- 1) Formal convergence analysis of the original PNMF algorithms is given in Sections II-B and II-C. We decouple the auto-association of the factorizing matrix by using the Lagrangian technique and prove that the resultant regularized learning objective is monotonically decreasing at each iteration. We provide the proof for the PNMF based on the Frobenius norm (Section II-B) as well as for the divergence-based algorithm (Section II-C).
- 2) Orthonormal PNMF (OPNMF) is introduced in Section II-D. An additional orthonormality constraint is imposed on the weight matrix and the Lagrangian technique is used to derive the corresponding learning rule, where the monotonicity of the regularized objective at each iteration is proven. The Lagrangian solution finds jointly a PNMF approximation and steers the factorizing matrix

towards the Stiefel manifold of orthogonal matrices. The learning rule turns out to be a simplified version of the PNMf learning rule and reveals the underlying reason that leads to high sparseness by using PNMf. It is related to the nonnegative multiplicative version of a PCA learning rule (“Oja rule”) [11].

- 3) Nonlinear extension of PNMf in Section II-E. We notice that the multiplicative update rule for PNMf based on the Frobenius norm relies only on inner products between data vectors. This enables us to apply PNMf for any nonnegative kernel Principal Component Analysis, which in turn suggests many new applications of nonlinear PNMf such as graph partitioning.
- 4) Comparison of PNMf with two recent NMF variants in Sections III and IV. In addition to the classical NMF and K-means algorithms, we also compare PNMf with two more recent algorithms, the *Orthogonal Nonnegative Matrix Factorization* [10] and *Convex Nonnegative Matrix Factorization* [19]. Some theoretical considerations are given in Section III, and in Section IV, empirical results on three real-world datasets show that PNMf can achieve the best or close to the best in clustering. Furthermore, PNMf is more advantageous in terms of high sparseness and fast training for high-dimensional data.
- 5) A new application of PNMf for recovering the projection matrix in a nonnegative mixture model. Our experimental results show that PNMf can achieve small recovery errors for various source distributions.
- 6) Comparison of PNMf with the approach of discretizing eigenvectors in Section IV-C. PNMf is the only nonnegative matrix factorization method that can handle the nonnegative principal component analysis among the selected algorithms. In terms of small reconstruction error, PNMf also defeats a two-step approach that first applies eigendecomposition and then discretizes the eigenvectors [20].
- 7) Theoretical justification of moving a term in the generic multiplicative update rule. Some auxiliary mathematical results are given in the Appendix. Especially, in mathematical convergence analysis of a multiplicative update rule, one often needs to move a term from the numerator to the denominator or vice versa in order to maintain the nonnegativity. We present a common technique with theoretical justification of such moving by introducing an additional tight upper bounding term in the auxiliary function.

## II. PROJECTIVE NONNEGATIVE MATRIX FACTORIZATION

Given a nonnegative input matrix  $\mathbf{X} \in \mathbb{R}_+^{m \times n}$ , whose columns are typically  $m$ -dimensional data vectors, one tries to find a nonnegative projection matrix  $\mathbf{P} \in \mathbb{R}_+^{m \times m}$  of rank  $r$  such that (see Table 1 for notations)

$$\mathbf{X} \approx \hat{\mathbf{X}} \equiv \mathbf{P}\mathbf{X} \quad (1)$$

In particular, *Projective Nonnegative Matrix Factorization* (PNMF) calculates the factorization  $\mathbf{P} = \mathbf{W}\mathbf{W}^T$  with  $\mathbf{W} \in$

$\mathbb{R}_+^{m \times r}$ . Compared with the Non-negative Matrix Factorization (NMF) [8] where  $\mathbf{X} \approx \mathbf{W}\mathbf{H}$ , PNMf replaces the second factorizing matrix  $\mathbf{H}$  with  $\mathbf{W}^T\mathbf{X}$ . This brings PNMf close to non-negative Principal Component Analysis. A trivial solution  $\mathbf{W} = \mathbf{I}$  appears when  $r = m$ , which will produce zero error but is practically useless. Useful PNMf results usually appear when  $r \ll m$  for real-world applications.

The term “projective” refers to the fact that  $\mathbf{W}\mathbf{W}^T$  would indeed be a projection matrix if  $\mathbf{W}$  were an orthogonal matrix:  $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ . It turns out that in PNMf learning,  $\mathbf{W}$  becomes approximately, although not exactly, orthogonal. As will be seen, this has positive consequences in sparseness of the approximation, orthogonality of the factorizing matrix, decreased computational complexity in learning, close equivalence to clustering, generalization of the approximation to new data without heavy re-computations, and easy extension to a non-linear kernel method with wide applications for optimization problems.

TABLE I  
SOME FREQUENTLY USED NOTATIONS.

|                             |   |
|-----------------------------|---|
| $m, n, r$                   | data dimensionality, sample size, reduced rank of matrix                |
| $\mathbb{R}_+^{m \times r}$ | space of non-negative $m \times r$ matrices                             |
| $\mathbf{X}$                | data matrix of size $m \times n$  |
| $\hat{\mathbf{X}}$          | approximated data matrix of size $m \times n$                           |
| $\mathbf{Z}$                | $Z_{ij} = X_{ij} / \hat{X}_{ij}$  |
| $\mathbf{W}$                | factorizing matrix of size $m \times r$                                 |
| $\mathbf{H}$                | factorizing matrix of size $r \times n$                                 |
| $\mathbf{U}, \mathbf{V}$    | factorizing matrices of size $n \times r$                               |
| $\bar{\mathbf{U}}$          | binary matrix of size $n \times r$                                      |
| $\mathbf{K}$                | kernel matrix or a similarity matrix between samples, size $n \times n$ |
| $\Lambda$                   | Lagrangian multiplier matrix of size $r \times r$                       |
| $\Psi$                      | Lagrangian multiplier matrix of size $r \times n$                       |
| $i, a, b$                   | $1, \dots, m$   |
| $j, s, t$                   | $1, \dots, n$   |
| $k, l$                      | $1, \dots, r$   |

There exist two popularly used approaches that quantify the approximation error (1). One is based on the Frobenius norm of the matrix difference and the other employs a modified Kullback-Leibler divergence. In this formulation, PNMf was first introduced in [15]. In this section, these two error criteria are recapitulated and new convergence justifications are given.

### A. Iterative Lagrangian solution

Before presenting the PNMf algorithms, it should be emphasized that nonnegative learning is essentially a constrained optimization problem. More constraints can be considered in addition to nonnegativity, for example, the orthonormality and the replacement  $\mathbf{H} = \mathbf{W}^T\mathbf{X}$ . Here we employ a common procedure to derive the iterative solutions using the Lagrangian technique, which is the best known method to handle the constraints.

Given the unconstrained objective  $\mathcal{J}(\mathbf{W})$  to be minimized together with a set of equality constraints  $\{F_p(\mathbf{W}) = 0\}$ , the generalized objective can be formulated by introducing the Lagrangian multipliers  $\{\lambda_p\}$ :

$$\tilde{\mathcal{J}}(\mathbf{W}, \{\lambda_p\}) = \mathcal{J}(\mathbf{W}) + \sum_p \lambda_p F_p(\mathbf{W}).$$

For nonnegative optimization, we construct the auxiliary function  $G(\mathbf{W}, \mathbf{W}')$  for  $\tilde{\mathcal{J}}(\mathbf{W}, \{\lambda_p\})$  as described in Appendix B by using one or more bounds in Appendices C and D. An update rule that includes the Lagrangian multipliers

$$\mathbf{W}' = \tilde{\pi}(\mathbf{W}, \{\lambda_p\}) \quad (2)$$

can then be obtained by setting  $\partial G(\mathbf{W}, \mathbf{W}')/\partial \mathbf{W}' = 0$ . Finally, the quantities  $\{\lambda_p\}$  are solved by using the Karush-Kuhn-Tucker (K.K.T.) conditions and substituted into (2).

The resulting update rule

$$\mathbf{W}' = \pi(\mathbf{W}) \quad (3)$$

is called an *iterative Lagrangian solution* for the constrained optimization problem. The definition of auxiliary function guarantees that  $\tilde{\mathcal{J}}(\mathbf{W}, \{\lambda_p\})$  is monotonically decreasing if one repeatedly applies (3). The iterations will converge to a local minimum if there is a lower bound of  $\tilde{\mathcal{J}}(\mathbf{W}, \{\lambda_p\})$ . Furthermore, the multiplicative nature of the update rule assures the parameters remain nonnegative during the optimization.

Notice that the iterative Lagrangian solution does not necessarily remain in the manifold specified by  $\{F_p(\mathbf{W}) = 0\}$ . Instead, it can jointly minimize  $\mathcal{J}(\mathbf{W})$  and force  $\mathbf{W}$  to approach the constraint manifold. Actually  $\mathbf{W}$  never exists in the manifold for some constraints. For example, the orthonormality requires many entries of a nonnegative matrix become exactly zero. If  $\mathbf{W}$  is initialized in such a manifold, the convergence would be very poor because the multiplicative update rule cannot recover a zero entry to be positive.

### B. PNMF based on the Frobenius norm (Euclidean PNMF)

The Frobenius norm measures the Euclidean distance between two (vectorized) matrices:

$$\|\mathbf{A} - \mathbf{B}\|_F = \sqrt{\sum_{ij} (A_{ij} - B_{ij})^2}$$

Equipped with such a metric, PNMF looks for a solution of the optimization problem

$$\underset{\mathbf{W} \geq 0}{\text{minimize}} \quad \mathcal{J}_F(\mathbf{W}) = \frac{1}{2} \|\mathbf{X} - \mathbf{W}\mathbf{W}^T \mathbf{X}\|_F^2. \quad (4)$$

A nonnegative update rule for this optimization can be developed as follows [15]. First, the unconstrained derivative of  $\mathcal{J}_F$  with respect to  $\mathbf{W}$  is

$$\begin{aligned} \frac{\partial \mathcal{J}_F(\mathbf{W})}{\partial W_{ik}} &= -2(\mathbf{X}\mathbf{X}^T \mathbf{W})_{ik} + (\mathbf{W}\mathbf{W}^T \mathbf{X}\mathbf{X}^T \mathbf{W})_{ik} \\ &\quad + (\mathbf{X}\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W})_{ik}. \end{aligned}$$

Inserting

$$\eta_{ik} = \frac{W_{ik}}{(\mathbf{W}\mathbf{W}^T \mathbf{X}\mathbf{X}^T \mathbf{W})_{ik} + (\mathbf{X}\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W})_{ik}}$$

into the additive update rule

$$W'_{ik} = W_{ik} - \eta_{ik} \frac{\partial \mathcal{J}_F(\mathbf{W})}{\partial W_{ik}},$$

one obtains the multiplicative update rule

$$W'_{ik} = W_{ik} \frac{2(\mathbf{X}\mathbf{X}^T \mathbf{W})_{ik}}{(\mathbf{W}\mathbf{W}^T \mathbf{X}\mathbf{X}^T \mathbf{W})_{ik} + (\mathbf{X}\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W})_{ik}}. \quad (5)$$

Based on preliminary results given in the Appendix, we are now ready to relate this update rule to an iterative Lagrangian solution. First we rewrite the problem (4) as

$$\underset{\mathbf{W} \geq 0, \mathbf{H} \geq 0}{\text{minimize}} \quad \mathcal{J}_F(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2. \quad (6)$$

$$\text{subject to } \mathbf{H} = \mathbf{W}^T \mathbf{X}. \quad (7)$$

Then we have

*Theorem 1:* The update rule (5) is an iterative Lagrangian solution of (6)-(7).

*Proof:* The generalized objective is

$$\tilde{\mathcal{J}}_F(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 + \text{Tr}(\Psi^T (\mathbf{H} - \mathbf{W}^T \mathbf{X})) \quad (8)$$

by introducing Lagrangian multipliers  $\{\Psi_{jk}\}$ . Next, we construct

$$G_F(\mathbf{W}, \mathbf{W}') \equiv \text{Tr}(-2\mathbf{X}^T \mathbf{W}' \mathbf{H} - \Psi^T \mathbf{W}'^T \mathbf{X}) \quad (9)$$

$$+ \sum_{ik} \frac{(\mathbf{W}\mathbf{H}\mathbf{H}^T)_{ik} W_{ik}'^2}{W_{ik}} \quad (10)$$

$$- \text{Tr}(\mathbf{A}^T \mathbf{W}) + \sum_{ik} A_{ik} \left( \frac{W_{ik}'^2 + W_{ik}^2}{2W_{ik}} \right) \quad (11)$$

$$+ \text{Tr}(\mathbf{X}^T \mathbf{X} + \Psi^T \mathbf{H}) \quad (12)$$

as an auxiliary function (see Appendix B) of

$$\mathcal{L}_F(\mathbf{W}') \equiv \tilde{\mathcal{J}}_F(\mathbf{W}', \mathbf{H}) \quad (13)$$

$$= \text{Tr}(-2\mathbf{X}^T \mathbf{W}' \mathbf{H} - \Psi^T \mathbf{W}'^T \mathbf{X}) \quad (14)$$

$$+ \text{Tr}(\mathbf{W}'^T \mathbf{W}' \mathbf{H}\mathbf{H}^T) \quad (15)$$

$$+ 0 \quad (16)$$

$$+ \text{Tr}(\mathbf{X}^T \mathbf{X} + \Psi^T \mathbf{H}) \quad (16)$$

Here we denote  $\mathbf{A} = 2\mathbf{X}\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W}$  for notational brevity.  $G_F$  tightly upper bounds  $\mathcal{L}_F$  as we apply the *quadratic upper bound* (10) to (14) and the *moving-term upper bound (type II)* (11) to (15) according to Appendices C and E.

Setting  $\partial G_F(\mathbf{W}, \mathbf{W}')/\partial W'_{ik} = 0$ , we get

$$W'_{ik} = W_{ik} \frac{(2\mathbf{X}\mathbf{H}^T + \mathbf{X}\Psi^T + 2\mathbf{X}\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W})_{ik}}{(2\mathbf{W}\mathbf{H}\mathbf{H}^T + 2\mathbf{X}\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W})_{ik}}. \quad (17)$$

The Lagrangian multipliers can be determined by using the K.K.T. conditions. According to

$$\frac{\partial \tilde{\mathcal{J}}_F(\mathbf{W}, \mathbf{H})}{\partial \mathbf{H}} = -2\mathbf{W}^T \mathbf{X} + 2\mathbf{W}^T \mathbf{W}\mathbf{H} + \Psi = \mathbf{0},$$

one obtains

$$\begin{aligned} \Psi &= 2\mathbf{W}^T \mathbf{X} - 2\mathbf{W}^T \mathbf{W}\mathbf{H}, \\ \mathbf{X}\Psi^T &= 2\mathbf{X}\mathbf{X}^T \mathbf{W} - 2\mathbf{X}\mathbf{H}^T \mathbf{W}^T \mathbf{W}. \end{aligned} \quad (18)$$

Substituting (7) and (18) into (17), the update rule becomes identical to (5).  $\square$

### C. PNMF based on the Kullback-Leibler divergence (Divergence PNMF)

The difference of two matrices can also be asymmetrically measured by the modified Kullback-Leibler divergence:

$$D(\mathbf{A}||\mathbf{B}) = \sum_{ij} \left( A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right).$$

PNMF based on such a dissimilarity measure solves the optimization problem

$$\underset{\mathbf{W} \geq 0}{\text{minimize}} \quad \mathcal{J}_D(\mathbf{W}) = D(\mathbf{X}||\mathbf{W}\mathbf{W}^T\mathbf{X}). \quad (19)$$

The gradient

$$\begin{aligned} \frac{\partial \mathcal{J}_D(\mathbf{W})}{W_{ik}} &= - \sum_j \mathbf{X}_{ij} (\mathbf{W}^T \mathbf{X})_{kj} / (\mathbf{W}\mathbf{W}^T \mathbf{X})_{ij} \\ &\quad - \sum_j \mathbf{X}_{ij} \sum_a \mathbf{W}_{ak} \mathbf{X}_{aj} / (\mathbf{W}\mathbf{W}^T \mathbf{X})_{aj} \\ &\quad + \sum_j \left( (\mathbf{W}^T \mathbf{X})_{kj} + \sum_a \mathbf{W}_{ak} \mathbf{X}_{ij} \right) \end{aligned}$$

also implies a multiplicative update rule:

$$W'_{ik} = W_{ik} \frac{(\mathbf{Z}\mathbf{X}^T \mathbf{W})_{ik} + (\mathbf{X}\mathbf{Z}^T \mathbf{W})_{ik}}{\sum_j (\mathbf{W}^T \mathbf{X})_{kj} + \left( \sum_j X_{ij} \right) \left( \sum_a W_{ak} \right)} \quad (20)$$

by putting the unsigned negative terms to the numerator and positive terms to the denominator [15], [16], where we introduce a new matrix  $\mathbf{Z}$  with

$$Z_{ij} = \frac{X_{ij}}{(\mathbf{W}\mathbf{W}^T \mathbf{X})_{ij}}.$$

for notational simplicity.

The problem (19) can be rewritten as

$$\underset{\mathbf{W} \geq 0, \mathbf{H} \geq 0}{\text{minimize}} \quad D(\mathbf{X}||\mathbf{W}\mathbf{H}) \quad (21)$$

$$\text{subject to} \quad \mathbf{H} = \mathbf{W}^T \mathbf{X}. \quad (22)$$

We then have

*Theorem 2:* The update rule (20) is an iterative Lagrangian solution of (21)-(22).

*Proof:* The generalized objective is

$$\tilde{\mathcal{J}}_D(\mathbf{W}, \mathbf{H}) = D(\mathbf{X}||\mathbf{W}\mathbf{H}) + \text{Tr} \left( \Psi^T (\mathbf{H} - \mathbf{W}^T \mathbf{X}) \right), \quad (23)$$

by using Lagrangian multipliers  $\{\Psi_{jk}\}$ .

We then construct

$$G_D(\mathbf{W}, \mathbf{W}') \equiv \quad (24)$$

$$\sum_{ij} (\mathbf{W}' \mathbf{H}^T)_{ij} \quad (25)$$

$$- \sum_{ij} X_{ij} \sum_k \frac{W_{ik} H_{kj}}{\sum_l W_{il} H_{lj}} \left( \log W'_{ik} H_{kj} - \log \frac{W_{ik} H_{kj}}{\sum_l W_{il} H_{lj}} \right) \quad (26)$$

$$- \text{Tr} \left( \Psi^T \mathbf{W}'^T \mathbf{X} \right) \quad (27)$$

$$+ \sum_{ik} A_{ik} W'_{ik} - \sum_{ik} A_{ik} W_{ik} - \sum_{ik} \left( A_{ik} W_{ik} \log \frac{W'_{ik}}{W_{ik}} \right) \quad (28)$$

$$+ \text{Tr}(\Psi^T \mathbf{H}) + \sum_{ij} (X_{ij} \log X_{ij} - X_{ij}), \quad (29)$$

as an auxiliary function of

$$\mathcal{L}_D(\mathbf{W}') \equiv \tilde{\mathcal{J}}_D(\mathbf{W}', \mathbf{H}) \quad (30)$$

$$= \sum_{ij} (\mathbf{W}' \mathbf{H})_{ij} \quad (31)$$

$$- \sum_{ij} X_{ij} \log (\mathbf{W}' \mathbf{H})_{ij} \quad (32)$$

$$- \text{Tr} \left( \Psi^T \mathbf{W}'^T \mathbf{X} \right) \quad (33)$$

$$+ 0 \quad (34)$$

$$+ \text{Tr}(\Psi^T \mathbf{H}) + \sum_{ij} (X_{ij} \log X_{ij} - X_{ij}) \quad (35)$$

Here we denote  $\mathbf{A} = \mathbf{X}\mathbf{Z}^T \mathbf{W}$  for notational brevity.  $G_D$  tightly upper bounds  $\mathcal{L}_D$  as we apply the *Jensen upper bound* to (26) to (32) and the *moving-term upper bound (type I)* (28) to (34) according to Appendices C and E.

Setting  $\partial G_D(\mathbf{W}, \mathbf{W}') / \partial W'_{ik} = 0$ , we get

$$W'_{ik} = W_{ik} \frac{(\mathbf{Z}\mathbf{X}^T \mathbf{W})_{ik} + A_{ik}}{\sum_j H_{kj} - (\mathbf{X}\Psi^T)_{ik} + A_{ik}}. \quad (36)$$

Again, the Lagrangian multipliers can be obtained by using the K.K.T conditions. According to

$$\frac{\partial \tilde{\mathcal{J}}_D(\mathbf{W}, \mathbf{H})}{\partial H_{jk}} = -(\mathbf{W}^T \mathbf{Z})_{jk} + \sum_i W_{ik} + \Psi_{jk} = 0,$$

one obtains

$$\Psi_{jk} = (\mathbf{W}^T \mathbf{Z})_{jk} - \sum_i W_{ik}$$

$$\left( \mathbf{X}\Psi^T \right)_{ik} = (\mathbf{X}\mathbf{Z}^T \mathbf{W})_{ik} - \left( \sum_j X_{ij} \right) \left( \sum_i W_{ik} \right). \quad (37)$$

Substituting (22) and (37) into (36), the update rule becomes identical to (20).  $\square$

### D. Orthonormality

Orthonormality is usually desired for a projection. First, an orthonormal matrix forms a basis of a subspace, which facilitates geometric interpretation and signal reconstruction.

Second, two nonnegative vectors are orthogonal if and only if their non-zero dimensions do not overlap. For some problems such as clustering, this property is especially useful for approximating discrete solutions. PNMf with the orthonormality constraint is called *Orthonormal Projective Nonnegative Matrix Factorization* (OPNMf) to be distinguished from the ones described in Sections II-B and II-C.

Surprisingly, the enforced orthonormality constraint leads to an even simpler Lagrangian solution for PNMf based on the Frobenius norm. Consider the optimization problem:

$$\underset{\mathbf{W} \geq 0, \mathbf{H} \geq 0}{\text{minimize}} \quad \mathcal{J}_F^\perp(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 \quad (38)$$

$$\text{subject to} \quad \mathbf{H} = \mathbf{W}^T \mathbf{X} \quad (39)$$

$$\mathbf{W}^T \mathbf{W} = \mathbf{I} \quad (40)$$

Applying the procedure described in Section II-A, we obtain:

*Theorem 3:* The update rule

$$W'_{ik} = W_{ik} \frac{(\mathbf{X}\mathbf{X}^T \mathbf{W})_{ik}}{(\mathbf{W}\mathbf{W}^T \mathbf{X}\mathbf{X}^T \mathbf{W})_{ik}}, \quad (41)$$

is an iterative Lagrangian solution of (38)-(40).

The proof can be found in Appendix F. It is interesting that the update rule (41) drops the term  $\mathbf{X}\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W}$  compared with (5), which makes the multiplicative updates even faster. This simplification has also been justified in adaptive PCA learning [16], [11], as the gradient term  $\hat{\nabla} = -\mathbf{X}\mathbf{X}^T \mathbf{W} + \mathbf{X}\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W}$  has little effect in learning the principal directions [21].

Some empirical results have shown that the update rule (5) can also yield a highly orthogonal factorizing matrix  $\mathbf{W}$  [15], [16], [11]. This can be interpreted by our derivation procedure in Appendix F, where the Lagrangian multipliers  $\Lambda_{kl}$  equal zero in the derivation. That is, the orthonormality constraint is ineffective during iterative updates. This finding also reveals that the orthonormality force has already implicitly been included in the PNMf learning, which thus explains the performance resemblance of PNMf and OPNMf in terms of orthogonality [11]. Note, however, that computationally the rule (41) is simpler than (5).

For the divergence-based PNMf with orthonormality constraint

$$\underset{\mathbf{W} \geq 0, \mathbf{H} \geq 0}{\text{minimize}} \quad D(\mathbf{X} \|\mathbf{W}\mathbf{H}) \quad (42)$$

$$\text{subject to} \quad \mathbf{H} = \mathbf{W}^T \mathbf{X}, \quad (43)$$

$$\mathbf{W}^T \mathbf{W} = \mathbf{I}, \quad (44)$$

we can similarly derive an multiplicative update rule

$$W'_{ik} = W_{ik} \frac{B_{ik} + (\mathbf{W}\mathbf{W}^T \mathbf{C})_{ik}}{C_{ik} + (\mathbf{W}\mathbf{W}^T \mathbf{B})_{ik}}, \quad (45)$$

where  $\mathbf{B} = \mathbf{Z}\mathbf{X}^T \mathbf{W} + \mathbf{X}\mathbf{Z}^T \mathbf{W}$  and  $C_{ik} = \sum_j (\mathbf{W}^T \mathbf{X})_{kj} + \sum_j X_{ij} \sum_a W_{ak}$  for notational brevity.

*Theorem 4:* The update rule (45) is an iterative Lagrangian solution of (42)-(44).

The proof can be found in Appendix G. Similar to the Euclidean case, previous empirical studies have shown that the Divergence PNMf update rule (20) can already yield a

highly orthogonal factorizing matrix  $\mathbf{W}$  [15], [16]. This can be explained again by investigating the values  $\Lambda_{kl}$  in (90) which indicates how effective the orthonormality constraint is. Notice the matrix  $\mathbf{Z}$  actually is the ratio of the data entries over their approximates. Thus all  $Z_{ij}$  should be close to one if the approximation is good. In this case one can find that  $\mathbf{\Lambda}$  approaches the zero matrix in (90). That is, a good approximation leads to an inactive orthonormality constraint. This finding indicates that the explicit orthonormality constraint is not necessary, if there exists a good initialization that provides a fairly good approximation, such as the K-means cluster indicators added by a small constant that are used in [10], [19]. In this case, the update rule (20) is more advantageous as it requires less computation. On the other hand, random initialization that results in a poor approximation probably leads to effective orthonormality constraints. The update rules (45) and (20) may then behave very differently.

### E. Nonnegative Kernel PCA

The optimization problem (38)-(40) is equivalent to the classical *Principal Component Analysis* with the additional nonnegativity constraint. Furthermore, the multiplicative update rule (41) requires only  $\mathbf{X}\mathbf{X}^T$  instead of the original data matrix. Thus one can easily extend the iterative Lagrangian solution to nonlinear cases by using the kernel technique.

Denote  $\Phi = [\phi(\mathbf{x}_1) \phi(\mathbf{x}_2) \dots \phi(\mathbf{x}_n)]^T$ , where  $\mathbf{x}_i$  are the data vectors and they are implicitly mapped into another vector space  $\mathcal{S}$  by a function  $\phi$ . The PNMf objective based on Frobenius norm with orthonormality constraint in  $\mathcal{S}$  can now be formulated as

$$\underset{\mathbf{U} \geq 0}{\text{minimize}} \quad \mathcal{J}_K(\mathbf{U}) = \|\Phi - \mathbf{U}\mathbf{U}^T \Phi\|_F^2 \quad (46)$$

$$\text{subject to} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I}. \quad (47)$$

Define the *kernel matrix*  $\mathbf{K} = \Phi\Phi^T$  with  $K_{st} = \phi(\mathbf{x}_s)^T \phi(\mathbf{x}_t)$ . Suppose all  $K_{st} \geq 0$ . The optimization problem (46)-(47) leads to the multiplicative update rule

$$U'_{jk} = U_{jk} \frac{(\mathbf{K}\mathbf{U})_{jk}}{(\mathbf{U}\mathbf{U}^T \mathbf{K}\mathbf{U})_{jk}}. \quad (48)$$

In addition to the nonlinearity extension, the update rule (48) is particularly suitable for applications where the input is given in the form of pairwise relationship and the original data vectors are not available. For example, one may perform nonnegative *Normalized Cut* [22] to find multiple partitions in an undirected graph which is represented by a nonnegative affinity matrix.

### F. Projective Semi-Nonnegative Matrix Factorization

In the above we assume that the data matrix or the kernel matrix contains no negative elements. Otherwise the multiplicative update rules using the principle described in Section A cannot guarantee that the updated parameters remain nonnegative. The nonnegativity restriction can be removed by decomposing a matrix into its positive and negative parts and employing the *linear lower bound* and/or *quadratic lower*

bound in Section D and slightly modifying the multiplicative update rule.

Let us take the Semi-Nonnegative Kernel PCA (Semi-NKPCA) for example. The modifications of other algorithms can be obtained similarly. Consider now the matrix  $\mathbf{K}$  has both positive and negative entries. One can always separate the positive and negative parts of  $\mathbf{K}$  by calculating

$$\begin{aligned} K_{st}^+ &= (|K_{st}| + K_{st})/2 \\ K_{st}^- &= (|K_{st}| - K_{st})/2. \end{aligned}$$

In this way  $\mathbf{K} = \mathbf{K}^+ - \mathbf{K}^-$  and the entries of both  $\mathbf{K}^+$  and  $\mathbf{K}^-$  are all nonnegative.

We next rewrite the PNMf optimization problem as

$$\underset{\mathbf{U} \geq \mathbf{0}}{\text{minimize}} \quad \mathcal{J}_K^\pm(\mathbf{U}, \mathbf{V}) = \|\Phi - \mathbf{U}\mathbf{V}^T\Phi\|_F^2, \quad (49)$$

$$\text{subject to} \quad \mathbf{V} = \mathbf{U}, \quad (50)$$

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}, \quad (51)$$

and obtain its iterative Lagrangian solution

$$U'_{jk} = U_{jk} \frac{(\mathbf{K}^+\mathbf{U} + \mathbf{U}\mathbf{U}^T\mathbf{K}^-)_{jk}}{(\mathbf{K}^-\mathbf{U} + \mathbf{U}\mathbf{U}^T\mathbf{K}^+\mathbf{U})_{jk}}$$

by using the derivation procedure in Section II-A (see Appendix H).

### G. Stabilization

The Lagrangian solution of PNMf iteratively performs one of the multiplicative update rules presented in Section II-B-II-E. However, we find the convergence path is often very zigzag in practice and some numerical computation problems may occur after several iterations if the factorizing matrix  $\mathbf{W}$  does not have a proper magnitude.

Theoretically, let us look at the update rule (41) for example. If the (Frobenius or Euclidean) matrix norm of the current  $\mathbf{W}$  is very large, then the norm will become very small in the next iteration because the norm is cubically powered in the denominator. On the other hand, if the current norm is very small, it will become very large in the next iteration. Consequently, the norm of  $\mathbf{W}$  will change drastically between odd and even iterations. A similar problem happens for other method such as the *Orthogonal Nonnegative Matrix Factorization* (ONMF) [10].

We propose to overcome this problem by introducing one more parameter  $\rho$ . For PNMf based on the Frobenius norm, the modified objective becomes

$$\underset{\mathbf{W} \geq \mathbf{0}}{\text{minimize}} \quad \hat{\mathcal{J}}_F(\mathbf{W}, \rho) = \|\mathbf{X} - \rho\mathbf{W}\mathbf{W}^T\mathbf{X}\|_F^2. \quad (52)$$

Fixing  $\mathbf{W}$ , the global optimal  $\rho^*$  can be solved by setting  $\partial\hat{\mathcal{J}}_F(\mathbf{W}, \rho)/\partial\rho = 0$ :

$$\rho^* = \frac{\text{Tr}(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W})}{\text{Tr}(\mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}\mathbf{W}^T)}.$$

Similarly, we can modify the divergence-based PNMf as

$$\underset{\mathbf{W} \geq \mathbf{0}}{\text{minimize}} \quad \hat{\mathcal{J}}_D(\mathbf{W}, \varrho) = D(\mathbf{X}||\varrho\mathbf{W}\mathbf{W}^T\mathbf{X}), \quad (53)$$

with the optimal

$$\varrho^* = \frac{\sum_{ij} X_{ij}}{\sum_{ij} (\mathbf{W}\mathbf{W}^T\mathbf{X})_{ij}}.$$

Next, fixing  $\rho$  or  $\varrho$ , the optimal  $\mathbf{W}$  given its current estimate can be found by putting  $\rho^*$  or  $\varrho^*$  in the denominator of (5), (41), (20) or (45). Or equivalently, one can apply the original multiplicative update rule and then calculate

$$W_{ik}^{\text{new}} = W'_{ik} \sqrt{\frac{\text{Tr}(\mathbf{W}'^T\mathbf{X}\mathbf{X}^T\mathbf{W}')}{\text{Tr}(\mathbf{W}'\mathbf{W}'^T\mathbf{X}\mathbf{X}^T\mathbf{W}'\mathbf{W}'^T)}} \quad (54)$$

or

$$W_{ik}^{\text{new}} = W'_{ik} \sqrt{\frac{\sum_{ij} X_{ij}}{\sum_{ij} (\mathbf{W}\mathbf{W}^T\mathbf{X})_{ij}}} \quad (55)$$

as the new estimate.

If  $\mathbf{W}\mathbf{W}^T\mathbf{X}$  well approximates  $\mathbf{X}$ , both  $\rho^*$  and  $\varrho^*$  approach one and the modified objective is equivalent to the original one. Thus  $\rho$  or  $\varrho$  serves as an intermediate variable that stabilizes and speeds up the algorithm especially in early iterations.

The stabilization (54) requires extra  $O(m^2r)$  computations at each iteration if  $\mathbf{X}\mathbf{X}^T$  is precomputed. It can be empirically shown that the simple normalization

$$\mathbf{W}^{\text{new}} = \frac{\mathbf{W}'}{\|\mathbf{W}'\|_2}$$

can achieve similar stabilization effect [15], [16], where  $\|\mathbf{W}'\|_2$  equals the square root of maximal eigenvalue of  $\mathbf{W}'^T\mathbf{W}'$ . The normalization approach requires only  $O(mr^2)$  extra computation. Yet, whether the above normalization would affect the monotonicity of the Lagrangian iterations remains theoretically unknown. For the divergence case, the stabilization (55) requires  $O(mr)$  computing cost if  $\sum_{ij} X_{ij}$  and  $\bar{x}_i = \sum_j X_{ij}$  are precomputed.

### H. On the optimization of original objective functions

The underlying principle for deriving an iterative Lagrangian solution is relaxed constraint optimization. Here relaxation means the allowance of small violation of the involved constraints and leads to a finite regularized learning objective by using Lagrangian multipliers. This in turn gives sound interpretation of two forces, one for optimizing the original objective and the other for steering the estimate to approach the constraint manifold. In this sense, it is critical to optimize the regularized objective instead of the original objective.

For readers who still have concern on the monotonic decrease in original PNMf (not OPNMf) objective, we provide an alternative theoretical justification by means of approximated upper-bound minimization. Take the PNMf based on the Frobenius norm (4) for example. We linearize the objective function  $\mathcal{J}_F(\mathbf{W}') = \frac{1}{2}\|\mathbf{X} - \mathbf{W}'\mathbf{W}'^T\mathbf{X}\|_F^2$  by its first-order

Taylor expansion at  $\mathbf{W}$ :

$$\hat{\mathcal{J}}_F(\mathbf{W}') \approx \frac{1}{2} \text{Tr}(\mathbf{X}^T \mathbf{X}) \quad (56)$$

$$- \text{Tr}(\mathbf{W}'^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \quad (57)$$

$$+ \frac{1}{2} \text{Tr}(\mathbf{W}'^T (\mathbf{W} \mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W} + \mathbf{X} \mathbf{X}^T \mathbf{W} \mathbf{W}^T \mathbf{W})). \quad (58)$$

Next, we apply the *linear lower bound* to the second line (57) and construct its auxiliary function

$$\begin{aligned} \hat{G}_F(\mathbf{W}', \mathbf{W}) = & \frac{1}{2} \text{Tr}(\mathbf{X}^T \mathbf{X}) \\ & - \sum_{ik} (\mathbf{X} \mathbf{X}^T \mathbf{W})_{ik} W'_{ik} \left( 1 + \log \frac{W'_{ik}}{W_{ik}} \right) \\ & + \text{Tr}(\mathbf{W}'^T (\mathbf{W} \mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W} + \mathbf{X} \mathbf{X}^T \mathbf{W} \mathbf{W}^T \mathbf{W})). \end{aligned}$$

Setting  $\partial \hat{G}_F(\mathbf{W}', \mathbf{W}) / \partial W' = \mathbf{0}$ , we also obtain the PNMf multiplicative update rule (5). Therefore, the approximated PNMf objective function is non-increasing under the PNMf multiplicative updates. Because  $\mathcal{J}(\mathbf{W}')$  is a quartic function, the above linearization (56)-(58) is good only if  $\mathbf{W}$  is kept small, which also necessitates the stabilization described in Section II-G.

### III. CONNECTIONS TO RELATED WORK

Table II summarizes four variants of nonnegative matrix factorization based on the Frobenius norm. Compared with the *Nonnegative Matrix Factorization* (NMF) approximation [8], [23], PNMf replaces  $\mathbf{H}$  with  $\mathbf{W}^T \mathbf{X}$  in the objective. It has been shown that such a replacement leads to much higher sparseness which is desired for extracting part-based representations [15], [16], [11]. NMF is known to be sensitive to the starting values of  $\mathbf{W}$  and  $\mathbf{H}$  and heuristic initialization is therefore required [24]. By contrast, the sparseness can always be achieved by using PNMf even with different random seeds [16], [17].

The projective replacement has also been proposed in the *Convex Nonnegative Matrix Factorization* (CNMF) [19]. By using a different matrix for reconstruction, CNMF is able to achieve better approximation accuracy for training data. It may however poorly generalize to the testing data because there are twice as many parameters to be learned.

The orthonormality constraint proposed in Section II-D is another approach to increase sparseness and reduce the number of local minima. This idea has also been adopted by the *Orthogonal Nonnegative Matrix Factorization* (ONMF) [10], where the objective of NMF is accompanied with the orthonormality constraint on  $\mathbf{W}$  or  $\mathbf{H}$ .

Compared with NMF and ONMF, the CNMF or PNMf optimization requires only the correlation or kernel matrix instead of the original data vectors, which is advantageous for nonlinear extensions. This property also enables fast training when the dimensionality is much higher than the number of samples. For feature extraction, CNMF or PNMf can output a

projection matrix which can be used to transform the unseen data while NMF and ONMF cannot.

It is well-known that K-means clustering is tightly related to nonnegative matrix factorizations [12]. Assume we want to cluster a set of  $m$ -dimensional vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n$  into  $r$  clusters  $\mathcal{C}_1, \dots, \mathcal{C}_r$ . The classical K-means clustering uses  $r$  cluster centroids  $\mathbf{m}_1, \dots, \mathbf{m}_r$  to characterize the clusters. The objective function is

$$\mathcal{J}_{\text{K-means}} = \sum_{k=1}^r \sum_{j \in \mathcal{C}_k} \|\mathbf{x}_j - \mathbf{m}_k\|^2.$$

As shown in [12], [13], this can be written as

$$\mathcal{J}_{\text{K-means}} = \text{Tr}(\mathbf{X}^T \mathbf{X}) - \text{Tr}(\bar{\mathbf{U}}^T \mathbf{X}^T \mathbf{X} \bar{\mathbf{U}}) \quad (59)$$

with  $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_n]$  the data matrix, and  $\bar{\mathbf{U}}$  the indicator matrix for the clusters:  $\bar{U}_{jk} = 1$  if vector  $\mathbf{x}_j$  belongs to cluster  $\mathcal{C}_k$ , zero otherwise. Thus  $\bar{\mathbf{U}}$  is a binary ( $n \times r$ ) matrix, whose columns are orthogonal if each sample belongs to one and only one cluster. Minimizing  $\mathcal{J}_{\text{K-means}}$  under the binary and orthogonality constraints on  $\bar{\mathbf{U}}$  is equivalent to maximizing  $\text{Tr}(\bar{\mathbf{U}}^T \mathbf{X}^T \mathbf{X} \bar{\mathbf{U}})$  under these constraints.

The PNMf has a direct relation to this. Consider the PNMf criterion for the transposed data matrix  $\mathbf{X}^T$ :

$$\begin{aligned} \|\mathbf{X}^T - \mathbf{U} \mathbf{U}^T \mathbf{X}^T\|^2 = & \\ \text{Tr}(\mathbf{X}^T \mathbf{X}) - 2\text{Tr}(\mathbf{U}^T \mathbf{X}^T \mathbf{X} \mathbf{U}) + \text{Tr}(\mathbf{U} \mathbf{U}^T \mathbf{X}^T \mathbf{X} \mathbf{U} \mathbf{U}^T) \end{aligned}$$

Together with the orthonormality constraint  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ , the last term becomes  $\text{Tr}(\mathbf{U}^T \mathbf{X}^T \mathbf{X} \mathbf{U})$  and the whole PNMf criterion becomes exactly equal to the K-means criterion  $\mathcal{J}_{\text{K-means}}$  in (59), except for the binary constraint.

PNMF solves the *Principal Component Analysis* (PCA) problem with the nonnegativity constraint. It corresponds to the nonnegative version of rule (68) in Appendix A that implements the PCA algorithm [18]. Given the input  $\mathbf{X}$  and the output  $\mathbf{Y} = \mathbf{W}^T \mathbf{X}$  one can form the non-normalized Hebbian update direction  $\mathbf{X} \mathbf{Y}^T = \mathbf{X} \mathbf{X}^T \mathbf{W}$ , to which the rule (68) attaches the regularized term  $-\mathbf{W} \mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}$  that comes from the normalization. Using this technique, Yang and Laaksonen obtained the *Nonnegative Linear Hebbian Network* (NLHN) algorithm that iteratively applies the update rule (41) [11]. However, the derivation in [11] is mainly inspired by the biological neural networks and lack of mathematical interpretations. Here we have given formal convergence analysis in Section II-D. Our derivation also demonstrates that the rule (68) may be extended to learn nonnegative projections in many other problems.

Recently it was reported that the graph partitioning problem can be solved by using eigenvectors of a real symmetric matrix. Two notable methods, for example, are *Normalized Cuts* [22] and the *Modularity* method [25]. These approaches can however identify only two partitions at a time. For simultaneous multi-partition finding, it remains difficult to convert the eigenvectors to binary cluster indicators. Yu and Shi have proposed a discretization algorithm called POD that finds a binary orthonormal matrix closest to the one composed of eigenvectors after some rotations [20]. Nevertheless, the

TABLE II  
SUMMARY OF SOME VARIANTS OF NONNEGATIVE MATRIX FACTORIZATION BASED ON THE FROBENIUS NORM.

| method          | problem formulation<br>$\mathbf{W} \geq \mathbf{0}, \mathbf{H} \geq \mathbf{0}, \tilde{\mathbf{W}} \geq \mathbf{0}$      | sparseness | data vectors<br>required | generalized to new data<br>without iterations | number of<br>parameters |
|-----------------|--|------------|--------------------------|---|-------------------------|
| NMF             | $\min \ \mathbf{X} - \mathbf{WH}\ _F$  | low        | yes                      | no  | $(m+n)r$                |
| ONMF            | $\min \ \mathbf{X} - \mathbf{WH}\ _F$<br>$\mathbf{W}^T \mathbf{W} = \mathbf{I}$ or $\mathbf{H}\mathbf{H}^T = \mathbf{I}$ | high       | yes                      | no  | $(m+n)r$                |
| CNMF            | $\min \ \mathbf{X} - \tilde{\mathbf{W}}\mathbf{W}^T \mathbf{X}\ _F$  | high       | no                       | yes   | $2mr$                   |
| PNMF<br>(OPNMF) | $\min \ \mathbf{X} - \mathbf{W}\mathbf{W}^T \mathbf{X}\ _F$<br>$\mathbf{W}^T \mathbf{W} = \mathbf{I}$                    | high       | no                       | yes   | $mr$                    |

resulting matrix is not necessarily optimal in terms of the partitioning objective.

What we propose here is to relax the binary constraint to nonnegativity and integrate the latter into the kernel PCA problem which is solved by multiplicative update rules. The empirical results shown in Section IV-C indicate that our method can outperform the POD approach.

#### IV. EXPERIMENTS

We have performed empirical studies of the PNMf algorithms for typical problems: feature extraction/compression, clustering, generalization for new data, and kernel clustering. Throughout, we used three real-world datasets:

- *Iris Plants Database* (*iris*), a dataset that contains 150 instances of 4 positive-valued attributes. The samples belong to three iris classes, *Setosa*, *Versicolour*, and *Virginica*, each including 50 instances. This small-scale dataset is selected mainly for comparison with the following larger-scale databases.
- *Optical Recognition of Handwritten Digits* (*digit*), a subset containing “0”, “2”, “4” and “6” selected from the UCI optical handwritten digit database. There are 2237 samples of 62 nonnegative integer attributes. This dataset is used to demonstrate the algorithm behavior when samples are much more than attributes.
- *ORL Database of Faces* (*orl*), a set of face images taken at the AT&T laboratory at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). There are 400 gray-scale images from 40 distinct subjects and of size  $92 \times 112$ . We have used this dataset to study the case where the dimensionality is much higher than the number of samples.

For comparisons, four other algorithms have been chosen: The Lloyd’s algorithm for K-means [26], NMF [8], [23], ONMF [10], and CNMF [19]. For better comparison, we only use Frobenius norm-based algorithms as there is no divergence-based implementation of ONMF and CNMF.

##### A. Clustering

Clustering is an important application of NMF and its variants. We have adopted two measurements, *purity* and *entropy* which are widely used in nonnegative learning literature, for comparing clustering results. These measurements provide fair comparison because they do not rely on assumptions of

the cluster distributions and quantify clustering performance by using ground truth class information which is independent of compared algorithms.

Suppose there is ground truth data that labels the samples by one of  $q$  classes. Purity is given by

$$\text{purity} = \frac{1}{n} \sum_{k=1}^r \max_{1 \leq l \leq q} n_k^l, \quad (60)$$

where  $n_k^l$  is the number of samples in the cluster  $k$  that belong to original class  $l$ . A larger purity value indicates better clustering performance. Entropy measures how classes are distributed on various clusters. Following [10], [27], the entropy of the entire clustering solution is computed as:

$$\text{entropy} = -\frac{1}{n \log_2 q} \sum_{k=1}^r \sum_{l=1}^q n_k^l \log_2 \frac{n_k^l}{n_k}, \quad (61)$$

where  $n_k = \sum_l n_k^l$ . Generally, a smaller entropy value corresponds to a better clustering quality.

We set  $r = q$  and repeated each algorithm on each dataset 100 times with different random seeds for initialization. The mean and standard deviation of the purities and entropies of each algorithm-dataset pair are shown in Table III (a) and (b), respectively. From these statistics, we can see that PNMf performs the best for all datasets in terms of purity. For the *orl*, K-means ranks top in terms of entropy, but we notice that PNMf as the runner-up performs very closely to the winner.

We have also compared the sparseness of factorizing matrices computed by the methods based on nonnegative matrix factorization. Given an  $u \times v$  nonnegative matrix  $\mathbf{A}$ , its sparseness is quantified by the fraction of number of entries that are smaller than the mean  $\sum_{pq} A_{pq}/uv$  against the total number of entries  $uv$ . A fraction close to one corresponds to an asymmetric distribution of entry values, where most entries are near zero and thus lead to high sparseness. The means and standard deviations of resulting sparseness are shown in Table III (c). PNMf achieves the highest sparseness for the *iris* and *digit* datasets and is the runner-up, which is very close to best, for the *orl* dataset. By contrast, NMF yields much less sparse factorizing matrices. ONMF has the same sparseness as PNMf for *iris* and *digit* but lower for *orl*. The sparseness of CNMF depends on data. For *iris* where dimensionality is far less than cardinality,  $\mathbf{W}$  is sparser than  $\tilde{\mathbf{W}}$ . On the other hand,  $\tilde{\mathbf{W}}$  is sparser than  $\mathbf{W}$  for *orl* where dimensionality is much larger than cardinality.

We have recorded the consumed time of the compared algorithms for the clustering task with the selected datasets.

TABLE III

CLUSTERING PERFORMANCE: (A) PURITIES, (B) ENTROPIES, AND (C) SPARSENESS. EACH ENTRY SHOWS THE MEAN±DEVIATION OF THE CLUSTERING RESULTS WITH 100 DIFFERENT RANDOM INITIALIZATIONS. BOLDFACE NUMBERS REPRESENT THE BEST MEAN IN THE CORRESPONDING ROW.

(a)

| dataset | K-means          | NMF              | ONMF             | CNMF      | PNMF             |
|---------|------------------|------------------|------------------|-----------|------------------|
| iris    | 0.83±0.10        | 0.78±0.05        | 0.85±0.03        | 0.81±0.04 | <b>0.97±0.01</b> |
| digit   | 0.92±0.10        | <b>0.98±0.00</b> | <b>0.98±0.00</b> | 0.96±0.00 | <b>0.98±0.00</b> |
| orl     | <b>0.72±0.03</b> | 0.47±0.03        | <b>0.72±0.02</b> | 0.68±0.02 | <b>0.72±0.03</b> |

(b)

| dataset | K-means          | NMF              | ONMF             | CNMF      | PNMF             |
|---------|------------------|------------------|------------------|-----------|------------------|
| iris    | 0.31±0.10        | 0.42±0.08        | 0.30±0.05        | 0.36±0.03 | <b>0.09±0.03</b> |
| digit   | 0.13±0.10        | <b>0.08±0.00</b> | <b>0.08±0.00</b> | 0.12±0.00 | <b>0.08±0.00</b> |
| orl     | <b>0.15±0.01</b> | 0.34±0.02        | 0.17±0.01        | 0.19±0.01 | 0.16±0.02        |

(c)

| dataset | NMF       |           | ONMF             |           | CNMF             |           | PNMF             |
|---------|-----------|-----------|------------------|-----------|------------------|-----------|------------------|
|         | <b>W</b>  | <b>H</b>  | <b>W</b>         | <b>H</b>  | <b>W</b>         | <b>W</b>  | <b>W</b>         |
| iris    | 0.74±0.05 | 0.81±0.11 | <b>0.96±0.01</b> | 0.82±0.06 | 0.81±0.02        | 0.93±0.01 | <b>0.96±0.01</b> |
| digit   | 0.93±0.00 | 0.79±0.00 | <b>0.97±0.00</b> | 0.77±0.00 | 0.93±0.00        | 0.94±0.00 | <b>0.97±0.00</b> |
| orl     | 0.65±0.00 | 0.67±0.00 | 0.96±0.00        | 0.51±0.00 | <b>0.98±0.00</b> | 0.91±0.00 | 0.97±0.00        |

The comparison also includes a recently proposed NMF implementation based on projected gradient called *NMFPG* [28], which is available in its author's website<sup>1</sup>. The experiment was repeatedly performed 100 times on a computer with an Intel Core Duo CPU, 2G DDR2 main memory and Linux Ubuntu 7.10 operating system. The resulting means in seconds are shown in Table IV.

The NMF algorithms, especially the projected gradient implementation, run quickly for the datasets *iris* and *digit* of low dimensionality. However, they become much slower for high-dimensional data in *orl*. The NMFPG algorithm is even more problematic in this case. In addition, The PNMF training is faster than the other nonnegative matrix factorization methods for the *orl* dataset. The speed advantage mainly comes from two factors. Firstly, PNMF as well as CNMF does not rely on the original data vectors but only their correlation matrix which can be calculated before the iterations. This is particularly beneficial when the dimensionality of data is high, for example, in the *orl* database. Secondly, PNMF has a simpler iterative Lagrangian solution as there is only one matrix to be learned in PNMF while the other three have to update two matrices at each iteration. ONMF inherits the dimensionality problem of NMF because it uses one of the NMF update rules.

### B. Projection recovery

We have tested the proposed PNMF method for recovering a nonnegative projection matrix  $\hat{\mathbf{P}} = \hat{\mathbf{G}}\hat{\mathbf{G}}^T$ , or equivalently its factorizing matrix  $\hat{\mathbf{G}}$ . Consider a quasi-projection mixture model

$$\mathbf{X} = \mathbf{P}\mathbf{Y} = \mathbf{G}\mathbf{G}^T\mathbf{Y},$$

where  $\mathbf{X} \in \mathbb{R}_+^{m \times n}$  is an observed nonnegative matrix;  $\mathbf{Y} \in \mathbb{R}^{m \times n}$  is some source matrix;  $\mathbf{P} = \mathbf{G}\mathbf{G}^T$ ; and  $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_r] \in \mathbb{R}^{m \times r}$  is a nonnegative quasi-orthonormal matrix, i.e.  $\mathbf{g}_i^T \mathbf{g}_j / \|\mathbf{g}_i\| \|\mathbf{g}_j\|$  very small if  $i \neq j$ . The noisy factorizing matrix  $\mathbf{G}$  is generated from a truly orthonormal

matrix  $\hat{\mathbf{G}}$  by setting  $\mathbf{G} = \hat{\mathbf{G}} + \epsilon$  with a small nonnegative noise matrix  $\epsilon$ .

Note now that if  $\mathbf{G}$  contains no noise, then the solution to

$$\min_{\mathbf{W} \geq 0} \|\mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X}\|_F^2 = \min_{\mathbf{W} \geq 0} \|\mathbf{G}\mathbf{G}^T\mathbf{Y} - \mathbf{W}\mathbf{W}^T\mathbf{G}\mathbf{G}^T\mathbf{Y}\|_F^2$$

subject to  $\mathbf{W}^T\mathbf{W} = \mathbf{I}$  is given by  $\mathbf{W} = \mathbf{G}\mathbf{R}$  with  $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ . Because both  $\mathbf{W}$  and  $\mathbf{G}$  are nonnegative,  $\mathbf{R}$  must be a permutation matrix. As shown in Section II-D and [11], the orthonormality constraint  $\mathbf{W}^T\mathbf{W} = \mathbf{I}$  approximately holds. Thus we would expect  $\mathbf{W}$  in the PNMF (or OPNMF) solution to closely resemble a column-permuted version of the original matrix  $\mathbf{G}$ .

We generated two sets of  $\mathbf{Y}$ s: one contains both nonnegative and negative entries, and the other contains only nonnegative ones. Euclidean PNMF (semi-nonnegative version) is used in the recovery test for the first set while Divergence PNMF for the second. For the first set of  $\mathbf{Y}$ s, we have tried three distributions: (1) *uniform*, uniform distribution in  $[-0.5, 0.5]$ ; (2) *gauss*, zero-mean radial Gaussian of unitary variance; (3) *laplace*, zero-mean radial Laplace distribution of unitary variance. For the second set of  $\mathbf{Y}$ s, we have also tried the above type of distributions but shifted the range of *uniform* to  $[0, 1]$  and taken the absolute values of (2) *gauss* and (3) *laplace*.

The matrix  $\mathbf{G}$  was generated as follows. First we randomly drew a nonnegative matrix  $\mathbf{\Gamma} \in [0, 1]^{m \times r}$  by the uniform distribution. Next, we binarized  $\mathbf{\Gamma}$  by setting the largest entry in each row to one and the others to zero. We repeated the above sampling until each column in  $\mathbf{\Gamma}$  contains at least one non-zero entry. Then we normalized each column of  $\mathbf{\Gamma}$  to unitary norm. In this way we obtained the truly orthonormal matrix  $\hat{\mathbf{G}}$ . Finally, the quasi-orthonormal matrix  $\mathbf{G}$  is formed by drawing a noise matrix  $\epsilon \in [0, 0.01]^{m \times r}$  by uniform distribution and adding it to  $\hat{\mathbf{G}}$ .

The PNMF algorithms take the mixed matrix  $\mathbf{X}$  as input and output a quasi-orthonormal matrix  $\hat{\mathbf{W}}$ . We next computed its true orthonormal version  $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_r]$  by binarization and column-wise normalization. If  $\hat{\mathbf{W}}$  well recovers  $\hat{\mathbf{G}}$ ,

<sup>1</sup><http://www.csie.ntu.edu.tw/~cjlin/nmf/>

TABLE IV  
MEAN ( $\mu$ ) AND STANDARD DEVIATION ( $\sigma$ ) OF TRAINING TIME (IN SECONDS) IN FORMAT  $\mu \pm \sigma$ . BOLDFACE NUMBERS REPRESENT THE BEST MEAN IN THE CORRESPONDING ROW

| dataset                 | NMF       | NMFPG            | ONMF      | CNMF      | PNMF             |
|-------------------------|-----------|------------------|-----------|-----------|------------------|
| iris                    | 0.85±0.17 | <b>0.22±0.10</b> | 1.32±0.01 | 1.14±0.00 | 1.34±0.00        |
| digit ( $\times 10^2$ ) | 2.15±0.06 | <b>0.07±0.10</b> | 3.94±0.43 | 2.79±0.15 | 2.72±0.14        |
| orl ( $\times 10^3$ )   | 3.21±0.02 | 26.52±5.16       | 6.16±0.01 | 0.24±0.00 | <b>0.23±0.00</b> |

TABLE V  
MEAN ( $\mu$ ) AND STANDARD DEVIATION ( $\sigma$ ) OF RECOVERY ERROR OF THE MIXTURE MATRIX IN FORMAT  $\mu \pm \sigma (\times 10^{-2})$ .

| Ytype           | uniform   | gauss     | laplace   |
|-----------------|-----------|-----------|-----------|
| Euclidean PNMf  | 1.07±2.12 | 0.37±0.11 | 1.76±2.87 |
| Divergence PNMf | 4.33±3.36 | 0.46±0.04 | 4.21±3.39 |

their difference should be small after proper ordering of the columns. That is,  $\tilde{\mathbf{W}} = [\hat{\mathbf{w}}_{l_1}, \dots, \hat{\mathbf{w}}_{l_r}]$ . The  $k$ -th permuted column index  $l_k$  can be determined by

$$l_k = \arg \max_{k'} \left( \hat{\mathbf{G}}^T \tilde{\mathbf{W}} \right)_{k'j}.$$

We can then measure the recovery quality by calculating the relative error  $\|\tilde{\mathbf{W}} - \hat{\mathbf{G}}\|_F / \|\hat{\mathbf{G}}\|_F = \|\tilde{\mathbf{W}} - \hat{\mathbf{G}}\|_F / r$ , where a small value indicates better projection recovery. We have repeated the experiment for each  $\mathbf{Y}$  type and PNMf algorithm for 100 times with different random seeds. The statistics of the results are recorded in Table V.

It can be seen that the mean relative recovery errors are less than five percent for all  $\mathbf{Y}$  distribution types using either algorithm. In particular, we have found that the recovery is the best when  $\mathbf{Y}$  is drawn from the *gauss* distribution, where Euclidean and Divergence PNMf algorithms can respectively achieve 0.37% and 0.46% mean relative error with small standard deviation. Remarkably, Divergence PNMf works very robustly in this case, resulting in only 0.04% standard deviation. By contrast, Euclidean PNMf is more stable across different initial  $\mathbf{Y}$  matrices, where mean errors are less than two percent for all three tested distribution types.

### C. Nonnegative Kernel Principal Component Analysis

Given an  $n \times n$  symmetric matrix  $\mathbf{K}$ , consider the trace maximization problem

$$\underset{\bar{\mathbf{U}}}{\text{maximize}} \quad \mathcal{J}_K(\bar{\mathbf{U}}) = \text{Tr}(\bar{\mathbf{U}}^T \mathbf{K} \bar{\mathbf{U}}) \quad (62)$$

$$\text{subject to} \quad \text{for all } j, \sum_{k=1}^r \bar{U}_{jk} = 1, \bar{\mathbf{U}} \in \{0, 1\}^{n \times r} \quad (63)$$

Such optimization is required in many clustering or graph partitioning algorithms such as *spectral partitioning* [29], [30], *Normalized Cut* [22] and the *modularity* method [25], where the matrix  $\mathbf{K}$  is derived from the similarity or affinity matrix. These algorithms mostly resort to finding eigenvectors of  $\mathbf{K}$ . Nevertheless, as multi-cluster indicators each column of  $\bar{\mathbf{U}}$  has to be binary valued. It remains difficult to obtain such indicators from the real-valued eigenvectors. One possibility is a discretization algorithm called POD that finds a non-negative orthonormal matrix closest to the one composed of eigenvectors after some rotations [20]. We have employed

the POD implementation from its author's website<sup>2</sup>, which takes a matrix of eigenvectors as input and returns discretized cluster indicators. The POD discretization depends on the initial rotation matrix. We repeated such discretization 100 times and selected the one with the best  $\mathcal{J}_K$ .

*Kernel K-means* (KK-means) (see e.g. [31]) is another approach that finds local optima of (62)-(63). It extends the *K-means* method [26] to nonlinear cases via the kernel principle [32]. Denote  $\phi$  a vector function that implicitly maps a sample  $\mathbf{x}$  to another space  $\mathcal{S}$  and  $n_k$  the number of samples in the  $k$ th cluster  $\mathcal{C}_k$ . The squared Euclidean distance between the  $j$ th sample  $\phi_j \equiv \phi(\mathbf{x}_j)$  and the  $k$ th cluster mean is

$$\begin{aligned} d_{jk}^2 &= \left\| \phi_j - \frac{1}{n_k} \sum_{t \in \mathcal{C}_k} \phi_t \right\|^2 \\ &= \phi_j^T \phi_j - \frac{2}{n_k} \sum_{t \in \mathcal{C}_k} \phi_j^T \phi_t + \frac{1}{n_k^2} \sum_{s \in \mathcal{C}_k} \sum_{t \in \mathcal{C}_k} \phi_s^T \phi_t \\ &= K_{jj} - \frac{2}{n_k} \sum_{t \in \mathcal{C}_k} K_{jt} + \frac{1}{n_k^2} \sum_{s \in \mathcal{C}_k} \sum_{t \in \mathcal{C}_k} K_{st}, \end{aligned}$$

where  $K_{st} = \phi_s^T \phi_t$ . The KK-means algorithm thus iteratively groups the samples to their nearest cluster by the above distance measurement. The cluster means need no explicit computation as they are not required in cluster indication. The matrix  $\bar{\mathbf{U}}$  is then obtained by setting  $\bar{U}_{jk} = 1$  if the  $j$ th sample belongs to the  $k$ th cluster and 0 otherwise. The KK-means result also depends on the initial setting of cluster indicators. We used 100 different initial guesses by uniform random sampling and took the one that achieves the largest objective.

Among the five compared nonnegative matrix factorization algorithms, only PNMf can handle the nonnegative kernel principal component analysis. Following [10], [19], we took the best resulting matrix from POD and KK-means and added 0.2 to it as the initialized matrix of PNMf. After PNMf converged, we discretized the PNMf output by setting the maximum entry of each row to 1 and the others to 0.

We have adopted two types of kernel in our experiments. One is the linear kernel

$$K_{st}^{\text{linear}} = \mathbf{x}_s^T \mathbf{x}_t,$$

i.e.  $\phi(\mathbf{x}_j) = \mathbf{x}_j$ , because of its simplicity. The other is the *radial basis function* (RBF) kernel

$$K_{st}^{\text{RBF}} = \exp \left( -\frac{\|\mathbf{x}_s - \mathbf{x}_t\|^2}{2\sigma^2} \right)$$

<sup>2</sup><http://www.seas.upenn.edu/~jshi/software/>

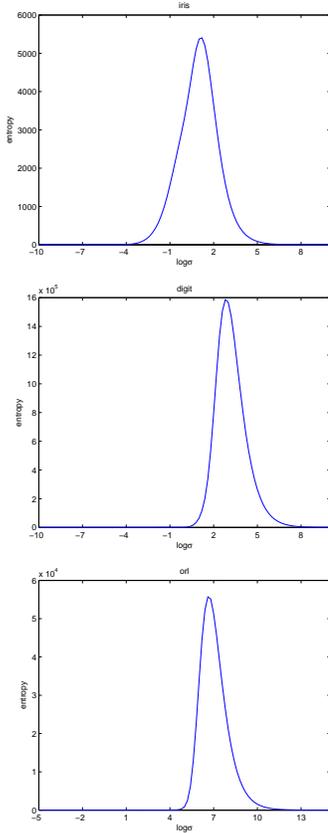


Fig. 1. Entropies of off-diagonal elements of the RBF kernel matrix with varying  $\sigma$ .

which is widely used in machine learning and data mining. Here the kernel width  $\sigma$  as a free parameter needs to be adjusted. We have employed an information-theoretic method to automatically determine the  $\sigma$  parameter as follows. Notice the diagonal elements of  $\mathbf{K}$  contribute nothing to clustering. Therefore we can consider only off-diagonal entries. It can be seen that  $K_{st}$  ( $s \neq t$ ) approaches 0 if  $\sigma \rightarrow 0$  and approaches 1 if  $\sigma \rightarrow \infty$ . That is, the uncertainty or Shannon information of  $K_{st}$  is close to zero at both ends. Starting from a sufficiently large value and then decreasing  $\sigma$  steadily, one can find a peak corresponding to the  $\mathbf{K}$  with locally maximal information. In this work we have used entropy for information measurement. To avoid dominance of some feature over the others, we first normalize the samples in a dataset by subtracting their mean and dividing each feature by their standard deviation. The entropy peaks for the selected datasets are shown in Figure 1, to which the corresponding  $\sigma$ 's are 2.18, 7.25, and 101.57 for *iris*, *digit*, and *orl*, respectively. Without losing clustering accuracy, the matrix columns of  $\bar{\mathbf{U}}$  are re-weighted to be unitary for better comparison:

$$\tilde{U}_{jk} = \frac{\bar{U}_{jk}}{\sqrt{\sum_t \bar{U}_{tk}^2}}.$$

We further normalize the objective  $\mathcal{J}_K^N(\bar{\mathbf{U}}) = \sqrt{\mathcal{J}_K(\tilde{\mathbf{U}})} / \|\mathbf{K}\|_F$  for better visual illustration. As PCA is known to achieve the global optimum of the objective

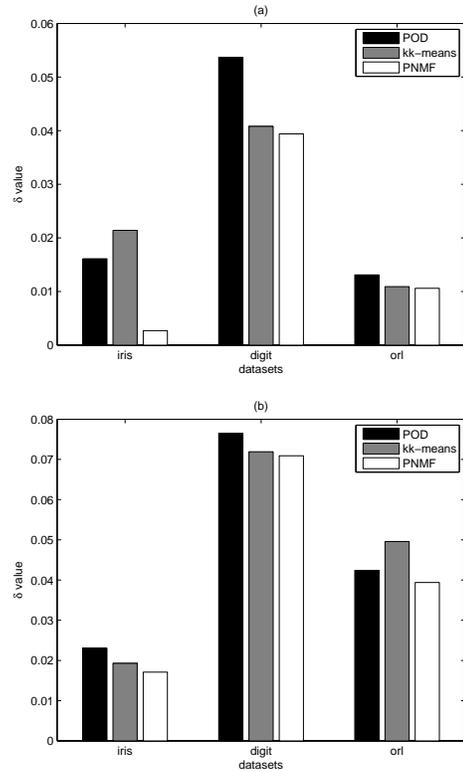


Fig. 2. Relative objectives of POD, KK-means, and PNMF with (a) linear and (b) RBF kernels.

(62) without the constraint (63), we recorded the difference between the PCA output and the resulting objectives:

$$\delta(\bar{\mathbf{U}}) = \mathcal{J}_{K, \text{PCA}}^N(\bar{\mathbf{U}}) - \mathcal{J}_K^N(\bar{\mathbf{U}}).$$

The relative objectives are shown in Figure 2, where a smaller  $\delta$  value indicates better performance. It can be seen that PNMF outperforms POD and KK-means for all selected datasets with both kernel types. Although the eigendecomposition finds a global optimum without the binary constraint, the extra discretization employed by POD however does not take  $\mathbf{K}$  into account. The POD output therefore can be farther from the optimum compared with PNMF. KK-means inherits both the advantage and disadvantage of K-means. In our experiment, it runs fast but easily falls into poor local optima. This can be partially remedied by repeating the algorithm with many different starting points. Occasionally KK-means can achieve performance next to PNMF.

## V. CONCLUSIONS

We have proposed a new variant of nonnegative matrix factorization called PNMf using the approximation scheme  $\mathbf{X} \approx \mathbf{W}\mathbf{W}^T\mathbf{X}$  for a given data matrix  $\mathbf{X}$  where matrix  $\mathbf{W}$  is non-negative. The approximation accuracy can be measured by the Frobenius matrix norm or the modified Kullback-Leibler divergence. Either dissimilarity measurement leads to multiplicative updates that learns a highly sparse factorized matrix whose columns are more orthogonal than for other variants of NMF. Our PNMf algorithm provides an efficient

solution for the nonnegative principal component analysis problem, which is in turn applicable to many practical problems such as clustering and graph partitioning. All algorithms have mathematically been proven to be iterative Lagrangian solutions, namely, jointly finds a PNMf approximation and steers the factorizing matrix towards the constraint manifold. Moreover, experiments on three real-world datasets show that PNMf is both efficient and accurate.

The PNMf method can be applied to grouping either features or samples. In the former application, the proof procedure using the Lagrangian technique may also imply a common guideline for learning a nonnegative projection by adapting the ‘‘Oja’s rule’’ [18]. The resulting subspace methods may become a new branch of blind source separation. For the grouping of sample vectors, the sparseness of PNMf can be employed to find better discriminative clusters of data [33]. For both applications, the tight connection between PNMf and PCA is worthy of further investigation for finding the potential generative model.

#### APPENDIX A MULTIPLICATIVE UPDATES

Suppose there is an algorithm which seeks an  $m$ -dimensional solution vector  $\mathbf{w}$  that maximizes an objective function  $\mathcal{J}(\mathbf{w})$ . The conventional *additive update* rule for such a problem is

$$\tilde{\mathbf{w}} = \mathbf{w} + \gamma \mathbf{g}(\mathbf{w}), \quad (64)$$

where  $\tilde{\mathbf{w}}$  is the new value of  $\mathbf{w}$ ,  $\gamma$  a positive learning rate and the function  $\mathbf{g}(\mathbf{w})$  outputs an  $m$ -dimensional vector which represents the *learning direction*, obtained e.g. from the gradient of the objective function. For notational brevity, we only discuss the learning for vectors in this section, but it is easy to generalize the results to the matrix case, where we will use capital letters  $\mathbf{W}$  in place of  $\mathbf{w}$ .

The multiplicative update technique first generalizes the common learning rate to different ones for individual dimensions:

$$\tilde{\mathbf{w}} = \mathbf{w} + \text{diag}(\boldsymbol{\eta})\mathbf{g}(\mathbf{w}), \quad (65)$$

where  $\boldsymbol{\eta}$  is an  $m$ -dimensional positive vector. Choosing different learning rates for individual dimensions changes the update direction and hence this method differs from the conventional steepest-gradient approaches in the full real-valued domain.

It has been shown that the following choice of  $\boldsymbol{\eta}$  has particularly interesting properties for the constraint of non-negativity (see e.g. [8], [34]). Suppose  $\mathbf{w}$  is non-negatively initialized. If there exists a separation of the learning direction into two positive terms  $\mathbf{g}(\mathbf{w}) = \mathbf{g}^+ - \mathbf{g}^-$  by some external knowledge, then one can always choose  $\eta_i = w_i/g_i^-, i = 1, \dots, m$ , such that the components of (65) become:

$$\tilde{w}_i = w_i + \eta_i [\mathbf{g}(\mathbf{w})]_i = w_i + \frac{w_i}{g_i^-} (g_i^+ - g_i^-) = w_i \frac{g_i^+}{g_i^-}. \quad (66)$$

The above *multiplicative update* maintains the non-negativity of  $\mathbf{w}$ . In addition,  $w_i$  increases when  $g_i^+ > g_i^-$ , i.e.  $[\mathbf{g}(\mathbf{w})]_i > 0$ , and decreases if  $[\mathbf{g}(\mathbf{w})]_i < 0$ . Thus the multiplicative change of  $w_i$  indicates how much the direction

of that axis conforms to the learning direction. There exist two kinds of stationary points in the iterative use of the multiplicative update rule (66): one satisfies  $g_i^+ = g_i^-$ , i.e.  $\mathbf{g}(\mathbf{w}) = \mathbf{0}$ , which is the same condition for local optima as in the additive updates (64), and the other one is  $w_i \rightarrow 0$ . The latter condition distinguishes the non-negative optimization from conventional ones and often yields sparseness in  $\mathbf{w}$ , which is desired in many applications. Furthermore, unlike steepest gradient or exponential gradient [35], the multiplicative update rule (66) does not require any user-specified learning rates, which facilitates its application.

As an example, assume that  $\mathbf{X}$  is an  $m \times n$  non-negative data matrix, and consider the adaptive PCA learning rule (‘‘Oja rule’’) [18] for computing the dominant eigenvector of  $\mathbf{X}\mathbf{X}^T$ :

$$\mathbf{w}' = \mathbf{w} + \gamma (\mathbf{X}\mathbf{X}^T\mathbf{w} - \mathbf{w}\mathbf{w}^T\mathbf{X}\mathbf{X}^T\mathbf{w}), \quad (67)$$

or its generalization to finding an  $m \times r$ -dimensional PCA basis matrix  $\mathbf{W}$  [36]

$$\mathbf{W}' = \mathbf{W} + \gamma (\mathbf{X}\mathbf{X}^T\mathbf{W} - \mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}), \quad (68)$$

where  $\gamma$  is a small positive learning rate. Assuming  $\mathbf{X}$  and  $\mathbf{W}$  nonnegative, a multiplicative rule is

$$W'_{ik} = W_{ik} \frac{(\mathbf{X}\mathbf{X}^T\mathbf{W})_{ik}}{(\mathbf{W}\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W})_{ik}}. \quad (69)$$

The above formulation principle of multiplicative updates helps us easily obtain an iterative algorithm, the convergence of which is however not guaranteed. In [11], the authors interpret the multiplicative updates as a special case of natural gradient learning. Such learning may albeit diverge due to the unitary learning step. In this paper we conform to the auxiliary function approach which is commonly accepted in convergence analysis.

#### APPENDIX B AUXILIARY FUNCTION

The *auxiliary function* method has widely been used for convergence analysis of optimization algorithms such as the nonnegative multiplicative updates and Expectation-Maximization (EM). Given an objection function  $\mathcal{J}(\mathbf{W})$  to be minimized,  $G(\mathbf{W}, \mathbf{W}')$  is called an auxiliary function if it is a tight upper bound of  $\mathcal{J}(\mathbf{W})$ , i.e.

$$G(\mathbf{W}, \mathbf{W}') \geq \mathcal{J}(\mathbf{W}), \quad G(\mathbf{W}, \mathbf{W}) = \mathcal{J}(\mathbf{W})$$

for any  $\mathbf{W}$  and  $\mathbf{W}'$ . Define

$$\mathbf{W}' = \arg \min_{\tilde{\mathbf{W}}} G(\tilde{\mathbf{W}}, \mathbf{W}). \quad (70)$$

By construction,

$$\mathcal{J}(\mathbf{W}) = G(\mathbf{W}, \mathbf{W}) \geq G(\mathbf{W}', \mathbf{W}) \geq G(\mathbf{W}', \mathbf{W}') = \mathcal{J}(\mathbf{W}'),$$

where the left inequality is the result of minimization and the right one comes from the upper bound. Iteratively applying the update rule (70) thus results in a monotonically decreasing sequence of  $\mathcal{J}$ . Besides the tight upper bound, it is often desired that the minimization (70) has a closed-form solution. In particular, setting  $\partial G/\partial \mathbf{W}' = 0$  should lead to the iterative update rule in analysis.

APPENDIX C  
UPPER BOUNDING POSITIVE TERMS

The objective function involves a number of terms which can be divided into two groups according to their leading signs. Finding the auxiliary function can in turn become an upper bound for each positive term and a lower bound for each unsigned negative term. For the former, we employ three existing approaches in this paper, two for trace-like terms and the other for cross-entropy-like terms.

*Proposition 5 (quadratic upper bound):* For any matrices  $\mathbf{A} \in \mathbb{R}_+^{r \times r}$ ,  $\mathbf{A}$  symmetric,  $\mathbf{W} \in \mathbb{R}_+^{m \times r}$ , and  $\mathbf{W}' \in \mathbb{R}_+^{m \times r}$ , it holds

$$\sum_{ik} \frac{(\mathbf{W}\mathbf{A})_{ik} W'_{ik}}{W_{ik}} \geq \text{Tr}(\mathbf{W}'^T \mathbf{W}' \mathbf{A}). \quad (71)$$

*Proposition 6 (linear upper bound):* For any matrices  $\mathbf{A} \in \mathbb{R}_+^{m \times r}$ ,  $\mathbf{W} \in \mathbb{R}_+^{m \times r}$ , and  $\mathbf{W}' \in \mathbb{R}_+^{m \times r}$ , we have

$$\sum_{ik} A_{ik} \frac{W'_{ik} + W_{ik}}{2W_{ik}} \geq \text{Tr}(\mathbf{A}^T \mathbf{W}'). \quad (72)$$

The proofs can be found in [23], [10] and [19]. Minimizing such upper bounds has a closed-form solution because, for example, the derivative of the left-hand side of (72) is

$$\frac{W'_{ik}}{W_{ik}} A_{ik}. \quad (73)$$

Combining other gradient terms, for instance  $-B_{ik}$ , this leads to a multiplicative update rule

$$\frac{W'_{ik}}{W_{ik}} A_{ik} - B_{ik} = 0 \Rightarrow W'_{ik} = W_{ik} \frac{B_{ik}}{A_{ik}}.$$

Similar property holds for (71).

*Proposition 7 (Jensen upper bound):* For any matrices  $\mathbf{X} \in \mathbb{R}_+^{m \times n}$ ,  $\mathbf{W} \in \mathbb{R}_+^{m \times r}$ ,  $\mathbf{W}' \in \mathbb{R}_+^{m \times r}$ , and  $\mathbf{H} \in \mathbb{R}_+^{r \times n}$

$$\begin{aligned} & - \sum_{ij} X_{ij} \log(\mathbf{W}'\mathbf{H})_{ij} \leq \\ & - \sum_{ij} X_{ij} \sum_k \alpha_{ijk} (\log W'_{ik} H_{kj} - \log \alpha_{ijk}) \end{aligned} \quad (74)$$

where

$$\alpha_{ijk} = \frac{W_{ik} H_{kj}}{\sum_l W_{il} H_{lj}}.$$

The proof follows from Jensen's inequality [23]. Minimizing such an upper bound requires the derivative of the right-hand side of (74)

$$\frac{W_{ik}}{W'_{ik}} (\mathbf{Z}\mathbf{X}^T \mathbf{W})_{ik}, \quad (75)$$

where  $Z_{ij} = X_{ij}/(\mathbf{W}\mathbf{H})_{ij}$ . Combining some other gradient terms, for instance  $-B_{ik}$ , the resulting multiplicative update rule becomes

$$W'_{ik} = W_{ik} \frac{(\mathbf{Z}\mathbf{X}^T \mathbf{W})_{ik}}{B_{ik}}.$$

APPENDIX D  
UPPER BOUNDING NEGATIVE TERMS

Upper bounding positive terms is sufficient to produce a multiplicative update rule. Alternatively, one can also lower bound the unsigned negative terms to obtain a different multiplicative update rule. It was reported that the latter approach has better performance when the input matrix contains negative entries [37]. The lower bound stems from the inequality

$$z \geq 1 + \log z,$$

for  $z \geq 0$ , where the equality holds if and only if  $z = 1$ .

*Proposition 8 (linear lower bound):* For  $\mathbf{B} \in \mathbb{R}_+^{m \times r}$ ,  $\mathbf{W} \in \mathbb{R}_+^{m \times r}$ , and  $\mathbf{W}' \in \mathbb{R}_+^{m \times r}$ ,

$$\text{Tr}(\mathbf{B}^T \mathbf{W}') \geq \sum_{ik} B_{ik} W_{ik} \left( 1 + \log \frac{W'_{ik}}{W_{ik}} \right). \quad (76)$$

*Proposition 9 (quadratic lower bound):* For  $\mathbf{B} \in \mathbb{R}_+^{r \times r}$ ,  $\mathbf{W} \in \mathbb{R}_+^{m \times r}$ , and  $\mathbf{W}' \in \mathbb{R}_+^{m \times r}$ ,

$$\text{Tr}(\mathbf{W}'^T \mathbf{W}' \mathbf{B}) \geq \sum_{ikl} B_{kl} W_{ik} W_{il} \left( 1 + \log \frac{W'_{ik} W'_{il}}{W_{ik} W_{il}} \right).$$

The negative derivative of the right-hand side of (76) is

$$\frac{W_{ik}}{W'_{ik}} B_{ik}, \quad (77)$$

which is also ready to generate multiplicative update rules. For example, combining the (77) and (73) leads to a multiplicative update rule:

$$\frac{W'_{ik}}{W_{ik}} A_{ik} - \frac{W_{ik}}{W'_{ik}} B_{ik} = 0 \Rightarrow W'_{ik} = W_{ik} \sqrt{\frac{B_{ik}}{A_{ik}}}.$$

APPENDIX E  
THE *moving term* TECHNIQUE

It is desired that all terms in multiplicative update rules are nonnegative. However, sometimes negative terms may appear in the numerator or denominator when setting the gradient of an auxiliary function to zero. According to the principle of formulating multiplicative update rules in Appendix A, one should neglect the sign of such terms and move them from the numerator to denominator or vice versa. This can be implemented by adding the same term to both numerator and denominator and justified as a corollary of the lower bounding technique in Appendix C-D.

*Proposition 10 (moving term upper bound, type I):*

$$\begin{aligned} F_1(\mathbf{A}, \mathbf{W}, \mathbf{W}') = \\ \sum_{ik} A_{ik} W'_{ik} - \sum_{ik} A_{ik} W_{ik} - \sum_{ik} A_{ik} W_{ik} \log \frac{W'_{ik}}{W_{ik}} \geq 0. \end{aligned}$$

The proof can be obtained by writing

$$F_1(\mathbf{A}, \mathbf{W}, \mathbf{W}') = \sum_{ik} A_{ik} \left( W'_{ik} - W_{ik} - W_{ik} \log \frac{W'_{ik}}{W_{ik}} \right).$$

The sum in parentheses is nonnegative according to  $z \geq 1 + \log z$  for  $z \geq 0$ . In addition, the function  $F$  vanishes if  $\mathbf{W} = \mathbf{W}'$ . Thus one can add  $F(\mathbf{A}, \mathbf{W}, \mathbf{W}')$  to the original

auxiliary function without violating the tight bound constraint. Furthermore,

$$\frac{\partial F_1}{\partial W'_{ik}} = A_{ik} - \frac{W_{ik}}{W'_{ik}} A_{ik},$$

Combined with the other terms from e.g. Jensen upper bound, linear or quadratic lower bound that lead to a multiplicative update rule, the above derivative will add  $A_{ik}$  to both the numerator and denominator.

Likewise, according to the *linear upper bound*, one can alternatively move  $A_{ik}$  by

*Proposition 11 (moving term upper bound, type II):*

$$F_2(\mathbf{A}, \mathbf{W}, \mathbf{W}') = \sum_{ik} A_{ik} \left( -W'_{ik} + \frac{W_{ik}^2 + W_{ik}^2}{2W_{ik}} \right) \geq 0.$$

such that

$$\frac{\partial F_2}{\partial W'_{ik}} = -A_{ik} + \frac{W_{ik}}{W_{ik}} A_{ik},$$

which is consistent with the form of linear and quadratic upper bounds.

#### APPENDIX F PROOF OF THEOREM 3

The generalized objective is

$$\tilde{\mathcal{J}}_F^\perp(\mathbf{W}, \mathbf{H}) = \tilde{\mathcal{J}}_F(\mathbf{W}, \mathbf{H}) + \text{Tr}(\mathbf{\Lambda}(\mathbf{W}^T \mathbf{W} - \mathbf{I})),$$

where  $\tilde{\mathcal{J}}_F(\mathbf{W}, \mathbf{H})$  is given in (8) and  $\{\Lambda_{kl}\}$  are the introduced Lagrangian multipliers. Similar to  $G_F$  in (9), we construct

$$G_F^\perp(\mathbf{W}, \mathbf{W}') \equiv \text{Tr} \left( -2\mathbf{X}^T \mathbf{W}' \mathbf{H} - \mathbf{\Psi}^T \mathbf{W}'^T \mathbf{X} \right) \quad (78)$$

$$+ \sum_{ik} \frac{(\mathbf{W} \mathbf{H} \mathbf{H}^T)_{ik} W_{ik}^2}{W_{ik}} \quad (79)$$

$$+ \sum_{ik} \frac{(\mathbf{W} \mathbf{\Lambda})_{ik} W_{ik}^2}{W_{ik}} \quad (80)$$

$$+ \text{Tr}(\mathbf{X}^T \mathbf{X} + \mathbf{\Psi}^T \mathbf{H} - \mathbf{\Lambda}) \quad (81)$$

as an auxiliary function of

$$\mathcal{L}_F^\perp(\mathbf{W}') \equiv \tilde{\mathcal{J}}_F^\perp(\mathbf{W}', \mathbf{H}) \quad (82)$$

$$= \text{Tr} \left( -2\mathbf{X}^T \mathbf{W}' \mathbf{H} - \mathbf{\Psi}^T \mathbf{W}'^T \mathbf{X} \right) \quad (83)$$

$$+ \text{Tr}(\mathbf{W}'^T \mathbf{W}' \mathbf{H} \mathbf{H}^T) \quad (84)$$

$$+ \text{Tr}(\mathbf{\Lambda} \mathbf{W}'^T \mathbf{W}') \quad (85)$$

$$+ \text{Tr}(\mathbf{X}^T \mathbf{X} + \mathbf{\Psi}^T \mathbf{H} - \mathbf{\Lambda}). \quad (86)$$

Here we apply the *quadratic upper bound* (79) to (84) and (80) to (85) according to Appendix C. Setting

$$\partial G_F^\perp(\mathbf{W}, \mathbf{W}') / \partial W'_{ik} = 0,$$

we get

$$W'_{ik} = W_{ik} \frac{\left( 2\mathbf{X} \mathbf{H}^T + \mathbf{X} \mathbf{\Psi}^T \right)_{ik}}{\left( 2\mathbf{W} \mathbf{H} \mathbf{H}^T + 2\mathbf{W} \mathbf{\Lambda} \right)_{ik}}. \quad (87)$$

Next we solve  $\mathbf{\Lambda}$  and  $\mathbf{\Psi}$  by using the K.K.T. conditions.

$$\frac{\partial \tilde{\mathcal{J}}_F^\perp(\mathbf{W}, \mathbf{H})}{\partial \mathbf{H}} = -2\mathbf{W}^T \mathbf{X} + 2\mathbf{W}^T \mathbf{W} \mathbf{H} + \mathbf{\Psi} = \mathbf{0},$$

one obtains

$$\mathbf{\Psi} = 2\mathbf{W}^T \mathbf{X} - 2\mathbf{W}^T \mathbf{W} \mathbf{H}.$$

By inserting  $\mathbf{H} = \mathbf{W}^T \mathbf{X}$  and  $\mathbf{W}^T \mathbf{W} = \mathbf{I}$ , we find  $\mathbf{\Psi} = \mathbf{0}$ . Using this result and from

$$\frac{\partial \tilde{\mathcal{J}}_F^\perp(\mathbf{W}, \mathbf{H})}{\partial \mathbf{W}} = -2\mathbf{X} \mathbf{H}^T - \mathbf{X} \mathbf{\Psi} + 2\mathbf{W} \mathbf{H} \mathbf{H}^T + 2\mathbf{W} \mathbf{\Lambda} = \mathbf{0},$$

we get

$$\mathbf{W} \mathbf{\Lambda} = \mathbf{X} \mathbf{H}^T - \mathbf{W} \mathbf{H} \mathbf{H}^T.$$

Left multiplying  $\mathbf{W}^T$  in both sides and using  $\mathbf{W}^T \mathbf{W} = \mathbf{I}$ , one obtains

$$\mathbf{\Lambda} = \mathbf{W}^T \mathbf{X} \mathbf{H}^T - \mathbf{H} \mathbf{H}^T.$$

Inserting  $\mathbf{H} = \mathbf{W}^T \mathbf{X}$ , we find  $\mathbf{\Lambda} = \mathbf{0}$ . Substituting  $\mathbf{\Psi} = \mathbf{0}$  and  $\mathbf{\Lambda} = \mathbf{0}$  back to (87), the multiplicative update rule becomes (41).  $\square$

#### APPENDIX G PROOF OF THEOREM 4

The generalized Lagrangian objective is

$$\tilde{\mathcal{J}}_D^\perp(\mathbf{W}, \mathbf{H}) = \tilde{\mathcal{J}}_D(\mathbf{W}, \mathbf{H}) + \text{Tr}(\mathbf{\Lambda}(\mathbf{I} - \mathbf{W}^T \mathbf{W})),$$

and is tightly upper bounded by

$$G_D^\perp(\mathbf{W}, \mathbf{W}') = G_D(\mathbf{W}, \mathbf{W}')$$

$$- \sum_{ikl} \Lambda_{kl} W_{ik} W_{il} \left( 1 + \log \frac{W'_{ik} W'_{il}}{W_{ik} W_{il}} \right),$$

where we apply the *quadratic lower bound* (see Appendix D) for the additional term and replace  $A_{ik}$  in (24) with

$$\mathbf{A}^\perp = \mathbf{X} \mathbf{Z}^T \mathbf{W} + \mathbf{W} \mathbf{W}^T \mathbf{X} \mathbf{Z}^T \mathbf{W} + \mathbf{W} \mathbf{W}^T \mathbf{Z} \mathbf{X}^T \mathbf{W}$$

Setting  $\partial G_D^\perp(\mathbf{W}, \mathbf{W}') / \partial W'_{ik} = 0$  yields

$$W'_{ik} = W_{ik} \frac{(\mathbf{Z} \mathbf{X}^T \mathbf{W})_{ik} + (\mathbf{W} \mathbf{\Lambda})_{ik} + A_{ik}^\perp}{\sum_j H_{kj} - (\mathbf{X} \mathbf{\Psi}^T)_{ik} + A_{ik}^\perp}. \quad (88)$$

The quantities  $\mathbf{\Psi}$  and  $\mathbf{\Lambda}$  can be solved by using the K.K.T. conditions:

$$\Psi_{jk} = (\mathbf{W}^T \mathbf{Z})_{ik} - \sum_b W_{bk} \quad (89)$$

$$\Lambda_{kl} = -(\mathbf{W}^T \mathbf{Z} \mathbf{X}^T \mathbf{W})_{kl} - (\mathbf{W}^T \mathbf{X} \mathbf{Z} \mathbf{W})_{kl} + \sum_i W_{ik} \sum_j H_{lj} + \sum_j (\mathbf{W}^T \mathbf{X})_{kj} \sum_i W_{il}. \quad (90)$$

Substituting (89) and (90) into (88), we get (45).  $\square$

## APPENDIX H

## DERIVATION OF UPDATE RULE OF SEMI-NKPCA

The generalized Lagrangian function is

$$\begin{aligned}\tilde{\mathcal{J}}_K^\pm(\mathbf{U}, \mathbf{V}) &= \mathcal{J}_K^\pm(\mathbf{U}, \mathbf{V}) + \text{Tr}(\mathbf{\Lambda}(\mathbf{U}^T\mathbf{U} - \mathbf{I})) \\ &\quad + \text{Tr}(\mathbf{\Psi}(\mathbf{V} - \mathbf{U})) \\ &= \text{Tr}(2\mathbf{V}^T\mathbf{K}^-\mathbf{U}) \\ &\quad + \text{Tr}(\mathbf{V}^T\mathbf{K}^+\mathbf{V}\mathbf{U}^T\mathbf{U}) \\ &\quad - \text{Tr}(2\mathbf{V}^T\mathbf{K}^+\mathbf{U}) \\ &\quad - \text{Tr}(\mathbf{V}^T\mathbf{K}^-\mathbf{V}\mathbf{U}^T\mathbf{U}) \\ &\quad + \text{Tr}(\mathbf{\Lambda}\mathbf{U}^T\mathbf{U}) \\ &\quad - \text{Tr}(\mathbf{\Psi}^T\mathbf{U}) \\ &\quad + \text{Tr}(\mathbf{K} - \mathbf{I} + \mathbf{\Psi}^T\mathbf{V})\end{aligned}$$

The function  $\mathcal{L}_K^\pm(\mathbf{U}') = \tilde{\mathcal{J}}_K^\pm(\mathbf{U}', \mathbf{V})$  has the auxiliary function

$$\begin{aligned}G_K^\pm(\mathbf{U}, \mathbf{U}') &= \sum_{jk} (\mathbf{K}^-\mathbf{V})_{jk} \frac{U'_{jk}{}^2 + U_{jk}^2}{U_{jk}} \\ &\quad + \sum_{jk} (\mathbf{U}\mathbf{V}^T\mathbf{K}^+\mathbf{V})_{jk} \frac{U'_{jk}{}^2}{U_{jk}} \\ &\quad - \sum_{jk} (2\mathbf{K}^+\mathbf{V})_{jk} U_{jk} \log\left(1 + \frac{U'_{jk}}{U_{jk}}\right) \\ &\quad - \sum_{jkl} (\mathbf{V}^T\mathbf{K}^-\mathbf{V})_{kl} U_{jk} U_{jl} \log\left(1 + \frac{U'_{jk} U'_{jl}}{U_{jk} U_{jl}}\right) \\ &\quad + \sum_{jk} \frac{(\mathbf{U}\mathbf{\Lambda})_{jk} U_{jk}{}^2}{U_{jk}} \\ &\quad - \sum_{jk} \Psi_{jk} U'_{jk} \log\left(1 + \frac{U'_{jk}}{U_{jk}}\right) \\ &\quad + \text{Tr}(\mathbf{K} - \mathbf{I} + \mathbf{\Psi}^T\mathbf{V})\end{aligned}$$

Setting  $\partial G_K^\pm(\mathbf{U}, \mathbf{U}') / \partial U'_{jk} = 0$ , we get

$$U'_{jk} = U_{jk} \sqrt{\frac{(2\mathbf{K}^+\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{K}^-\mathbf{V} + \mathbf{\Psi})_{jk}}{(2\mathbf{K}^-\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{K}^+\mathbf{V} + \mathbf{U}\mathbf{\Lambda})_{jk}}}$$

The quantities  $\mathbf{\Psi}$  and  $\mathbf{\Lambda}$  can be solved by the K.K.T. conditions, which results in  $\mathbf{\Psi} = \mathbf{0}$  and  $\mathbf{\Lambda} = \mathbf{0}$ . With  $\mathbf{V} = \mathbf{U}$ , the multiplicative update rule for the problem (49)-(51) is

$$U'_{jk} = U_{jk} \sqrt{\frac{(\mathbf{K}^+\mathbf{U} + \mathbf{U}\mathbf{U}^T\mathbf{K}^-\mathbf{U})_{jk}}{(\mathbf{K}^-\mathbf{U} + \mathbf{U}\mathbf{U}^T\mathbf{K}^+\mathbf{U})_{jk}}}$$

Alternatively, one can construct an auxiliary function without upper bounding the negative terms, which leads to the following multiplicative update rule:

$$U'_{jk} = U_{jk} \frac{(\mathbf{K}^+\mathbf{U} + \mathbf{U}\mathbf{U}^T\mathbf{K}^-\mathbf{U})_{jk}}{(\mathbf{K}^-\mathbf{U} + \mathbf{U}\mathbf{U}^T\mathbf{K}^+\mathbf{U})_{jk}}.$$

## REFERENCES

- [1] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 4164–4169, 2004.
- [2] M. Cooper and J. Foote, "Summarizing video using non-negative similarity matrix factorization," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, 2002, pp. 25–28.
- [3] T. Feng, S. Z. Li, H. Y. Shum, and H. J. Zhang, "Local non-negative matrix factorization as a visual representation," in *Proceedings. The 2nd International Conference on Development and Learning*, 2002, pp. 178–183.
- [4] W. Liu, N. Zheng, and X. Lu, "Non-negative matrix factorization for visual coding," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, vol. 3, 2003, pp. 293–296.
- [5] B. W. Xu, J. J. Lu, and G. S. Huang, "A constrained non-negative matrix factorization in information retrieval," in *Proceedings of The 2003 IEEE International Conference on Information Reuse and Integration (IRI2003)*, 2003, pp. 273–277.
- [6] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 2003, pp. 267–273.
- [7] A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis*. John Wiley, 2009.
- [8] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.
- [9] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [10] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 126–135.
- [11] Z. Yang and J. Laaksonen, "Multiplicative updates for non-negative projections," *Neurocomputing*, vol. 71, no. 1-3, pp. 363–373, 2007.
- [12] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of SIAM International Conference of Data Mining*, 2005, pp. 606–610.
- [13] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Proceedings of IEEE International Conference on Data Mining*, 2006, pp. 362–371.
- [14] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proceedings of International Conference on Machine Learning*, July 2004, pp. 225–232.
- [15] Z. Yuan and E. Oja, "Projective nonnegative matrix factorization for image compression and feature extraction," in *Proc. of 14th Scandinavian Conference on Image Analysis (SCIA 2005)*, Joensuu, Finland, June 2005, pp. 333–342.
- [16] Z. Yang, Z. Yuan, and J. Laaksonen, "Projective non-negative matrix factorization with applications to facial image processing," *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 21, no. 8, pp. 1353–1362, Dec. 2007.
- [17] Z. Yuan, Z. Yang, and E. Oja, "Projective nonnegative matrix factorization: Sparseness, orthogonality, and clustering," 2009, in submission.
- [18] E. Oja, "A simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology*, vol. 15, pp. 267–273, 1982.
- [19] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
- [20] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 313–319.
- [21] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks*, vol. 8, no. 4, pp. 549–562, 1995.
- [22] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.
- [23] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

- [24] A. D. Stefan Wild, James Curry, "Improving non-negative matrix factorizations through structured initialization," *Pattern Recognition*, vol. 37, no. 11, pp. 2217–2232, 2004.
- [25] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev.*, vol. 74, no. 036104, 2006.
- [26] S. Lloyd, "Last square quantization in pcm," *IEEE Transactions on Information Theory, special issue on quantization*, vol. 28, pp. 129–137, 1982.
- [27] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.
- [28] C.-J. Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, pp. 2756–2779, 2007.
- [29] M. Fiedler, "Algebraic connectivity of graphs," *Czech. Math.*, vol. 23, pp. 298–305, 1973.
- [30] A. Pothen, H. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM Journal on Matrix Analysis and Applications*, vol. 11, pp. 430–452, 1990.
- [31] I. Dhillon, Y. Guan, and B. Kulis, "Kernel kmeans, spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, Seattle, WA, USA, 2004, pp. 551–556.
- [32] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.
- [33] S. Kaski, J. Sinkkonen, and A. Klami, "Discriminative clustering," *Neurocomputing*, vol. 69, pp. 18–41, 2005.
- [34] F. Sha, L. K. Saul, and D. D. Lee, "Multiplicative updates for large margin classifiers," in *COLT*, 2003, pp. 188–202.
- [35] J. Kivinen and M. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Information and Computation*, vol. 132, no. 1, pp. 1–63, 1997.
- [36] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, pp. 927–935, 1992.
- [37] F. Sha, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming in support vector machines," in *NIPS 2002*, 2002, pp. 1065–1072.



**Erkki Oja** Erkki Oja (S'75-M'78-SM'90-F'00) received the D.Sc. degree from Helsinki University of Technology in 1977. He is Director of the Adaptive Informatics Research Centre and Professor of Computer Science at the Laboratory of Computer and Information Science, Aalto University (former Helsinki University of Technology), Finland, and the Chairman of the Finnish Research Council for Natural Sciences and Engineering. He holds an honorary doctorate from Uppsala University, Sweden. He has been research associate at Brown University,

Providence, RI, and visiting professor at the Tokyo Institute of Technology, Japan. He is the author or coauthor of more than 280 articles and book chapters on pattern recognition, computer vision, and neural computing, and three books: "Subspace Methods of Pattern Recognition" (New York: Research Studies Press and Wiley, 1983), which has been translated into Chinese and Japanese; "Kohonen Maps" (Amsterdam, The Netherlands: Elsevier, 1999), and "Independent Component Analysis" (New York: Wiley, 2001; also translated into Chinese and Japanese). His research interests are in the study of principal component and independent component analysis, self-organization, statistical pattern recognition, and applying artificial neural networks to computer vision and signal processing.

Prof. Oja is a member of the Finnish Academy of Sciences, Founding Fellow of the International Association of Pattern Recognition (IAPR), Past President of the European Neural Network Society (ENNS), and Fellow of the International Neural Network Society (INNS). He is a member of the editorial boards of several journals and has been in the program committees of several recent conferences including the International Conference on Artificial Neural Networks (ICANN), International Joint Conference on Neural Networks (IJCNN), and Neural Information Processing Systems (NIPS). Prof. Oja is the recipient of the 2006 IEEE Computational Intelligence Society Neural Networks Pioneer Award.



**Zhirong Yang** received his Bachelor and Master degrees from Sun Yat-Sen University, China in 1999 and 2002, and Doctoral degree from Helsinki University of Technology, Finland in 2008, respectively. Presently he is a postdoctoral researcher with the Department of Information and Computer Science at Aalto University School of Science and Technology. His current research interests include machine learning, pattern recognition, computer vision, and information retrieval, particularly focus on nonnegative learning, information visualization, optimization,

discriminative feature extraction and visual recognition. He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the International Neural Network Society (INNS).