Convex and Semi-Nonnegative Matrix Factorizations

Chris Ding, Tao Li, and Michael I. Jordan

Chris Ding is with the Department of Computer Science and Engineering, University of Texas, Arlington, TX 76019. Tao Li is with the School of Computer Science at Florida International University, Miami, FL 33199, USA. Michael I. Jordan is with the Department of Electrical Engineering and Computer Science and the Department of Statistics at the University of California at Berkeley, CA 94720, USA.

Abstract

We present several new variations on the theme of nonnegative matrix factorization (NMF). Considering factorizations of the form $X = FG^T$, we focus on algorithms in which G is restricted to contain nonnegative entries, but allow the data matrix X to have mixed signs, thus extending the applicable range of NMF methods. We also consider algorithms in which the basis vectors of F are constrained to be convex combinations of the data points. This is used for a kernel extension of NMF. We provide algorithms for computing these new factorizations and we provide supporting theoretical analysis. We also analyze the relationships between our algorithms and clustering algorithms, and consider the implications for sparseness of solutions. Finally, we present experimental results that explore the properties of these new methods.

Index Terms

Nonnegative Matrix Factorization, Singular Value Decomposition, Clustering

I. Introduction

Matrix factorization is a unifying theme in numerical linear algebra. A wide variety of matrix factorization algorithms have been developed over many decades, providing a numerical platform for matrix operations such as solving linear systems, spectral decomposition, and subspace identification. Some of these algorithms have also proven useful in statistical data analysis, most notably the singular value decomposition (SVD), which underlies principal component analysis (PCA).

Recent work in machine learning has focused on matrix factorizations that directly target some of the special features of statistical data analysis. In particular, nonnegative matrix factorization (NMF) (1; 2) focuses on the analysis of data matrices whose elements are nonnegative, a common occurrence in data sets derived from text and images. Moreover, NMF yields nonnegative factors, which can be advantageous from the point of view of interpretability.

The scope of research on NMF has grown rapidly in recent years. NMF has been shown to be useful in a variety of applied settings, including environmetrics (3), chemometrics (4), pattern recognition (5), multimedia data analysis (6), text mining (7; 8), DNA gene expression analysis (9; 10) and protein interaction (11). Algorithmic extensions of NMF have been developed to accommodate a variety of objective functions (12; 13) and a variety of data analysis problems, including classification (14) and collaborative filtering (15). A number of studies have focused on further developing computational methodologies for NMF (16; 17; 18; 19). Finally, researchers have begun to explore some of the relationships between matrix factorizations and K-means clustering (20), making use of the least square objectives of NMF; as we emphasize in the current paper, this relationship has implications for the interpretability of matrix factors. NMF with the Kullback-Leibler (KL) divergence objective has been shown (21; 13) to be equivalent to probabilistic latent semantic analysis (22) which has been further developed into the fully-probabilistic latent Dirichlet allocation model (23; 24).

Our goal in this paper is to expand the repertoire of nonnegative matrix factorization. Our focus is on algorithms that constrain the matrix factors; we do not require the data matrix to be similarly constrained. In particular, we develop NMF-like algorithms that yield nonnegative factors but do not require the data matrix to be nonnegative. This extends the range of application of NMF ideas. Moreover, by focusing on constraints on the matrix factors, we are able to strengthen the connections between NMF and K-means clustering. Note in particular that the result of a K-means clustering run can be written as a matrix factorization $X = FG^T$, where X is the data matrix, F contains the cluster centroids, and G contains the cluster membership indicators. Although F typically has entries with both positive and negative signs, G is nonnegative. This motivates us to propose general factorizations in which G is restricted to be nonnegative and F is unconstrained. We also consider algorithms that constrain F; in particular, restricting the columns of F to be convex combinations of data points in X we obtain a matrix factorization that can be interpreted in terms of weighted cluster centroids.

The paper is organized as follows. In Section II we present the new matrix factorizations and in Section III we present algorithms for computing these factorizations. Section IV provides a theoretical analysis which provides insights into the sparseness of matrix factors for a convex variant of NMF. In Section V we consider extensions of Convex-NMF and the relationships of NMF-like factorizations. In Section V-A we show that a convex variant of NMF has the advantage that it is readily kernelized. In Section VI we present comparative experiments that show that constraining the F factors to be convex combinations of input data enhances their interpretability. We also present experiments that compare the performance of the NMF variants to K-means clustering, where we assess the extent to which the imposition of constraints that aim to enhance interpretability leads to poorer clustering performance. Finally, we present our conclusions in Section VII.

II. Semi-NMF and Convex-NMF

Let the input data matrix $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ contain a collection of n data vectors as columns. We consider factorizations of the form:

$$X \approx FG^T, \tag{1}$$

where $X \in \mathbb{R}^{p\times n},\ F \in \mathbb{R}^{p\times k}$ and $G \in \mathbb{R}^{n\times k}$. For example, the SVD can be written in this form. In the case of the SVD, there are no restrictions on the signs of F and G; moreover, the data matrix X is also unconstrained. NMF can also be written in this form, where the data matrix X is assumed to be nonnegative, as are the factors F and G. We now consider some additional examples.

A. Semi-NMF

When the data matrix is unconstrained (i.e., it may have mixed signs), we consider a factorization that we refer to as *Semi-NMF*, in which we restrict G to be nonnegative while placing no restriction on the signs of F.

We can motivate Semi-NMF from the perspective of clustering. Suppose we do a K-means clustering on X and obtain cluster centroids $F = (\mathbf{f}_1, \ldots, \mathbf{f}_k)$. Let G denote the cluster indicators: i.e., $g_{ik} = 1$ if $\mathbf{x}_i$ belongs to cluster c_k; $g_{ik} = 0$ otherwise. We can write the K-means clustering objective function as

$$J_{K\text{-means}} = \sum_{i=1}^{n}\sum_{k=1}^{K} g_{ik}||\mathbf{x}_i - \mathbf{f}_k||^2 = ||X - FG^T||^2.$$

In this paper, $||\mathbf{v}||$ denotes the L_2 norm of a vector $\mathbf{v}$ and $||A||$ denotes the Frobenius norm of a matrix A. We see that the K-means clustering objective can be alternatively viewed as an objective function for matrix approximation. Moreover, this approximation will generally be tighter if we relax the optimization by allowing g_{ij} to range over values in $(0, 1)$, or values in $(0, \infty)$. This yields the Semi-NMF matrix factorization.

B. Convex-NMF

While in NMF and Semi-NMF there are no constraints on the basis vectors $F = (\mathbf{f}_1, \cdots, \mathbf{f}_k)$, for reasons of interpretability it may be useful to impose the constraint that the vectors defining F lie within the column space of X:

$$\mathbf{f}_\ell = w_{1\ell}\mathbf{x}_1 + \cdots + w_{n\ell}\mathbf{x}_n = X\mathbf{w}_\ell, \quad \text{or} \quad F = XW. \tag{2}$$

Moreover, again for reasons of interpretability, we may wish to restrict ourselves to convex combinations of the columns of X. This constraint has the advantage that we could interpret the columns $\mathbf{f}_\ell$ as weighted sums of certain data points; in particular, these columns would capture a notion of *centroids*. We refer to this restricted form of the F factor as *Convex-NMF*. Convex-NMF applies to both nonnegative and mixed-sign data matrices. As we will see, Convex-NMF has an interesting property: the factors W and G both tend to be very sparse.

(25) considered a model in which the F factors were restricted to the unit interval; i.e., $0 \leq F_{ik} \leq 1$. This so-called convex coding does not require the $\mathbf{f}_k$ to be nonnegative linear combinations of input data vectors and thus in general do not capture the notion of cluster centroid. Indeed, the emphasis in (25) and in (1; 2) is the parts-of-whole encoding provided by NMF, not the relationship of nonnegative factorizations to vector quantization.

To summarize our development thus far, let us write the different factorizations as follows:

$$\text{SVD:} \quad X_\pm \approx F_\pm G_\pm^T \tag{3}$$

$$\text{NMF:} \quad X_+ \approx F_+ G_+^T \tag{4}$$

$$\text{Semi-NMF:} \quad X_\pm \approx F_\pm G_+^T \tag{5}$$

$$\text{Convex-NMF:} \quad X_\pm \approx X_\pm W_+ G_+^T, \tag{6}$$

where the subscripts are intended to suggest the constraints imposed by the different factorizations.

Before turning to a presentation of algorithms for computing Semi-NMF and Convex-NMF factorizations and supporting theoretical analysis, we provide an illustrative example.

C. An Illustration

Consider the following data matrix:

$$X = \begin{pmatrix} 1.3 & 1.8 & 4.8 & 7.1 & 5.0 & 5.2 & 8.0 \\ 1.5 & 6.9 & 3.9 & -5.5 & -8.5 & -3.9 & -5.5 \\ 6.5 & 1.6 & 8.2 & -7.2 & -8.7 & -7.9 & -5.2 \\ 3.8 & 8.3 & 4.7 & 6.4 & 7.5 & 3.2 & 7.4 \\ -7.3 & -1.8 & -2.1 & 2.7 & 6.8 & 4.8 & 6.2 \end{pmatrix}.$$

The K-means clustering produces two clusters, where the first cluster includes the first three columns and the second cluster includes the last four columns.

We show that Semi-NMF and Convex-NMF factorizations gives clustering solutions which are identical to the K-means clustering results. We run SVD, Semi-NMF and Convex-NMF. The matrix factor G obtained for the three factorizations are

$$G_{\text{svd}}^T = \begin{pmatrix} 0.25 & 0.05 & 0.22 & -.45 & -.44 & -.46 & -.52 \\ 0.50 & 0.60 & 0.43 & 0.30 & -0.12 & 0.01 & 0.31 \end{pmatrix},$$

$$G_{\text{semi}}^T = \begin{pmatrix} 0.61 & 0.89 & 0.54 & 0.77 & 0.14 & 0.36 & 0.84 \\ 0.12 & 0.53 & 0.11 & 1.03 & 0.60 & 0.77 & 1.16 \end{pmatrix},$$

$$G_{\text{cnvx}}^T = \begin{pmatrix} 0.31 & 0.31 & 0.29 & 0.02 & 0 & 0 & 0.02 \\ 0 & 0.06 & 0 & 0.31 & 0.27 & 0.30 & 0.36 \end{pmatrix}.$$

Both the Semi-NMF and Convex-NMF results agree with the K-means clustering: for the first three columns, the values in the upper rows are larger than the lower rows, indicating they are in the same cluster, while for the last four columns the upper rows are smaller than the lower rows, indicating they are in another cluster. Note, however, that Convex-NMF gives sharper indicators of the clustering.

The computed basis vectors F for the different matrix factorizations are as follows:

$$F_{\text{svd}} = \begin{pmatrix} -0.41 & 0.50 \\ 0.35 & 0.21 \\ 0.66 & 0.32 \\ -0.28 & 0.72 \\ -0.43 & -0.28 \end{pmatrix}, \; F_{\text{semi}} = \begin{pmatrix} 0.05 & 0.27 \\ 0.40 & -0.40 \\ 0.70 & -0.72 \\ 0.30 & 0.08 \\ -0.51 & 0.49 \end{pmatrix}, \; F_{\text{cnvx}} = \begin{pmatrix} 0.31 & 0.53 \\ 0.42 & -0.30 \\ 0.56 & -0.57 \\ 0.49 & 0.41 \\ -0.41 & 0.36 \end{pmatrix},$$

and the cluster centroids obtained from K-means clustering are given by the columns of the following matrix:

$$C_{\text{Kmeans}} = \begin{pmatrix} 0.29 & 0.52 \\ 0.45 & -0.32 \\ 0.59 & -0.60 \\ 0.46 & 0.36 \\ -0.41 & 0.37 \end{pmatrix}.$$

We have rescaled all column vectors so that their L_2-norm is one for purposes of comparison.

One can see that F_{cnvx} is close to C_{Kmeans}: $||F_{\text{cnvx}} - C_{\text{Kmeans}}|| = 0.08$. F_{semi} deviates substantially from C_{Kmeans}: $||F_{\text{semi}} - C_{\text{Kmeans}}|| = 0.53$. Two of the elements in F_{semi} are particularly far from those in C_{Kmeans}: $(F_{\text{semi}})_{1,1} = 0.05$ vs. $(C_{\text{Kmeans}})_{1,1} = 0.29$ vs. and $(F_{\text{semi}})_{4,2} = 0.08$ vs. $(C_{\text{Kmeans}})_{4,2} = 0.36$. Thus restrictions on F can have large effects on subspace factorization. Convex-NMF gives F factors that are closer to cluster centroids, validating our expectation that this factorization produces centroid-like factors. More examples are given in Figure 1.

Finally, computing the residual values, we have $||X - FG^T|| = 0.27940,\ 0.27944,\ 0.30877$, for SVD, Semi-NMF and Convex-NMF, respectively. We see that the enhanced interpretability provided by Semi-NMF is not accompanied by a degradation in approximation accuracy relative to the SVD. The more highly constrained Convex-NMF involves a modest degradation in accuracy.

We now turn to a presentation of algorithms for computing the two new factorizations, together with theoretical results establishing convergence of these algorithms.

III. Algorithms and analysis

In this section we provide algorithms and accompanying analysis for the NMF factorizations that we presented in the previous section.

A. Algorithm for Semi-NMF

We compute the Semi-NMF factorization via an iterative updating algorithm that alternatively updates F and G:

(S0) Initialize G. Do a K-means clustering. This gives cluster indicators G: $G_{ik} = 1$ if $\mathbf{x}_i$ belongs to cluster k. Otherwise, $G_{ik} = 0$. Add a small constant (we use the value 0.2 in practice) to all elements of G. See Section IIV.C for more discussion on initialization.

(S1) Update F (while fixing G) using the rule

$$F = XG(G^TG)^{-1}. \tag{10}$$

Note G^TG is a $k \times k$ positive semidefinite matrix. The inversion of this small matrix is trivial. In most cases, G^TG is nonsingular. When G^TG is singular, we take the pseudo inverse.

(S2) Update G (while fixing F) using

$$G_{ik} \leftarrow G_{ik}\sqrt{\frac{(X^TF)^+_{ik} + [G(F^TF)^-]_{ik}}{(X^TF)^-_{ik} + [G(F^TF)^+]_{ik}}}, \tag{11}$$

where we separate the positive and negative parts of a matrix A as

$$A^+_{ik} = (|A_{ik}| + A_{ik})/2, \qquad A^-_{ik} = (|A_{ik}| - A_{ik})/2. \tag{12}$$

The computational complexity for Semi-NMF is of order $m(pnk+nk^2)$ for Step (S1) and of order $m(npk + kp^2 + n^2k)$ for Eq. (11), where $m \sim 100$ is the number of iterations to convergence.

Theorem 1: (A) Fixing F, the residual $||X - FG^T||^2$ decreases monotonically (i.e., it is non-increasing) under the update rule for G. (B) Fixing G, the update rule for F gives the optimal solution to $\min_F ||X - FG||^2$.

Proof. We first prove part (B). The objective function that we minimize is the following sum of squared residuals:

$$J = ||X - FG^T||^2 = \text{Tr } (X^TX - 2X^TFG^T + GF^TFG^T). \tag{13}$$

Fixing G, the solution for F is obtained by computing $dJ/dF = -2XG + 2FG^TG = 0$. This gives the solution $F = XG(G^TG)^{-1}$.

To prove part (A), we now fix F and solve for G while imposing the restriction $G \geq 0$. This is a constrained optimization problem. We present two results: (1) We show that at convergence, the limiting solution of the update rule of Eq. (11) satisfies the KKT condition. This is established in Proposition 2 below. This proves the correctness of the limiting solution. (2) We show that the iteration of the update rule of Eq. (11) converges. This is established in Proposition 3 below. □

Proposition 2: The limiting solution of the update rule in Eq. (11) satisfies the KKT condition.

Proof. We introduce the Lagrangian function

$$L(G) = \mathrm{Tr}\,(-2X^T F G^T + G F^T F G^T - \beta G^T), \tag{14}$$

where the Lagrangian multipliers β_{ij} enforce nonnegative constraints, $G_{ij} \geq 0$. The zero gradient condition gives $\frac{\partial L}{\partial G} = -2X^T F + 2GF^T F - \beta = 0$. From the complementary slackness condition, we obtain

$$(-2X^T F + 2GF^T F)_{ik} G_{ik} = \beta_{ik} G_{ik} = 0. \tag{15}$$

This is a fixed point equation that the solution must satisfy at convergence.

It is easy to see that the limiting solution of the update rule of Eq. (11) satisfies the fixed point equation. At convergence, $G^{(\infty)} = G^{(t+1)} = G^{(t)} = G$; i.e.,

$$G_{ik} = G_{ik} \sqrt{\frac{(X^T F)^+_{ik} + [G(F^T F)^-]_{ik}}{(X^T F)^-_{ik} + [G(F^T F)^+]_{ik}}}. \tag{16}$$

Note $F^T F = (F^T F)^+ - (F^T F)^-$; $F^T X = (F^T X)^+ - (F^T X)^-$. Thus Eq. (16) reduces to

$$\left(-2X^T F + 2GF^T F\right)_{ik} G^2_{ik} = 0. \tag{17}$$

Eq. (17) is identical to Eq. (15). Both equations require that at least one of the two factors is equal to zero. The first factor in both equations are identical. For the second factor G_{ik} or G^2_{ik}, if $G_{ik} = 0$ then $G^2_{ik} = 0$, and vice versa. Thus if Eq. (15) holds, Eq. (17) also holds and vice versa. □

Next we prove the convergence of the iterative update algorithm. We need to state two propositions that are used in the proof of convergence.

Proposition 3: The residual of Eq. (13) is monotonically decreasing (non-increasing) under the update given in Eq. (11) for fixed F.

Proof. We write $J(H)$ as

$$J(H) = \mathrm{Tr}(-2H^T B^+ + 2H^T B^- + HA^+ H^T - HA^- H^T) \tag{18}$$

where $A = F^T F$, $B = X^T F$, and $H = G$.

We use the auxiliary function approach (2). A function $Z(H, \tilde{H})$ is called an auxiliary function of $J(H)$ if it satisfies

$$Z(H, \tilde{H}) \geq J(H), \quad Z(H, H) = J(H), \tag{19}$$

for any $H, \tilde{H}$. Define

$$H^{(t+1)} = \arg\min_{H} Z(H, H^{(t)}), \tag{20}$$

where we note that we require the global minimum. By construction, we have $J(H^{(t)}) = Z(H^{(t)}, H^{(t)}) \geq Z(H^{(t+1)}, H^{(t)}) \geq J(H^{(t+1)})$. Thus $J(H^{(t)})$ is monotone decreasing (non-increasing). The key is to find (1) appropriate $Z(H, \tilde{H})$ and (2) its global minimum. According to Proposition 4 (see below), $Z(H, H')$ defined in Eq. (21) is an auxiliary function of J and its minimum is given by Eq. (22). According to Eq. (20), $H^{(t+1)} \leftarrow H$ and $H^{(t)} \leftarrow H'$; substituting $A = F^T F$, $B = F^T X$, and $H = G$, we recover Eq. (11). □

Proposition 4: Given the objective function J defined as in Eq. (18), where all matrices are nonnegative, the following function

$$\begin{aligned} Z(H, H') = & -\sum_{ik} 2B^+_{ik} H'_{ik} (1 + \log \frac{H_{ik}}{H'_{ik}}) + \sum_{ik} B^-_{ik} \frac{H^2_{ik} + H'^2_{ik}}{H'_{ik}} \\ & + \sum_{ik} \frac{(H'A^+)_{ik} H^2_{ik}}{H'_{ik}} - \sum_{ik\ell} A^-_{k\ell} H'_{ik} H'_{i\ell} (1 + \log \frac{H_{ik} H_{i\ell}}{H'_{ik} H'_{i\ell}}) \end{aligned} \tag{21}$$

is an auxiliary function for $J(H)$; i.e., it satisfies the requirements $J(H) \leq Z(H, H')$ and $J(H) = Z(H, H)$. Furthermore, it is a convex function in H and its global minimum is

$$H_{ik} = \arg\min_{H} Z(H, H') = H'_{ik} \sqrt{\frac{B^+_{ik} + (H'A^-)_{ik}}{B^-_{ik} + (H'A^+)_{ik}}}. \tag{22}$$

Proof. The function $J(H)$ is

$$J(H) = \text{Tr}(-2H^T B^+ + 2H^T B^- + HA^+ H^T - HA^- H^T). \tag{23}$$

We find upper bounds for each of the two positive terms, and lower bounds for each of the two negative terms. For the third term in $J(H)$, using Proposition 5 (see below) and setting $A \leftarrow I, B \leftarrow A^+$, we obtain an upper bound

$$\text{Tr}(HA^+ H^T) \leq \sum_{ik} \frac{(H'A^+)_{ik} H^2_{ik}}{H'_{ik}}.$$

The second term of $J(H)$ is bounded by

$$\text{Tr}(H^T B^-) = \sum_{ik} H_{ik} B^-_{ik} \leq \sum_{ik} B^-_{ik} \frac{H^2_{ik} + H'^2_{ik}}{2H'_{ik}},$$

using the inequality $a \le (a^2+b^2)/2b$, which holds for any $a, b > 0$.

To obtain lower bounds for the two remaining terms, we use the inequality $z \ge 1 + \log z$, which holds for any $z > 0$, and obtain

$$\frac{H_{ik}}{H'_{ik}} \ge 1 + \log\frac{H_{ik}}{H'_{ik}}, \tag{24}$$

and

$$\frac{H_{ik}H_{i\ell}}{H'_{ik}H'_{i\ell}} \ge 1 + \log\frac{H_{ik}H_{i\ell}}{H'_{ik}H'_{i\ell}}. \tag{25}$$

From Eq. (24), the first term in $J(H)$ is bounded by

$$\mathrm{Tr}(H^T B^+) = \sum_{ik} B^+_{ik} H_{ik} \ge \sum_{ik} B^+_{ik} H'_{ik}(1 + \log\frac{H_{ik}}{H'_{ik}}).$$

From Eq. (25), the last term in $J(H)$ is bounded by

$$\mathrm{Tr}(HA^- H^T) \ge \sum_{ik\ell} A^-_{k\ell} H'_{ik} H'_{i\ell}(1 + \log\frac{H_{ik}H_{i\ell}}{H'_{ik}H'_{i\ell}}).$$

Collecting all bounds, we obtain $Z(H, H')$ as in Eq. (21). Obviously, $J(H) \le Z(H, H')$ and $J(H) = Z(H, H)$.

To find the minimum of $Z(H, H')$, we take

$$\frac{\partial Z(H,H')}{\partial H_{ik}} = -2B^+_{ik}\frac{H'_{ik}}{H_{ik}} + 2B^-_{ik}\frac{H_{ik}}{H'_{ik}} \quad + \quad \frac{2(H'A^+)_{ik}H_{ik}}{H'_{ik}} - 2\frac{(H'A^-)_{ik}H'_{ik}}{H_{ik}}. \tag{26}$$

The Hessian matrix containing the second derivatives

$$\frac{\partial^2 Z(H,H')}{\partial H_{ik}\partial H_{j\ell}} = \delta_{ij}\delta_{k\ell}Y_{ik}$$

is a diagonal matrix with positive entries

$$Y_{ik} = \frac{4[(B^+)_{ik} + (H'A^-)_{ik}]H'_{ik}}{H^2_{ik}} + 2\frac{B^-_{ik} + (H'A^+)_{ik}}{H'_{ik}}$$

Thus $Z(H, H')$ is a convex function of H. Therefore, we obtain the global minimum by setting $\partial Z(H, H')/\partial H_{ik} = 0$ in Eq. (26) and solving for H. Rearranging, we obtain Eq. (22). □

Proposition 5: For any matrices $A \in \mathbb{R}^{n\times n}_+, B \in \mathbb{R}^{k\times k}_+, S \in \mathbb{R}^{n\times k}_+, S' \in \mathbb{R}^{n\times k}_+$, with A and B symmetric, the following inequality holds:

$$\sum_{i=1}^{n}\sum_{p=1}^{k}\frac{(AS'B)_{ip}S^2_{ip}}{S'_{ip}} \ge \mathrm{Tr}(S^T ASB). \tag{27}$$

Proof. Let $S_{ip} = S'_{ip} u_{ip}$. Using an explicit index, the difference Δ between the left-hand side and the right-hand side can be written as

$$\Delta = \sum_{i,j=1}^{n} \sum_{p,q=1}^{k} A_{ij} S'_{jq} B_{qp} S'_{ip} (u_{ip}^2 - u_{ip} u_{jq}).$$

Because A and B are symmetric, this is equal to

$$\Delta = \sum_{i,j=1}^{n} \sum_{p,q=1}^{k} A_{ij} S'_{jq} B_{qp} S'_{ip} (\frac{u_{ip}^2 + u_{jq}^2}{2} - u_{ip} u_{jq}) = \frac{1}{2} \sum_{i,j=1}^{n} \sum_{p,q=1}^{k} A_{ij} S'_{jq} B_{qp} S'_{ip} (u_{ip}^2 - u_{jq}^2)^2 \geq 0.$$

□

In the special case in which $B = I$ and S is a column vector, this result reduces to a result due to (2).

B. Algorithm for Convex-NMF

We describe the algorithm for computing the Convex-NMF factorization when X has mixed sign (denoted as $X_{\pm}$). When X is nonnegative, the algorithm is simplified in a natural way.

(C0) Initialize W and G. There are two methods. (A) Fresh start. Do a K-means clustering. Let the obtained cluster indicators be $H = (\mathbf{h}_1, \cdots, \mathbf{h}_k)$, $H_{ik} = \{0, 1\}$. Then set $G^{(0)} = H + 0.2E$, where E is a matrix of all 1's. The cluster centroids can be computed as $\mathbf{f}_k = X\mathbf{h}_k/n_k$, or $F = XHD_n^{-1}$, where $D_n = \mathrm{diag}(n_1, \cdots, n_k)$. Thus $W = HD_n^{-1}$. We smooth W and set $W^{(0)} = (H + 0.2E)D_n^{-1}$. (B) Suppose we already have an NMF or Semi-NMF solution. In this case G is known and we set $G^{(0)} = G + 0.2E$. We solve $X = XWG^T$ for W. This leads to $W = G(G^TG)^{-1}$. Since W must be nonnegative, we set $W^{(0)} = W^+ + 0.2E\langle W^+ \rangle$, where $\langle A \rangle = \sum_{ij} |A_{ij}|/||A||_0$ and where $||A||_0$ is the number of nonzero elements in A.

Then update G_+ and W_+ alternatively until convergence as follows:

(C1) Update G_+ using

$$G_{ik} \leftarrow G_{ik} \sqrt{\frac{[(X^TX)^+W]_{ik} + [GW^T(X^TX)^-W]_{ik}}{[(X^TX)^-W]_{ik} + [GW^T(X^TX)^+W]_{ik}}}. \qquad (28)$$

This can be derived in a manner similar to Eq. (11), replacing F by XW;

(C2) Update W_+ using

$$W_{ik} \leftarrow W_{ik} \sqrt{\frac{[(X^TX)^+G]_{ik} + [(X^TX)^-WG^TG]_{ik}}{[(X^TX)^-G]_{ik} + [(X^TX)^+WG^TG]_{ik}}}. \qquad (29)$$

The computational complexity for convex-NMF is of order $n^2p+m(2n^2k+nk^2)$ for Eq. (28) and is of order $m(2n^2k+2nk^2)$ for Eq. (29), where $m \sim 100$ is the number of iterations to convergence. These are matrix multiplications and can be computed efficiently on most computers.

The correctness and convergence of the algorithm are addressed in the following:

Theorem 6: Fixing G, under the update rule for W of Eq. (29), (A) the residual $||X - XWG^T||^2$ decreases monotonically (non-increasing), and (B) the solution converges to a KKT fixed point.

The proof of part (B) is given by Proposition 7, which ensures the correctness of the solution. The proof of part (A) is given by Proposition 8, which ensures the convergence of the algorithm.

Proposition 7: The limiting solution of update rule of Eq. (29) satisfies the KKT condition.

Proof. We minimize

$$J_2 = ||X - XWG^T||^2 = \text{Tr }(X^TX - 2G^TX^TXW + W^TX^TXWG^TG),$$

where $X \in \mathbb{R}^{p\times n},\ W \in \mathbb{R}_+^{n\times k},\ G \in \mathbb{R}_+^{k\times n}$. The minimization with respect to G is the same as in Semi-NMF. We focus on the minimization with respect to W; that is, we minimize

$$J(W) = \text{Tr }(-2G^TX^TXW + W^TX^TXWG^TG). \tag{30}$$

We can easily obtain the KKT complementarity condition

$$(-X^TXG + X^TXWG^TG)_{ik}W_{ik} = 0. \tag{31}$$

Next we can show that the limiting solution of the update rule of Eq. (29) satisfies

$$(-X^TXG + X^TXWG^TG)_{ik}W_{ik}^2 = 0. \tag{32}$$

These two equations are identical for the same reasons that Eq. (17) is identical to Eq. (15). Thus the limiting solution of the update rule satisfies the KKT fixed point condition. □

Proposition 8: The residual, Eq. (30), decreases monotonically (it is non-increasing). Thus the algorithm converges.

Proof. We write $J(W)$ as

$$J(H) = \text{Tr }(-2H^TB^+ + 2H^TB^- + H^TA^+HC - H^TA^-HC), \tag{33}$$

where $B = X^TXG$, $A = X^TX$, $C = G^TG, H = W$. $J(H)$ differs from $J(H)$ of Eq. (23) in that the last two terms has four matrix factors instead of three. Following the proof of Proposition 4, with the aid of Proposition 5, we can prove that the following function

$$\begin{aligned} Z(H,H') = \quad & -\sum_{ik} 2B^+_{ik}H'_{ik}(1+\log\frac{H_{ik}}{H'_{ik}}) + \sum_{ik} B^-_{ik}\frac{H^2_{ik}+H'^2_{ik}}{H'_{ik}} \\ & +\sum_{ik}\frac{(A^+H'C)_{ik}H^2_{ik}}{H'_{ik}} - \sum_{ijk\ell} A^-_{ij}H'_{jk}C_{k\ell}H'_{i\ell}(1+\log\frac{H_{jk}H_{i\ell}}{H'_{jk}H'_{i\ell}}) \end{aligned} \tag{34}$$

is an auxiliary function of $J(H)$, and furthermore, $Z(H,H')$ is a convex function of H and its global minimum is

$$H_{ik} = \arg\min_{H} = H'_{ik}\sqrt{\frac{B^+_{ik}+(A^-H'C)_{ik}}{B^-_{ik}+(A^+H'C)_{ik}}}. \tag{35}$$

From its minima and setting $H^{(t+1)} \leftarrow H$ and $H^{(t)} \leftarrow H'$, we recover Eq. (29), letting $B^+ = (X^TX)^+G$, $B^- = (X^TX)^-G$, $A = X^TX$, $C = G^TG$ and $H = W$. □

C. Some generic properties of NMF algorithms

First, we note all these multiplicative updating algorithms are guaranteed to converge to a local minimum, but not necessarily to a global minimum. This is also true for many other algorithms that have a clustering flavor, including K-means, EM for mixture models, and spectral clustering. Practical experience suggests that K-means clustering tends to converge to a local minimum that is close to the initial guess, whereas NMF and EM tend to explore a larger range of values.

Second, we note that NMF updating rules are invariant with respect to rescaling of NMF. By rescaling, we mean $FG^T = (FD^{-1})(GD^T)^T = \widetilde{F}\widetilde{G}^T$, where D is a k-by-k positive diagonal matrix. Under this rescaling, Eq. (11) becomes

$$\widetilde{G}_{ik} \leftarrow \widetilde{G}_{ik}\sqrt{\frac{(X^T\widetilde{F})^+_{ik}+[\widetilde{G}(\widetilde{F}^T\widetilde{F})^-]_{ik}}{(X^T\widetilde{F})^-_{ik}+[\widetilde{G}(\widetilde{F}^T\widetilde{F})^+]_{ik}}}, \tag{36}$$

Since $(X^T\widetilde{F})_{ik} = (X^TF)_{ik}D^{-1}_{kk}$, $(\widetilde{G}\widetilde{F}^T\widetilde{F})_{ik} = (\widetilde{G}\widetilde{F}^TF)_{ik}D^{-1}_{kk}$ and $(\widetilde{G})_{ik} = (G)_{ik}D_{kk}$, Eq. (36) is identical to Eq. (11).

Third, we note that the convergence rate of the NMF multiplicative updating algorithm is generally of first order. To see this, we set $\Theta = \begin{pmatrix} F \\ G \end{pmatrix}$ and view the updating algorithms as a mapping $\Theta^{(t+1)} = M(\Theta^{(t)})$. At convergence, $\Theta^* = M(\Theta^*)$. The objective functions have been proved to be non-increasing, $J(\Theta^{(t+1)}) \leq J(\Theta^{(t)})$. Following Xu & Jordan (26), we expand[1] $\Theta \simeq M(\Theta^*) + (\partial M/\partial\Theta)(\Theta - \Theta^*)$. Therefore,

$$\|\Theta^{(t+1)} - \Theta^*\| \leq \|\frac{\partial M}{\partial \Theta}\| \cdot \|\Theta^{(t)} - \Theta^*\|$$

under an appropriate matrix norm. In general, $\partial M/\partial\Theta \neq 0$. Thus these updating algorithms have a first-order convergence rate, which is the same as the EM algorithm (26).

Fourth, we note that there are many ways to initialize NMF. In our paper, we use the equivalence between NMF and relaxed K-means clustering to initialize F and G to the K-means clustering solution. Lee and Seung (2) suggest random initialization. An SVD-based initialization has recently been proposed by Boutsidis and Gallopoulos (27). See more initialization references in (27; 17).

IV. Sparsity of Convex-NMF

In the original presentation of NMF, (1) emphasized the desideratum of *sparsity*. For example, in the case of image data, it was hoped that NMF factors would correspond to a coherent part of the original image, for example a nose or an eye; these would be sparse factors in which most of the components would be zero. Further experiments have shown, however, that NMF factors are not necessarily sparse, and sparsification schemes have been developed on top of NMF (16; 5). Parts-of-whole representations are not necessarily recovered by NMF, but conditions for obtaining parts-of-whole representations have been discussed (28). See also (29) (30), and (31) for related literatures on sparse factorizations in the context of PCA.

Interestingly, the Convex-NMF factors W and G are naturally sparse. We provide theoretical support for this assertion in this section, and provide additional experimental support in Section VI. (Sparseness can also be seen in the illustrative example presented in Section II-C).

We first note that Convex-NMF can be reformulated as

$$\min_{W,G\geq 0} \sum_k \sigma_k^2 ||\mathbf{v}_k^T(I - WG^T)||^2, \qquad \text{s.t.} \quad W \in \mathcal{R}_+^{n\times k}, G \in \mathcal{R}_+^{k\times n}, \tag{37}$$

where we use the SVD of $X = U\Sigma V^T$ and thus have $X^TX = \sum_k \sigma_k^2 \mathbf{v}_k \mathbf{v}_k^T$. Therefore $||X - XWG^T||^2 = \text{Tr } (I - GW^T)X^TX(I - WG^T) = \sum_k \sigma_k^2 ||\mathbf{v}_k^T(I - WG^T)||^2$. We now claim that

[1] Note that a nonnegativity constraint needs to be enforced.

(a) this optimization will produce a sparse solution for W and G, and (b) the more slowly σ_k decreases, the sparser the solution.

This second part of our argument is captured in a Lemma:

Lemma 9: The solution of the following optimization problem

$$\min_{W,G\geq 0} ||I - WG^T||^2, \qquad \text{s.t.} \quad W, G \in \mathcal{R}_+^{n\times K},$$

is given by $W = G =$ any K columns of $(\mathbf{e}_1 \cdots \mathbf{e}_K)$, where $\mathbf{e}_k$ is a basis vector: $(\mathbf{e}_k)_{i\neq k} = 0$, $(\mathbf{e}_k)_{i=k} = 1$.

Proof. We first prove the result for a slightly more general case. Let $D = \mathrm{diag}(d_1, \cdots, d_n)$ be a diagonal matrix and let $d_1 > d_2 > \cdots > d_n > 0$. The Lemma holds if we replace I by D and $W = G = (\sqrt{d_1}\mathbf{e}_1 \cdots \sqrt{d_1}\mathbf{e}_K)$.[2] The proof follows from the fact that we have a unique spectral expansion $D\mathbf{e}_k = d_k\mathbf{e}_k$ and $D = \sum_{k=1}^n d_k \mathbf{e}_k \mathbf{e}_k^T$. Now we take the limit: $d_1 = \cdots = d_n = 1$. The spectral expansion is not unique: $I = \sum_{k=1}^n \mathbf{u}_k \mathbf{u}_k^T$ for any orthogonal basis $(\mathbf{u}_1, \cdots, \mathbf{u}_n) = U$. However, due the nonnegativity constraint, $(\mathbf{e}_1 \cdots \mathbf{e}_n)$ is the only viable basis. Thus $W = G =$ for any K columns of $(\mathbf{e}_1 \cdots \mathbf{e}_n)$. □

The main point of Lemma 9 is that the solutions to $\min_{W,G} ||I - WG^T||^2$ are the sparsest possible rank-K matrices W, G. Now returning to our characterization of Convex-NMF in Eq. (37), we can write

$$||I - WG^T||^2 = \sum_k ||\mathbf{e}_k^T(I - WG^T)||^2.$$

Comparing to the Convex-NMF case, we see that the projection of $(I - WG^T)$ onto the principal components has more weight while the projection of $(I - WG^T)$ onto the non-principal components has less weight. Thus we conclude that sparsity is enforced strongly in the principal component subspace and weakly in the non-principal component subspace. Overall, Lemma 9 provides a basis for concluding that Convex-NMF tends to yield sparse solutions.

A more intuitive understanding of the source of the sparsity can be obtained by counting parameters. Note in particular that Semi-NMF is based on $N_{param} = kp + kn$ parameters whereas Convex-NMF is based on $N_{param} = 2kn$ parameters. Considering the usual case $n > p$ (i.e., the number of data points is more than the data dimension), Convex-NMF has more parameters than

[2] In NMF, the degree of freedom of diagonal rescaling is always present. Indeed, let $E = (\mathbf{e}_1 \cdots \mathbf{e}_K)$. Our choice of $W = G = E\sqrt{D}$ can be written in different ways $WG^T = (E\sqrt{D})(E\sqrt{D})^T = (ED^\alpha)(E^T D^{1-\alpha})^T$, where $-\infty < \alpha < \infty$.

Semi-NMF. But we know that Convex-NMF is a special case of Semi-NMF. The resolution of these two contradicting facts is that some of the parameters in Convex-NMF must be zero.

V. Additional remarks

Convex-NMF stands out for its interpretability and its sparsity properties. In this section we consider two additional interesting aspects of Convex-NMF and we also consider the relationship of all of the NMF-like factorizations that we have developed to K-means clustering.

A. Kernel-NMF

Consider a mapping $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$, or $X \rightarrow \phi(X) = (\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n))$. A standard NMF or Semi-NMF factorization $\phi(X) \approx FG^T$ would be difficult to compute since F and G depend explicitly on the mapping function $\phi(\cdot)$. However, Convex-NMF provides an appealing solution of this problem:

$$\phi(X) \approx \phi(X) W G^T.$$

Indeed, it is easy to see that the minimization objective

$$||\phi(X) - \phi(X)WG^T||^2 = \mathrm{Tr}[\phi(X)^T\phi(X) - 2G^T\phi^T(X)\phi(X)W + W^T\phi^T(X)\phi(X)WG^TG]$$

depends only on the kernel $K = \phi^T(X)\phi(X)$. In fact, the update rules for Convex-NMF presented in Eqs.(29) and (28) depend on X^TX only. Thus it is possible to "kernelize" Convex-NMF in a manner analogous to the kernelization of PCA and K-means.

B. Cluster-NMF

In Convex-NMF, we require the columns of F to be convex combinations of input data. Suppose now that we interpret the entries of G as posterior cluster probabilities. In this case the cluster centroids can be computed as $\mathbf{f}_k = X\mathbf{g}_k/n_k$, or $F = XGD_n^{-1}$, where $D_n = \mathrm{diag}(n_1, \cdots, n_k)$. The extra degree of freedom for F is not necessary. Therefore, the pair of desiderata: (1) F encodes centroids, and (2) G encodes posterior probabilities motivates a factorization $X \approx XGD_n^{-1}G^T$. We can absorb $D_n^{-\frac{1}{2}}$ into G and solve for

$$\text{Cluster-NMF}: \; X \approx XG_+G_+^T. \tag{38}$$

We call this factorization *Cluster-NMF* because the degree of freedom in this factorization is the cluster indicator G, as in a standard clustering problem. The objective function is $J = \|X - XGG^T\|^2$.

C. Relation to relaxed K-means clustering

NMF, Semi-NMF, Convex-NMF, Cluster-NMF and Kernel-NMF all have K-means clustering interpretations when the factor G is orthogonal ($G^TG = I$). Orthogonality and nonnegativity together imply that each row of G has only one nonnegative element; i.e., G is a bona fide cluster indicator. This relationship to clustering is made more precise in the following theorem.

Theorem 10: G-orthogonal NMF, Semi-NMF, Convex-NMF, Cluster-NMF and Kernel-NMF are all relaxations of K-means clustering.

Proof. For NMF, Semi-NMF and Convex-NMF, we first eliminate F. The objective is $J = \|X - FG^T\|^2 = \text{Tr}(X^TX - 2X^TFG^T + FF^T)$. Setting $\partial J/\partial F = 0$, we obtain $F = XG$. Thus we obtain $J = \text{Tr}(X^TX - G^TX^TXG)$. For Cluster-NMF, we obtain the same result directly: $J = \|X - XGG^T\|^2 = \text{Tr}(X^TX - G^TX^TXG)$. For Kernel-NMF, we have $J = \|\phi(X) - \phi(X)WG^T\|^2 = \text{Tr}(K - G^TKW + W^TKW)$, where K is the kernel. Setting $\partial J/\partial W = 0$, we have $KG = KW$. Thus $J = \text{Tr}(X^TX - G^TKG)$. In all five of these cases, the first terms are constant and do not affect the minimization. The minimization problem thus becomes $\max_{G^TG=I} \text{Tr}(G^TKG)$, where K is either a linear kernel X^TX or $\langle\phi(X), \phi(X)\rangle$. It is known that this is identical to (kernel-) K-means clustering (32; 33). □

In the definitions of NMF, Semi-NMF, Convex-NMF, Cluster-NMF and Kernel-NMF, G is not restricted to be orthogonal; these NMF variants are *soft* versions of K-means clustering.

VI. Experiments

We first present the results of an experiment on synthetic data which aims to verify that Convex-NMF can yield factorizations that are close to cluster centroids. We then present experimental results for real data comparing K-means clustering and the various factorizations.

A. Synthetic dataset

One main theme of our work is that the Convex-NMF variants may provide subspace factorizations that have more interpretable factors than those obtained by other NMF variants (or PCA). In particular, we expect that in some cases the factor F will be interpretable as containing cluster representatives (centroids) and G will be interpretable as encoding cluster indicators. We begin with a simple investigation of this hypothesis. In Figure 1, we randomly generate four two-dimensional datasets with three clusters each. Computing both the Semi-NMF and Convex-NMF factorizations, we display the resulting F factors. We see that the Semi-NMF factors (denoted $F_{\rm semi}$ in the figure) tend to lie distant from the cluster centroids. On the other hand, the Convex-NMF factors (denoted $F_{\rm cnvx}$) almost always lie within the clusters.

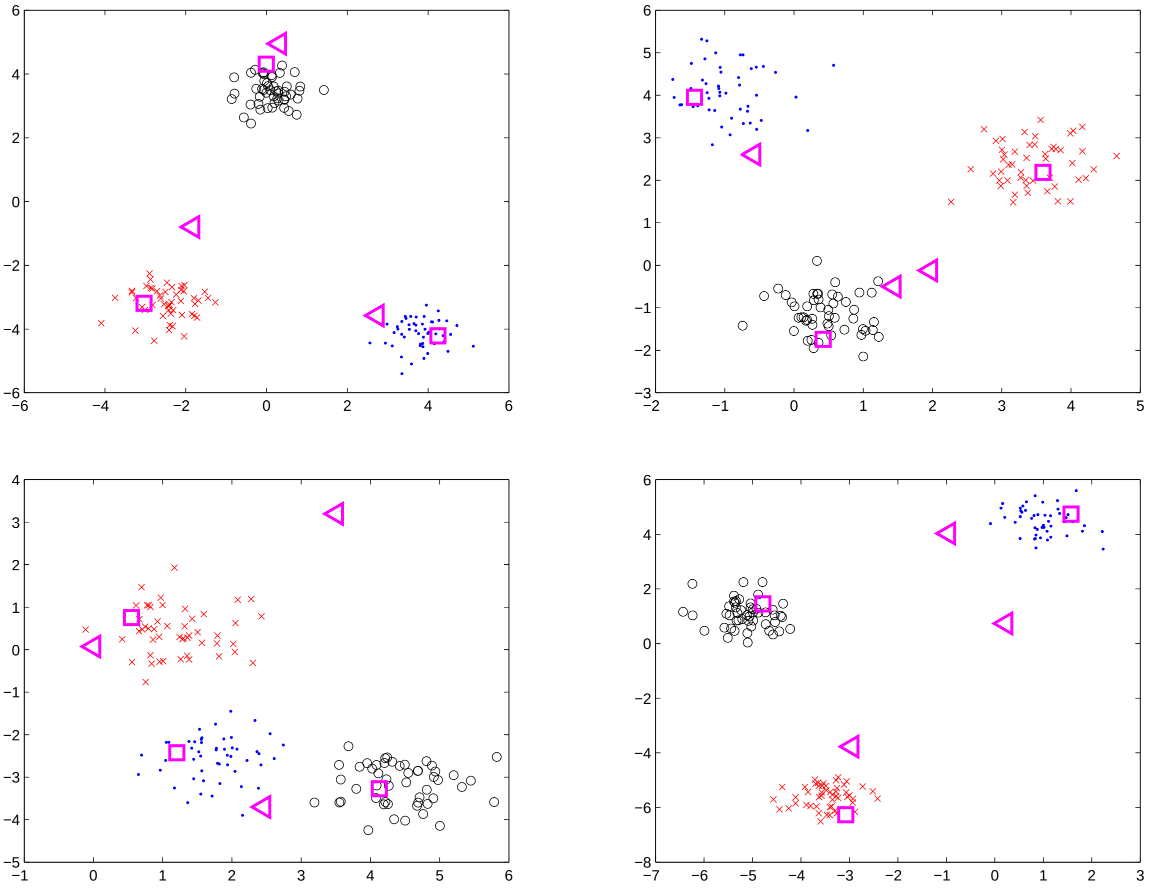

Fig. 1. Four random datasets, each with 3 clusters. "◁" are $F_{\rm semi}$ factors and "□" are $F_{\rm cnvx}$ factors.

B. *Real life datasets*

We conducted experiments on the following datasets: *Ionosphere* and *Wave* from the UCI repository, the document datasets *URCS*, *WebkB4*, *Reuters* (using a subset of the data collection which includes the 10 most frequent categories), *WebAce* and a dataset which contains 1367 log messages collected from several different machines with different operating systems at the

School of Computer Science at Florida International University. The log messages are grouped into 9 categories: *configuration*, *connection*, *create*, *dependency*, *other*, *report*, *request*, *start*, and *stop*. Stop words were removed using a standard stop list. The top 1000 words were selected based on frequencies.

Table I summarizes the datasets and presents our experimental results. These results are averages over 10 runs for each dataset and algorithm.

We compute clustering accuracy using the known class labels. This is done as follows: The confusion matrix is first computed. The columns and rows are then reordered so as to maximize the sum of the diagonal. We take this sum as a measure of the accuracy: it represents the percentage of data points correctly clustered under the optimized permutation.

To measure the sparsity of G in the experiments, we compute the average of each column of G and set all elements below 0.001 times the average to zero. We report the number of the remaining nonzero elements as a percentage of the total number of elements. (Thus small values of this measure correspond to large sparsity).

A consequence of the sparsity of G is that the rows of G tend to become close to orthogonal. This indicates a hard clustering (if we view G as encoding posterior probabilities for clustering). We compute the normalized orthogonality, $(G^T G)_{\text{nm}} = D^{-1/2}(G^T G)D^{-1/2}$, where $D = \text{diag}(G^T G)$. Thus $\text{diag}[(G^T G)_{\text{nm}}] = I$. We report the average of the off-diagonal elements in $(G^T G)_{\text{nm}}$ as the quantity "Deviation from Orthogonality" in the table.

From the experimental results, we observe the following: (1) All of the matrix factorization models are better than K-means on all of the datasets. This is our principal empirical result. It indicates that the NMF family is competitive with K-means for the purposes of clustering. (2) On most of the nonnegative datasets, NMF gives somewhat better accuracy than Semi-NMF and Convex-NMF (with WebKb4 the exception). The differences are modest, however, suggesting that the more highly-constrained Semi-NMF and Convex-NMF may be worthwhile options if interpretability is viewed as a goal of a data analysis. (3) On the datasets containing both positive and negative values (where NMF is not applicable) the Semi-NMF results are better in terms of accuracy than the Convex-NMF results. (3) In general, Convex-NMF solutions are sparse, while Semi-NMF solutions are not. (4) Convex-NMF solutions are generally significantly more orthogonal than Semi-NMF solutions.

TABLE I

DATASET DESCRIPTIONS AND RESULTS.

	Reuters	URCS	WebKB4	Log	WebAce	Ionosphere	Wave
data sign	+	+	+	+	+	±	±
# instance	2900	476	4199	1367	2340	351	5000
# class	10	4	4	9	20	2	2
Clustering Accuracy							
K-means	0.4448	0.4250	0.3888	0.6876	0.4001	0.4217	0.5018
NMF	0.4947	0.5713	0.4218	0.7805	0.4761	-	-
Semi-NMF	0.4867	0.5628	0.4378	0.7385	0.4162	0.5947	0.5896
Convex-NMF	0.4789	0.5340	0.4358	0.7257	0.4086	0.5470	0.5738
Sparsity (percentage of non-zeros in matrix G)							
Semi-NMF	0.9720	0.9688	0.9993	0.9104	0.9543	0.8177	0.9747
Convex-NMF	0.6152	0.6448	0.5976	0.5070	0.6427	0.4986	0.4861
Deviation from Orthogonality							
Semi-NMF	0.6578	0.5527	0.7785	0.5924	0.7253	0.9069	0.5461
Convex-NMF	0.1979	0.1948	0.1146	0.4815	0.5072	0.1604	0.2793

C. Shifting mixed-sign data to nonnegative

While our algorithms apply directly to mixed-sign data, it is also possible to consider shifting mixed-sign data to be nonnegative by adding the smallest constant so all entries are nonnegative. We performed experiments on data shifted in this way for the Wave and Ionosphere data. For Wave, the accuracy decreases to 0.503 from 0.590 for Semi-NMF and decreases to 0.5297 from 0.5738 for Convex-NMF. The sparsity increases to 0.586 from 0.498 for Convex-NMF. For Ionosphere, the accuracy decreases to 0.647 from 0.729 for Semi-NMF and decreases to 0.618 from 0.6877 for Convex-NMF. The sparsity increases to 0.829 from 0.498 for Convex-NMF. In short, the shifting approach does not appear to provide a satisfactory alternative.

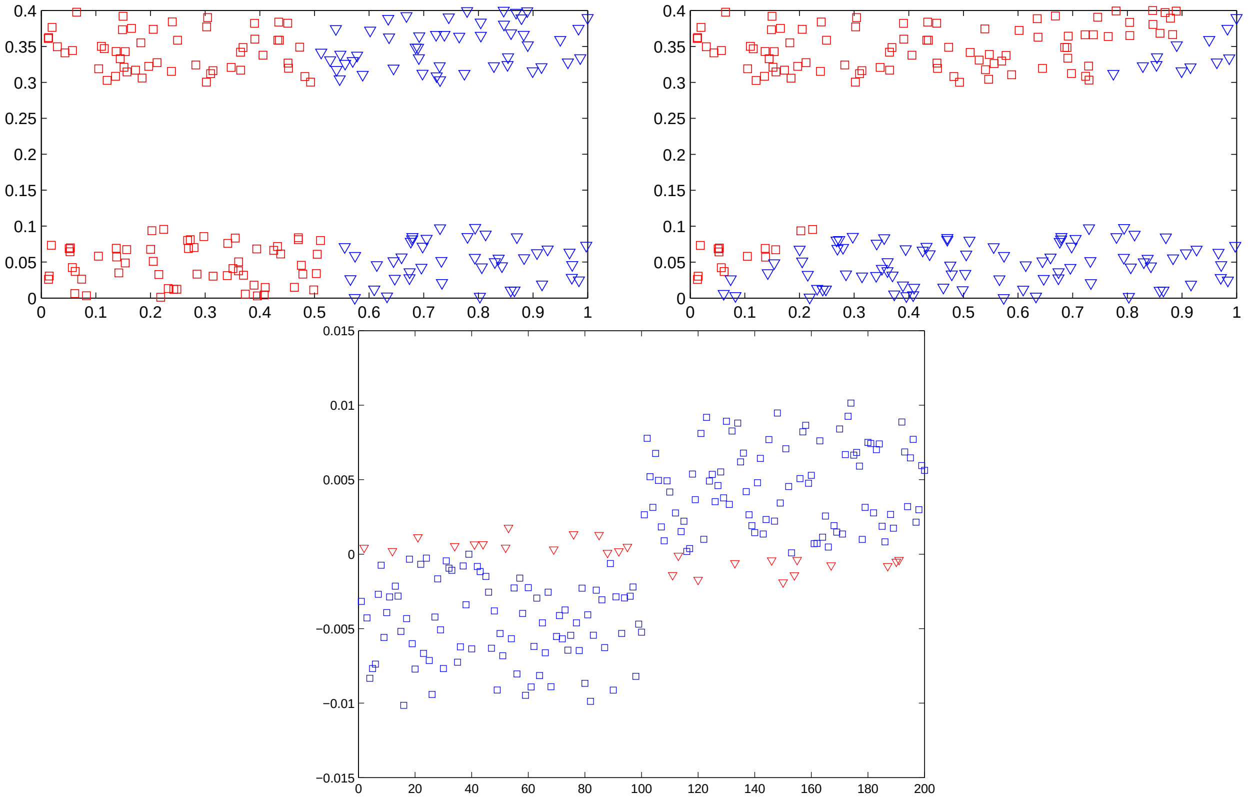

Fig. 2. A dataset with 2 clusters in 3D. Top Left: clusters obtained using K-means, as indicated by either "∇" or "$\square$". Top Right: clusters obtained using NMF. Bottom: The difference $\mathbf{g}_2(i) - \mathbf{g}_1(i)$, $i = 1, \cdots, 200$, "∇" for those mis-clustered points, and "$\square$" for correctly-clustered points.

D. *Flexibility of NMF*

A general conclusion is that NMF almost always performs better than K-means in terms of clustering accuracy while providing a matrix approximation. We believe this is due to the flexibility of matrix factorization as compared to the rigid spherical clusters that the K-means clustering objective function attempts to capture. When the data distribution is far from a spherical clustering, NMF may have advantages. Figure 2 gives an example. The dataset consists of two parallel rods in 3D space containing 200 data points. The two central axes of the rods are 0.3 apart and they have diameter 0.1 and length 1. As seen in the figure, K-means gives a poor clustering, while NMF yields a good clustering. The bottom panel of Figure 2 shows the differences in the columns of G (each column is normalized to $\sum_i \mathbf{g}_k(i) = 1$). The mis-clustered points have small differences.

Finally, note that NMF is initialized randomly for the different runs. We investigated the stability of the solution over multiple runs and found that NMF converges to solutions F and G that are very similar across runs; moreover, the resulting discretized cluster indicators were identical.

VII. Conclusions

We have presented a number of new nonnegative matrix factorizations. We have provided algorithms for these factorizations and theoretical analysis of the convergence of these algorithms. The ability of these algorithms to deal with mixed-sign data makes them useful for many applications, particularly given that covariance matrices are often centered.

Semi-NMF offers a low-dimensional representation of data points which lends itself to a convenient clustering interpretation. Convex-NMF further restricts the basis vectors to be convex combinations of data points, providing a notion of cluster centroids for the basis. We also briefly discussed additional NMF algorithms—Kernel-NMF and Cluster-NMF—that are further specializations of Convex-NMF.

We also showed that the NMF variants can be viewed as relaxations of K-means clustering, thus providing a closer tie between NMF and clustering than has been present in the literature to date. Moreover, our empirical results showed that the NMF algorithms all outperform K-means clustering on all of the datasets that we investigated in terms of clustering accuracy. We view these results as indicating that the NMF family is worthy of further investigation. We view Semi-NMF and Convex-NMF as particularly worthy of further investigation, given their native capability for handling mixed-sign data and their particularly direct connections to clustering.

Acknowledgments

We would like to thank the anonymous reviewers and the editor for suggesting various changes. Chris Ding is supported in part by a University of Texas STARS Award and NSF grant DMS-0844497. Tao Li is partially supported by NSF CAREER Award IIS-0546280 and NSF grant DMS-0844513. Michael Jordan is supported in part by a grant from Microsoft Research and by NSF grant 0509559.

REFERENCES

[1] D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.

[2] ——, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001.

[3] P. Paatero and U. Tapper, “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values,” *Environmetrics*, vol. 5, pp. 111–126, 1994.

[4] Y.-L. Xie, P. Hopke, and P. Paatero, “Positive matrix factorization applied to a curve resolution problem,” *Journal of Chemometrics*, vol. 12, no. 6, pp. 357–364, 1999.

[5] S. Li, X. Hou, H. Zhang, and Q. Cheng, “Learning spatially localized, parts-based representation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, pp. 207–212.

[6] M. Cooper and J. Foote, “Summarizing video using non-negative similarity matrix factorization,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, 2002, pp. 25–28.

[7] W. Xu, X. Liu, and Y. Gong, “Document clustering based on non-negative matrix factorization,” in *Proc. ACM Conf. Research development in IR(SIRGIR)*, 2003, pp. 267–273.

[8] V. P. Pauca, F. Shahnaz, M. Berry, and R. Plemmons, “Text mining using non-negative matrix factorization,” in *Proc. SIAM Int’l conf on Data Mining*, 2004, pp. 452–456.

[9] J.-P. Brunet, P. Tamayo, T. Golub, and J. Mesirov, “Metagenes and molecular pattern discovery using matrix factorization,” *Proc. Nat’l Academy of Sciences USA*, vol. 102, no. 12, pp. 4164–4169, 2004.

[10] 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.

[11] D. Greene, G. Cagney, N. Krogan, and P. Cunningham, “Ensemble non-negative matrix factorization methods for clustering protein-protein interactions,” *Bioinformatics*, vol. 24, no. 15, pp. 1722–1728, 2008.

[12] I. Dhillon and S. Sra, “Generalized nonnegative matrix approximations with Bregman

divergences," in *Advances in Neural Information Processing Systems 17.* MIT Press, 2005.

[13] C. Ding, T. Li, and W. Peng, "Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence, chi-square statistic, and a hybrid method," *Proc. National Conf. Artificial Intelligence*, 2006.

[14] F. Sha, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming in support vector machines," in *Advances in Neural Information Processing Systems 15.* Cambridge, MA: MIT Press, 2003.

[15] N. Srebro, J. Rennie, and T. Jaakkola, "Maximum margin matrix factorization," in *Advances in Neural Information Processing Systems.* Cambridge, MA: MIT Press, 2005.

[16] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[17] M. Berry, M. Browne, A. Langville, P. Pauca, and R. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *To Appear in Computational Statistics and Data Analysis*, 2006.

[18] T. Li and S. Ma, "IFD: Iterative feature and data clustering," in *Pro. SIAM Int'l conf. on Data Mining (SDM 2004)*, 2004, pp. 472–476.

[19] T. Li, "A general model for clustering binary data." in *KDD*, 2005, pp. 188–197.

[20] C. Ding, X. He, and H. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering." *Proc. SIAM Data Mining Conf*, 2005.

[21] E. Gaussier and C. Goutte, "Relation between PLSA and NMF and implications," in *Proc. of ACM SIGIR conference.* New York, NY, USA: ACM Press, 2005, pp. 601–602.

[22] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of ACM Conf. on Research and Development in Information Retrieval (SIGIR)*, 1999, pp. 50–57.

[23] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[24] M. Girolami and K. Kaban, "On an equivalence between PLSI and LDA," *Proc. ACM Conf. Research and Develop. Info. Retrieval (SIGIR)*, 2003.

[25] D. Lee and H. S. Seung, "Unsupervised learning by convex and conic coding," in *Advances in Neural Information Processing Systems 9.* Cambridge, MA: MIT Press, 1997.

[26] L. Xu and M. Jordan, "On convergence properties of the EM algorithm for gaussian

mixtures," *Neural Computation*, pp. 129–151, 1996.

[27] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization," *Pattern Recogn.*, vol. 41, no. 4, pp. 1350–1362, 2008.

[28] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press, 2004.

[29] A. D'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," *to appear in SIAM Review*, 2006.

[30] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Computational and Graphical Statistics*, vol. 15, pp. 265–286, 2006.

[31] Z. Zhang, H. Zha, and H. Simon, "Low-rank approximations with sparse factors II: Penalized methods with discrete Newton-like iterations," *SIAM J. Matrix Analysis Applications*, vol. 25, pp. 901–920, 2004.

[32] H. Zha, C. Ding, M. Gu, X. He, and H. Simon, "Spectral relaxation for K-means clustering," *Advances in Neural Information Processing Systems 14 (NIPS'01)*, pp. 1057–1064, 2002.

[33] C. Ding and X. He, "K-means clustering and principal component analysis," *Int'l Conf. Machine Learning (ICML)*, 2004.