Hybrid projective nonnegative matrix factorization based on α-divergence and the alternating least squares algorithm

Melisew Tefera Belachew[a,*], Nicoletta Del Buono[b]

[a] *Department of Mathematics, Haramaya University, Ethiopia*
[b] *Department of Mathematics, University of Bari-Aldo Moro, Italy*

ARTICLE INFO

Article history:
Received
Revised 2
Accepted
Available

MSC:
62-07
91C20
68T10
78M50
94A08
62H35

Keywords:
Hybrid projective nonnegative matrix factorization
α-Divergence
Feature extraction
Clustering
Orthogonality
Sparsity

ABSTRACT

Nonnegative Matrix Factorization (NMF) is a linear dimensionality reduction technique for extracting hidden and intrinsic features of high-dimensional data sets. Recently, several Projective NMF (P-NMF) methods have been proposed for the purpose of resolving issues associated with the standard NMF approach. Experimental results show that P-NMF algorithms outperform the standard NMF method in some aspects. But some basic issues still affect the existing NMF and P-NMF methods, these include slow convergence rate, low reconstruction accuracy and dense basis factors. In this article, we propose a new and generalized hybrid algorithm by combining the concept of alternating least squares with the multiplicative update rules of the α-divergence-based P-NMF method. We have conducted extensive numerical experiments on 7 real-world data sets and compared the new algorithm with several state-of-the-art methods. The attractive features and added advantages of the new algorithm include remarkable clustering performances, providing highly "orthogonal" and very sparse basis factors, and extracting distinctive and better localized features of the original data than its counterparts.

1. Introduction

Several data mining and analysis techniques can be employed for extracting basic and hidden features of high-dimensional data sets. Data sets obtained from real-world applications are nonnegative in nature and usually stored as high-dimensional nonnegative data matrices. Such data matrices do not allow negative entries because such entries are misleading and contradict physical realities. For the purpose of avoiding misinterpretations and ambiguous results, the decomposition of real-life data sets usually takes nonnegativity constraints into account. A linear dimensionality reduction technique called nonnegative matrix factorization (NMF) is very well-known for decomposing real-world data sets and providing nonnegative factors. Unlike other multivariate data analysis techniques such as vector quantization (VQ), singular value decomposition (SVD) and principal component analysis (PCA), NMF algorithms enable the so-called "additive parts-

[*] Corresponding author.
E-mail address: melisewt@gmail.com (M.T. Belachew).

https://doi.org/10.1016/j.amc.2019.124825

based" representation of data and summation of parts to make a whole [1–3]. On the contrary, the factors obtained from VQ, SVD and PCA contain both positive and negative entries, as a result they do not facilitate physical interpretations.

NMF algorithms are designed to solve the following optimization problem: given a data matrix $Y \in \mathbb{R}_+^{m\times n}$ and a reduced rank k, find two low-rank matrices $W \in \mathbb{R}_+^{m\times k}$ and $H \in \mathbb{R}_+^{k\times n}$ that approximate Y in a low-dimensional form as $Y \approx WH$. In this factorization, the factors W and H have different physical meanings in different applications. For instance, in image feature extraction and data clustering, W is a basis matrix (factor) and H is a weight matrix (factor). Each of the n columns in Y represent data points in an m-dimensional space, and k is the required number of basis vectors. In order to achieve the above approximation, one usually attempts to solve minimization problems that involve different kinds of cost functions. The most commonly used cost functions are the squared Frobenius norm, the generalized Kullback–Leibler (KL) divergence and the Amari α-divergence. In this paper, we are interested in the more general divergence measure of the above three, i.e., Amari's α-divergence. In fact, the learning algorithm derived from the α-divergence, namely α-NMF, is proved to be more flexible and efficient than those based on the Frobenius norm and the KL-divergence [4,5].

In general, the NMF minimization problem based on the α-divergence measure can be written as

$$\min_{W\in\mathbb{R}_+^{m\times k},\, H\in\mathbb{R}_+^{k\times n}} D_\alpha(Y\|WH) = \sum_{ij}\left(y_{ij}\frac{(y_{ij}/(WH)_{ij})^{\alpha-1}-1}{\alpha(\alpha-1)} + \frac{(WH)_{ij}-y_{ij}}{\alpha}\right), \tag{1}$$

where $\alpha \in (-\infty, +\infty)$.

Recently, methods called projective nonnegative matrix factorization (P-NMF) based on α-divergence (α-PNMF) have been proposed to solve problem (1) and improve the performances of α-NMF methods [6]. P-NMF algorithms are designed by projecting the data matrix Y by a nonnegative $m \times m$ approximative projection matrix $P = WW^\top$ of a given rank k onto a subspace of nonnegative matrices. It should be noted that we look for a matrix W whose columns are approximately orthogonal.[1] It has been reported that α-PNMF algorithms outperform α-NMF methods in some circumstances [6]. Unfortunately, α-PNMF also has some drawbacks; these include relatively dense basis factors and low reconstruction accuracies. In addition, we realized that there are possibilities for improvement concerning the orthogonality of the columns of the basis matrix (factor) W and the ability of learning localized features of the original data.

Several techniques are used for solving minimization problems that arise in NMF. Among these techniques, the multiplicative update rules (MUR) of Lee and Seung adopted for NMF, α-NMF, P-NMF, α-PNMF methods [1,2,6,7] and the alternating least squares (ALS) algorithms [3] could be mentioned. Since the existing α-NMF, P-NMF and α-PNMF algorithms are based solely on MUR, they inherit at least one major drawback called locking of zero entries (once an entry is zero it can never take another value; that is, it is locked). Due to this locking phenomenon, the aforementioned algorithms might be trapped at non-stationary points or even at saddle points. On the other hand, the ALS-based NMF algorithms are well-known for providing sparse factors and for their flexibility due to the absence of the aforementioned locking phenomenon.

Contributions and outline of the paper. In this paper, we propose a new and generalized hybrid algorithm that exploits the nice properties of the basic ALS algorithm and the multiplicative update rule of the α-PNMF method. According to the extensive numerical experiments conducted on 7 real-life data sets, this hybridizing strategy results in a new class of high-performance algorithms. The new algorithm is shown to outperform several state-of-the-art methods in various aspects. The advantages of the newly proposed algorithm includes high clustering performances, providing very sparse factors consisting of better localized features, and giving rise to highly "orthogonal" basis factors.

The remaining part of this article is organized as follows: Section 2 presents a short revision of α-NMF, α-PNMF and the basic ALS algorithm. A detailed discussion of the proposed algorithm is presented in Section 3. Section 4 is devoted to the numerical experiments conducted on various data sets coming from real-life applications. In Section 5, we present the concluding remarks.

2. Review of related works

This section presents a short survey on α-NMF, α-PNMF, and the basic ALS algorithm.

2.1. NMF based on alpha-divergence

Formally, the basic NMF problem can be stated as: given a nonnegative data matrix $Y \in \mathbb{R}_+^{m\times n}$ and a (small) rank $k < < \min(m, n)$, find two nonnegative matrices—a basis matrix $W \in \mathbb{R}_+^{m\times k}$ and an encoding matrix $H \in \mathbb{R}_+^{k\times n}$—such that

$$Y \approx WH. \tag{2}$$

In NMF, irrespective of the divergence measure used, the approximate factorization (2) is treated as a non-linear optimization problem. For α-NMF, the factors W and H are computed by minimizing Amari's α-divergence

$$D_\alpha(Y\|WH) = \sum_{ij}\left(y_{ij}\frac{(y_{ij}/(WH)_{ij})^{\alpha-1}-1}{\alpha(\alpha-1)} + \frac{(WH)_{ij}-y_{ij}}{\alpha}\right), \tag{3}$$

[1] Due to the nonnegativity constraints in NMF (and its variants), the basis matrix W (or its columns) can never be completely orthogonal. Therefore, in this paper, by orthogonality ("orthogonal") we mean approximate orthogonality in the sense of a small $\rho = \|W^\top W - I\|_F$, where $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$.

with respect to W and H. Here $\alpha \in (-\infty, \infty)$ is a pre-specified constant which is typically chosen in the interval (0,2]. The α-divergence is a quite general divergence measure in the sense that for special values $\alpha \to 1$, $\alpha \to 0$, $\alpha = 0.5$, and $\alpha = 2$, one obtains the generalized Kullback–Leibler (KL) divergence, the dual generalized KL divergence, the squared Hellinger's distance, and the Pearson's distance, respectively [4].

The α-NMF minimization process requires W and H to be subjected to the pointwise nonnegativity constraints $W \geq 0$ and $H \geq 0$. By projecting W and H onto ϕ and ψ spaces by the formulae $\phi(w_{ij}) = w_{ij}^{\alpha}$ and $\psi(h_{ij}) = h_{ij}^{\alpha}$, one can derive the multiplicative update rules (MUR)

$$w_{ij}^{(l+1)} = w_{ij}^{(l)} \left(\frac{\sum_k h_{jk}^{(l)} \left(\frac{y_{ik}}{[W^{(l)}H^{(l)}]_{ik}} \right)^{\alpha}}{\sum_p h_{jp}^{(l)}} \right)^{\frac{1}{\alpha}},$$

$$h_{ij}^{(l+1)} = h_{ij}^{(l)} \left(\frac{\sum_k w_{ki}^{(l+1)} \left(\frac{y_{kj}}{[W^{(l+1)}H^{(l)}]_{kj}} \right)^{\alpha}}{\sum_p w_{pi}^{(l+1)}} \right)^{\frac{1}{\alpha}},$$

where $w^{(l)}$ stands for the lth iterate. It is claimed [4,5] that the cost function (3) is non-increasing under the above update rules.

2.2. P-NMF based on alpha-divergence

P-NMF algorithms are designed by projecting the data matrix $Y \in \mathbb{R}_{+}^{m \times n}$ by a nonnegative $m \times m$ approximative projection matrix $P = WW^{\top}$ onto a subspace of nonnegative matrices, where $W \in \mathbb{R}_{+}^{m \times k}$ and k is a reduced rank fixed by the user. The objective function for the α-PNMF optimization problem is

$$D_{\alpha}(Y \| WW^{\top}Y) = \sum_{ij} \left(y_{ij} \frac{(y_{ij}/(WW^{\top}Y)_{ij})^{\alpha-1} - 1}{\alpha(\alpha - 1)} + \frac{(WW^{\top}Y)_{ij} - y_{ij}}{\alpha} \right). \tag{4}$$

The MUR for minimizing (4) with respect to W is given by [6]

$$w_{ik}^{(l+1)} = w_{ik}^{(l)} \left(\frac{\left(U^{(l)}Y^{\top}W^{(l)} + YU^{(l)\top}W^{(l)}\right)_{ik}}{\sum_j \left(W^{(l)\top}Y\right)_{kj} + \sum_j y_{ij} \sum_p w_{pk}^{(l)}} \right)^{\frac{1}{\alpha}}, \tag{5}$$

where $U = (Y./[W^{(l)}W^{(l)\top}Y])^{.\alpha}$, ./ stands for element-wise division, and $(\ )^{.\alpha}$ denotes element-wise exponentiation.

Yang and Oja [8] proved that the cost function (4) is non-increasing under the update rule (5).

2.3. The basic ALS algorithm for NMF

The alternating least squares (ALS) algorithms are one of the most widely used methods for solving minimization problems that arise in NMF. These class of algorithms are proven to work very well in practice. In fact, they are chosen for their sparse factors, flexibility and lack of the locking phenomenon among other things. These algorithms solve for the unknowns W and H from the equation $Y = WH$ in an iterated alternating two-step procedure by optimizing $\|Y - WH\|_F$. That is, one factor, say W, is fixed and a step of least squares is employed to solve $Y = WH$ for H and then H is fixed and another step of least squares is used to solve the above equation for W. This process continues in an alternating way until some stopping criteria are met; usually these algorithms are terminated when the maximum number of iterations set by the user is reached. For more information about the basic ALS algorithm and its variants, see [3].

3. Proposed algorithm

The α-PNMF algorithms discussed in the previous section managed to resolve some of the issues associated with standard α-NMF methods and work better in areas where the latter experiences some difficulties. However, some basic issues still affect α-NMF and α-PNMF; these include, unsatisfactory clustering performance, non-localized basis matrices, dense factors, and low reconstruction accuracies. For these reasons, the design of new algorithms that can address the aforementioned issues and improve the performance of the existing variants of α-NMF and α-PNMF methods is required.

In this section, we propose a new and generalized hybrid algorithm for solving the minimization problem given by (5).

The algorithm is designed in such a way that it exploits the nice properties of the basic ALS algorithm and the α-PNMF method. In particular, we propose a 2-stage hybrid projective nonnegative matrix factorization algorithm called α-HPNMF. α-HPNMF combines the ALS algorithm with the multiplicative update rule (5). By changing the value of α to 2, 1 and 0.5, we obtain other new hybrid algorithms called Pearson-HPNMF, KL-HPNMF, and Hellinger-HPNMF, respectively. The new algorithm consists of two separate stages. In the first stage we combine ALS with α-PNMF in order to benefit from the resulting sparse factors. Note that in P-NMF we need to minimize objective functions of the form $F(W) = \|Y - WW^\top Y\|_F$. One should observe that it is impossible to apply ALS on P-NMF (and α-PNMF) directly, for this reason one needs to introduce a matrix $H = W^\top Y$ and attempt to minimize $F(W, H) = \|Y - WH\|_F$ under the constraint $H = W^\top Y$. Therefore, in the newly proposed algorithms we first provide initialization for W and solve $Y = WH$ for H. But we require the solutions (H) obtained above to satisfy the constraint $H = W^\top Y$. For this reason, we use the above solution H as an input and solve $H = W^\top Y$ for W. At the end, we update W using the multiplicative update rule (5); see Algorithm 1 for the detail. In the first stage, solving for H and W continues in an alternating fashion until the desired value of *stage1iter* is reached, where *stage1iter* (number of iterations required for the first stage) is an integer defined a priori by the user. We will discuss how to choose *stage1iter* a little later. In the second stage of our algorithms, we solely update W on a separate loop using the multiplicative update rule (5). Orthogonality of the basis factor W is crucial for many applications such as text mining, feature extraction and data clustering [9,10]. In the new algorithms orthogonality of W is achieved through approximating the original data Y by PY, where $P = WW^\top$ is a projection matrix. Since W is nonnegative, approximating Y by $WW^\top Y$ can be achieved only when W is approximately orthogonal and $WW^\top \approx I$. In addition, the new algorithms combine ALS with the α-PNMF method which is shown to facilitate the orthogonality and sparsity of W.

The main steps of the proposed algorithm, α-HPNMF are summarized in Algorithm 1. It should be noted that the user is free to choose the value of *stage1iter* accordingly, and obviously suitable choices are made based on the quality and the precision needed. In our experiments, we realized that for small values of *stage1iter* the aforementioned algorithms tend to the multiplicative update rules. For this reason, we recommend values in the interval $[10, \mathit{maxiter})$ that happen to work very well in practice (see Section 4). Here, *maxiter* stands for maximum number of iterations. In Section 4.2, we investigate the behavior of Algorithm 1 for different values of *stage1iter*. One of the many advantages of these algorithms is that, since they combine ALS steps with that of the multiplicative update rules, the zero (very small) elements locked by the multiplicative update rules get a chance to take new values in the ALS steps. As a result, they provide some level of flexibility and enable the algorithms to lay on a very good and sustainable foundation. On the contrary, the multiplicative update rules of α-NMF and α-PNMF do not behave this way, that means, they suffer from locking of zero entries. This is one of the disadvantages of using these update rules and it is responsible for leading them to poor local minima and non-stationary points.

Remark 1. In the first stage of Algorithm 1, we use ALS to solve $W^\top Y = W^\top WH$ and $YH^\top = YY^\top W$. We learned from our various experiments that there are cases where the matrices $W^\top W$ and $YY^\top$ become singular, in such cases adding an identity matrix of the appropriate size avoids the non-invertibility issue.

Remark 2. In the new algorithm the iterates are updated using the multiplicative update rule of α-PNMF, see Algorithm 1. The details of the convergence properties of this update rule (5) are given in [8]. It should be noted that the ALS steps in the newly proposed algorithm are used only as inner initialization steps in the first stage. As a result, the new algorithm inherits the convergence properties of the α-PNMF method.

Computational complexity. Here, we calculate the computational complexity of Algorithm 1. Computational complexity is simply a measure of how many steps or operations are required for solving a given problem. In the case of NMF, computational complexity depends on the size of the matrices involved in solving the associated minimization problem. The steps of Algorithm 1 are lines 3, 7–11, and 17. Table 1 summarizes the approximate number of floating point operations

Algorithm 1 α-HPNMF.

```
1:  Input: $Y \in \mathbb{R}_+^{m\times n}$, a rank $k$ and a small positive number $\delta$;
2:  Initialize $W \in \mathbb{R}_+^{m\times k}$ and set maxiter and stage1iter;
3:  Compute $A = YY^\top$;
4:  begin
5:  for $i := 1$ to stage1iter do
6:      begin
7:      solve $W^\top Y = W^\top WH$ for $H = (h_{ij})$ using ALS;
8:      $H = (\mathtt{max}(h_{ij}, \delta))$;
9:      solve $YH^\top = AW$ for $W = (w_{ij})$ using ALS;
10:     $W = (\mathtt{max}(w_{ij}, \delta))$;
11:     use (5) to update $W$;
12:     end
13: end.
14: begin
15: for $i :=$ stage1iter $+ 1$ to maxiter do
16:     begin
17:     use (5) to update $W$;
18:     end
19: end.
```

Table 1
Rough estimate of the number of `flops` needed for the main steps in the tested algorithms: α-HPNMF, α-PNMF and α-NMF.

Steps		$\approx$ `#flops`
$A = YY^\top$	→	$m^2(2n-1)$
$B = W^\top W$	→	$k^2(2m-1)$
$C = W^\top Y$	→	$nk(2m-1)$
$H = \texttt{linsolve}(B, C)$	→	$\frac{2}{3}k^3 + 2nk(2k-1)$
$H = \texttt{max}(H, \delta)$	→	$2nk$
$D = YH^\top$	→	$mk(2m-1)$
$W = \texttt{linsolve}(A, D)$	→	$\frac{2}{3}m^3 + 2mk(2m-1)$
$W = \texttt{max}(W, \delta)$	→	$2mk$
(5)	→	$2m^2k + mk(8n-3) + 2nk(2m-1)$
α-HPNMF	→	$\frac{2}{3}(m^3 + k^3) + m^2(4k + 2n - 1) + k^2(2m + 4n - 1) + nk(2m-1) + mk(2m-1) - m - k$
α-PNMF	→	$2m^2k + mk(8n-3) + 2nk(2m-1)$
α-NMF	→	$12mnk + 2mn + 2nk$

(`flops`) needed for the aforementioned steps and the other tested algorithms. One can see from the aforementioned table that the newly proposed algorithm requires more `flops` when compared to the other methods. Actually, this is due to the additional computations in stage one of the said algorithm which are very crucial for obtaining quality factors that facilitate interpretation.

4. Experimental analysis

In this section, we present and analyze experimental results. We considered 7 real-world data sets. Particularly, four well-known data sets that contain photographs of faces of different persons were used for feature extraction and image analysis tasks. In addition, three commonly used data sets from the UCI repository are employed for clustering purposes. The description of each of the data sets is given in Sections 4.4.1 and 4.5.1, respectively. In order to evaluate and compare the behaviors of the tested algorithms, various evaluations have been performed on the basis of different qualitative and quantitative measures.

To speed up the comparison process, we provided plots of the approximation error (measured by Amari's α-divergence given in (4)) and that of orthogonality measure ρ given by

$$\rho = \|W^\top W - I\|_F, \ \|W(:, j)\|_2 = 1, \forall j = 1, 2, \ldots, k. \tag{6}$$

We have used (6) to measure the orthogonality of the basis matrix W. Note that $W(:, j)$ stands for the jth column of W. Obviously, the smaller ρ, the better orthogonality is approximated. Moreover, for the sake of speeding up the comparisons in the light of sparsity of factors and the ability in extracting better localized features, we have presented the images of the basis matrices arising from each of the tested algorithms. In addition, quantitative measures such as *Hoyer's sparseness*, the *τ-measure, purity* and *entropy* are employed to further assess the performances of the new and existing algorithms.

All experiments were conducted using MATLAB on a laptop Intel(R) Core(TM) i7-6500U CPU @2.50 GHz 2.59 GHz 8 GB RAM.

4.1. Initial conditions, parameters, and stopping criteria

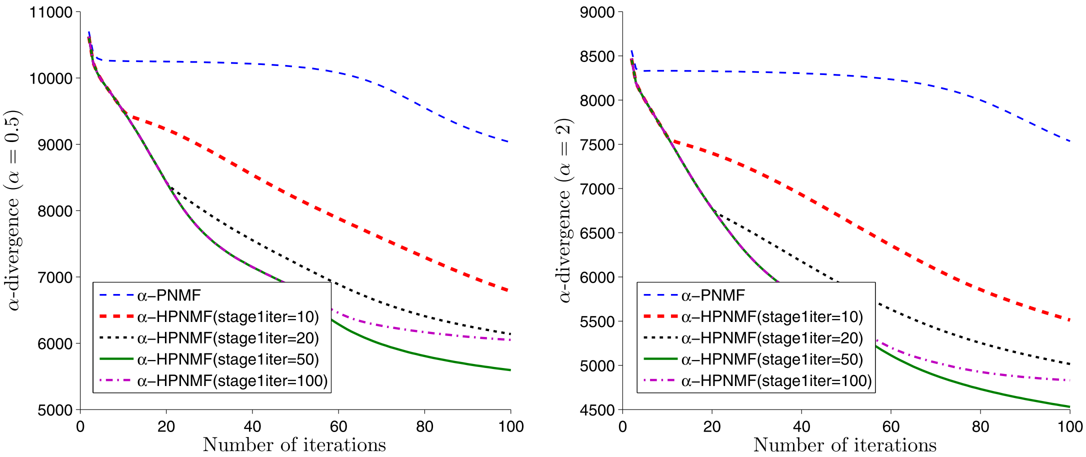
For all experiments, random initialization was used for its simplicity; that is, for a given data matrix $Y \in \mathbb{R}_+^{m \times n}$ and a reduced (small) rank k, we initialized the basis matrix W with $W_0 = \texttt{rand}(m, k)$[2] and H with $H_0 = \texttt{rand}(k, n)$. When working with image analysis and feature extraction, the parameter *stage1iter* was set to 30, 50, 100 and 20 for ORL, MIT, Georgia Tech and Yale data sets, respectively; while for experiments regarding data clustering *stage1iter*=50 was used in all cases. All iterations were terminated when the maximum number of iterations (*maxiter*) reached 200.

4.2. Behavior of Algorithm 1 for different values of stage1iter

In Section 3, we recommended that the parameter *stage1iter* in Algorithm 1 can be chosen freely from the interval $[10, \textit{maxiter}]$.[3] To support this claim with some practical evidence, we vary the values of *stage1iter* and test the proposed algorithms on Georgia Tech face data set (see Section 4.4.1 for description) and compare the results with other algorithms. The results depicted in Fig. 1 show that each of the different values of *stage1iter* provides a version of α-HPNMF that outperforms the α-PNMF method.

[2] $\texttt{rand}(m, k)$ is a MATLAB command for generating an $m \times k$ random matrix containing pseudo random values drawn from the standard uniform distribution on the open interval (0,1).

[3] In all of our figures the label 'Number of iterations' represents the interval $[1, \textit{maxiter}]$.



(a) α-HPNMF and α-PNMF for $\alpha = 0.5$: Hellinger distance

(b) α-HPNMF and α-PNMF for $\alpha = 2$: Pearson distance

Fig. 1. Behavior of Algorithms 1 on Georgia Tech face data set for $k = 25$ and different values of *stage1iter*.

4.3. Orthogonality and sparsity

It is worth mentioning that approximate orthogonality for the basis matrix W in NMF can be enforced by adding a constraint in the form equivalent to $\|W^\top W - I\|_F$, thereby forming an orthogonal NMF problem. Hence, the degree of orthogonality depends on the specific algorithm used to solve the orthogonal NMF problem. As a consequence of satisfying the approximate orthogonality condition, the matrix W should have a large number of zero entries and become very sparse. Moreover, orthogonality guarantees less overlap among basis elements, hence minimizing the redundancy of information. In fact, the introduction of orthogonality as a constraint in NMF enables to obtain sparse factors which are crucial in many applications such as clustering, image analysis and text mining [9,10].

4.4. Image analysis and feature extraction

In this section, we illustrate the numerical results obtained from applying the new and the existing algorithms on four well-known face data sets. First, we give a brief description of each of these data sets.

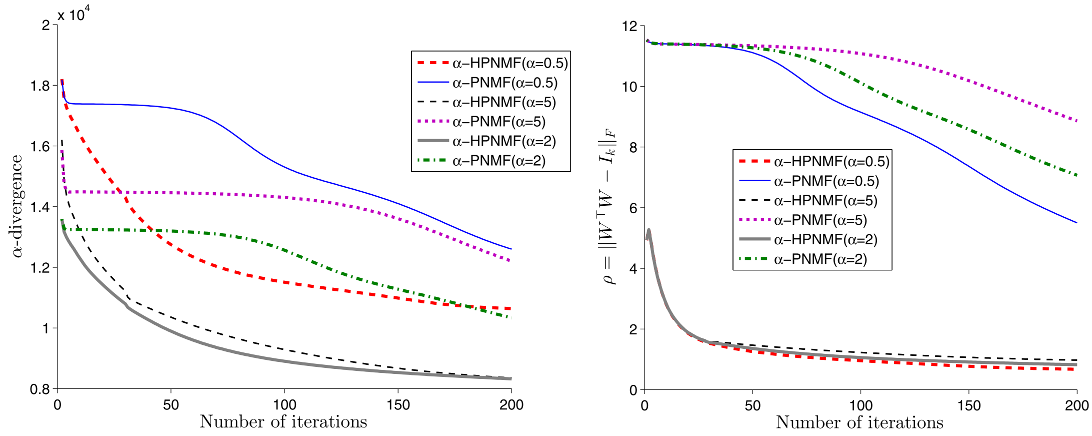
4.4.1. Description and preparation of data sets

The following data sets which are basically photographs of faces of different persons taken at different times by varying the lighting, facial expressions and other conditions were used for conducting the experiments regarding image analysis and local feature extraction. For each data set a preprocessing phase aimed at histogram-equalizing and normalizing the images has been done. In addition, the pixel values of each image were stacked into a column vector so that each image is represented by a column of the data matrix.

- The *Cambridge ORL face database* [11] consists of 400 images of 40 different subjects in PGM format. Each subject has 10 images of size 112 × 92 which are taken at different times by varying the lighting, the facial expressions (open/closed eyes, smiling/not smiling) and the facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). For the sake of computational convenience each image has been resized to 25 × 25 pixels. With the concatenation of the pixels of each image into a column we obtain a data matrix of size $m \times n = 625 \times 400$.
- The *MIT CBCL face database* [12] contains 2429 faces that has been used extensively at the Center for Biological and Computational Learning (CBCL) at MIT. Each face has $19 \times 19 = 361$ pixels. With the concatenation of the pixels of each image into a column we obtain a data matrix of size $m \times n = 361 \times 2429$.
- The *Georgia Tech face database* [13] contains 750 images of faces of 50 different individuals stored in JPG format. For each individual, there are 15 color JPEG images with cluttered background taken at a resolution of 640 × 480 pixels. The pictures show frontal and/or tilted faces with different facial expressions, lighting conditions and scale. Each image is manually labeled to determine the position of the face in the image. For the sake of computational convenience each image was converted into a gray scale image and resized to a resolution of 16 × 16 pixels. With the concatenation of the pixels of each image into a column we obtain a data matrix of size $m \times n = 256 \times 750$.
- The *Yale face database* [14] contains 165 grayscale images of faces in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, without glasses,

Table 2
Summary of used facial image datasets.

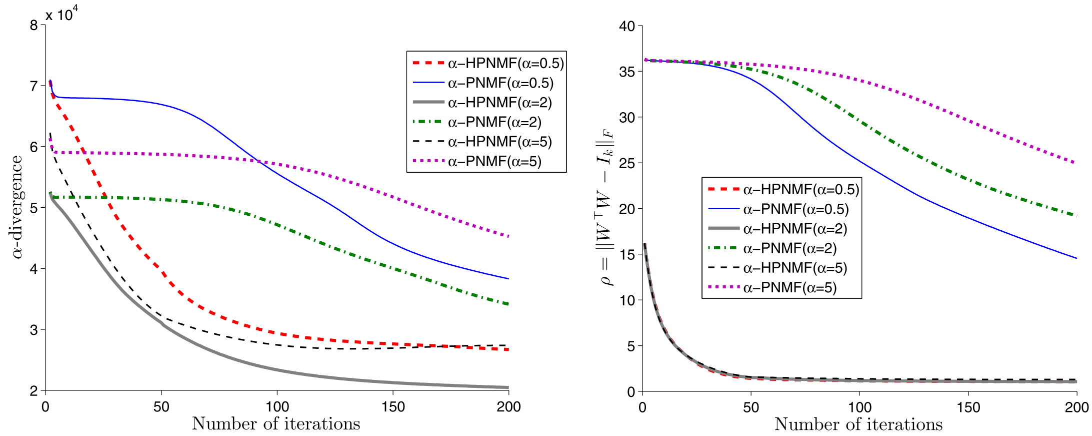
Dataset	# pixels	m	n
Cambridge ORL	25×25	625	400
MIT CBCL	19×19	361	2429
Georgia Tech	16×16	256	750
Yale	25×25	625	165



(a) Error

(b) Orthogonality as measured by $\rho = \|W^\top W - I_k\|_F$

Fig. 2. Cost function (α-divergence) on the ORL data set. Here, we used rank $k = 16$.



(a) Error

(b) Orthogonality as measured by $\rho = \|W^\top W - I_k\|_F$

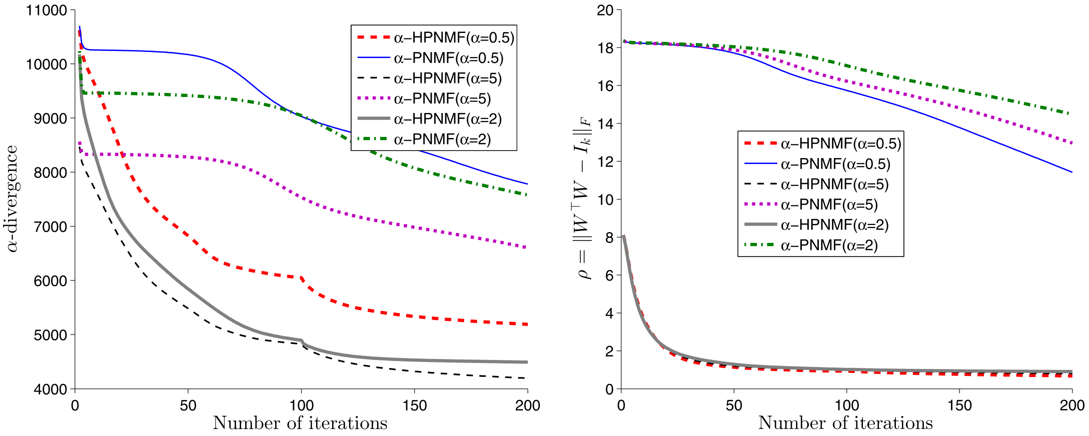
Fig. 3. Cost function (α-divergence) on the MIT data set. Here, we used rank $k = 49$.

normal, right-light, sad, sleepy, surprised, and wink. For the sake of computational convenience all the images in the database have been resized to 25×25 pixels. With the concatenation of the pixels of each image into a column we obtain a data matrix of size $m \times n = 625 \times 165$.

We summarize the data sets described above in Table 2.

4.4.2. Approximation error and orthogonality

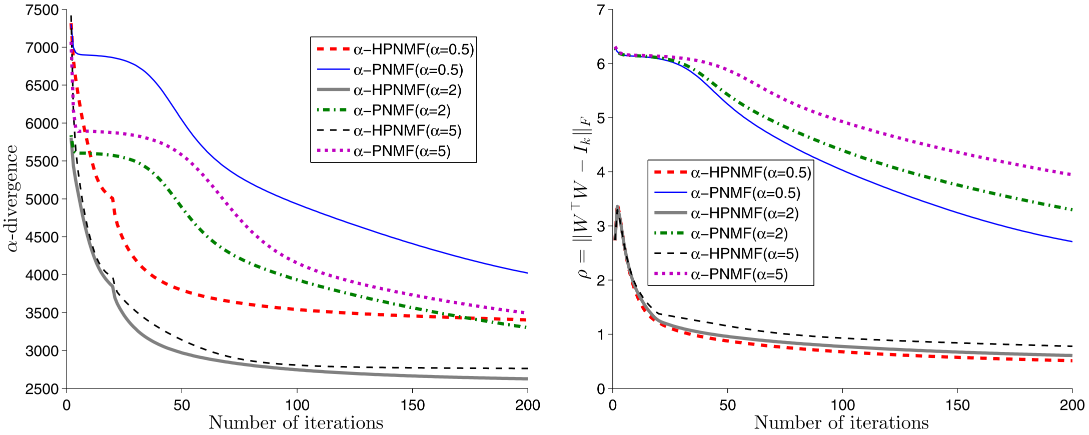
In this section, we present and analyze numerical results regarding the reconstruction accuracy of the tested algorithms and the orthogonality of their basis matrices. Algorithms that provide smaller errors (divergences) are considered to possess high level of reconstruction accuracies. This actually makes sense, since NMF aims at minimizing the error between the original data and its approximation. Hence, the smaller the error, the better the approximation is. As shown in the left sides (panels (a)) of Figs. 2–5, the errors corresponding to α-HPNMF are smaller than those of α-PNMF in all the three cases $\alpha = 0.5$, $\alpha = 2$ and $\alpha = 5$. Therefore, α-HPNMF provides factors whose product approximates the original data much better



(a) Error

(b) Orthogonality as measured by $\rho = \|W^\top W - I_k\|_F$

Fig. 4. Cost function (α-divergence) on the Georgia Tech data set. Here, we used rank $k = 25$.



(a) Error

(b) Orthogonality as measured by $\rho = \|W^\top W - I_k\|_F$

Fig. 5. Cost function (α-divergence) on the Yale data set. Here, we used rank $k = 9$.

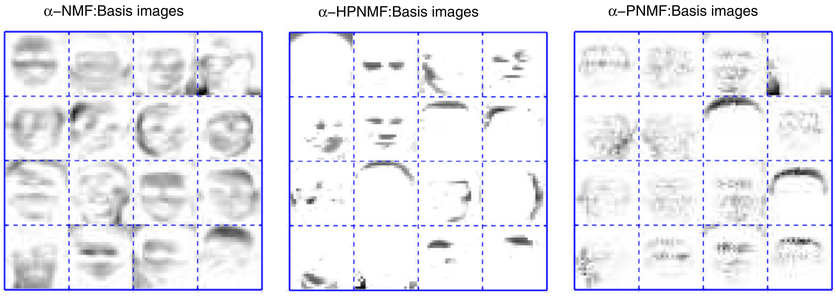
than those of α-PNMF. As a result, the new hybrid algorithms (especially for $\alpha = 2$) have higher reconstruction accuracies than their counterparts.

To compare the orthogonality of the basis factors of the tested algorithms, we used the ρ measure (6). The right sides (panels (b)) of Figs. 2–5 are the orthogonality plots corresponding to α-HPNMF and α-PNMF for different values of α. The ρ-plots in the aforementioned figures signify that the factors obtained from α-HPNMF have highly orthogonal columns—because of the corresponding smaller values of ρ—than those of α-PNMF. For this reason, the new hybrid algorithms outperform their counterparts by providing highly orthogonal and very sparse basis factors. As a consequence, the factors of α-HPNMF (especially for $\alpha = 2$) guarantee minimum overlaps among basis images and hence avoid redundancy of information.

4.4.3. Basis images, local feature extraction and face reconstruction

In this section, we apply the methods α-HPNMF, α-PNMF and α-NMF on the data sets of faces described in Section 4.4.1 and present the images of their basis matrices. All the basis matrices are of size $m \times k$. Each column is reshaped into a matrix of size $\sqrt{m} \times \sqrt{m}$ and then the resulting k matrices are arranged into a block matrix of size $\sqrt{mk} \times \sqrt{mk}$.

One can see from the numerical results depicted in Figs. 6, 8, 10 and 12 that the new algorithm, α-HPNMF (especially for $\alpha = 2$), managed to learn better localized features and is able to provide very sparse basis images which are clearly representatives of facial parts. On the contrary, α-NMF and α-PNMF fail to learn localized facial parts and their basis matrices are dense and they certainly represent full faces; especially those of α-NMF. As to the reconstructed faces, our algorithms are the runner ups while α-NMF, the one with dense factors is the best. This obviously indicates that there is a trade-off between learning highly orthogonal local features and reconstruction accuracy (Figs. 7, 9, 11 and 13).



(a) α-NMF (b) α-HPNMF (c) α-PNMF

Fig. 6. Basis images, ORL data set for $\alpha = 0.5$ and $k = 16$.

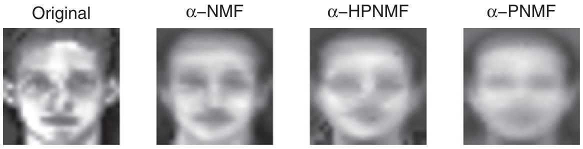
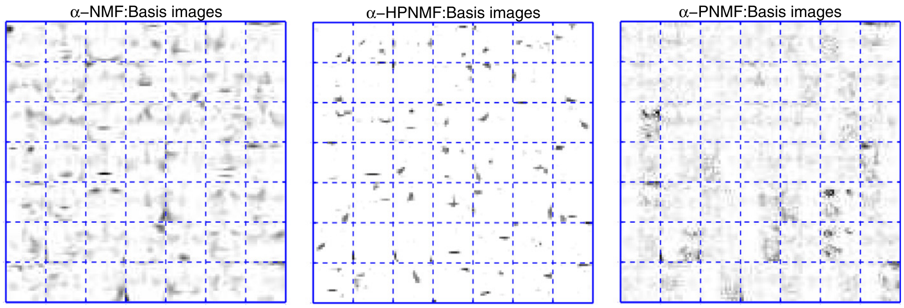


Fig. 7. Face reconstruction, ORL data set using $\alpha = 0.5$ and $k = 16$.



(a) α-NMF (b) α-HPNMF (c) α-PNMF

Fig. 8. Basis images, MIT data set for $\alpha = 5$ and $k = 49$.

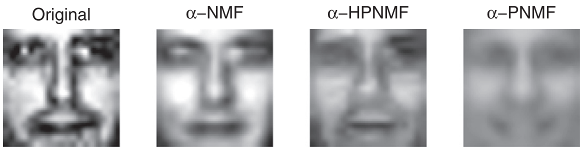
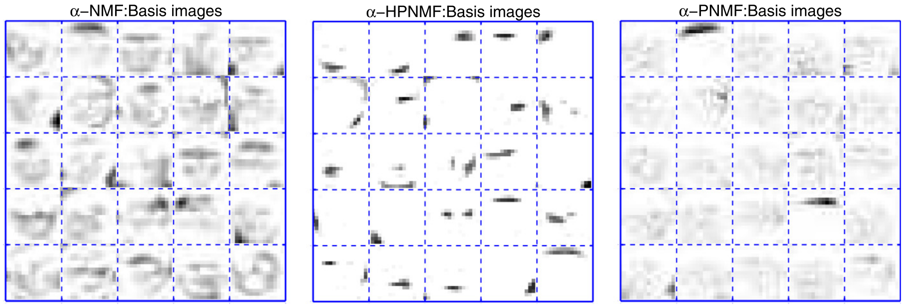


Fig. 9. Face reconstruction, MIT data set using $\alpha = 2$ and $k = 49$.



(a) α-NMF (b) α-HPNMF (c) α-PNMF

Fig. 10. Basis images, Georgia Tech data set for $\alpha = 2$ and $k = 25$.

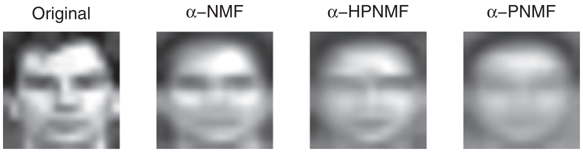
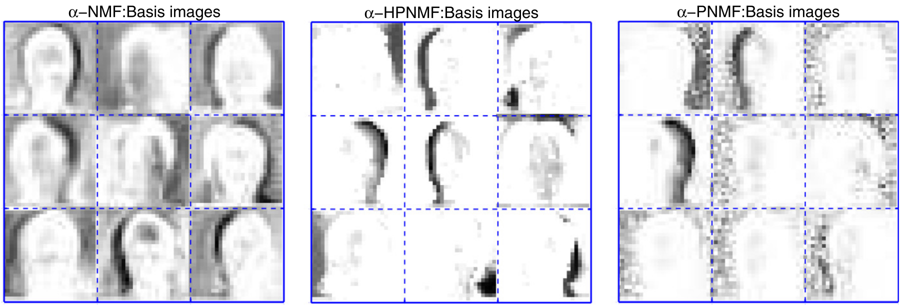


Fig. 11. Face reconstruction, Georgia Tech data set using $\alpha = 2$ and $k = 25$.



(a) α-NMF

(b) α-HPNMF

(c) α-PNMF

Fig. 12. Basis images, Yale data set for $\alpha = 2$ and $k = 9$.

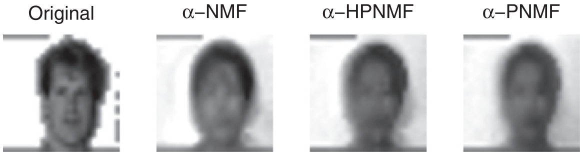


Fig. 13. Face reconstruction, Yale data set using $\alpha = 2$ and $k = 9$.

4.4.4. Sparsity, orthogonality and entropy

This section is devoted to comparing the performances of the various α-divergence based algorithms on the basis of *sparsity, orthogonality* and *entropy*. We start off by defining some terms; namely, *Hoyer's sparseness*, the τ*-measure*, and *average entropy*:

$$\textit{Hoyer's sparseness } (\mathbf{v}) = \frac{\sqrt{mk} - \left(\sum_i |\mathbf{v}_i|\right)/\sqrt{\sum_i \mathbf{v}_i^2}}{\sqrt{mk} - 1}, \tag{7}$$

$$\tau = 1 - \frac{\|\mathrm{Nr}_W^\top \mathrm{Nr}_W - I_k\|_F}{k(k-1)}, \tag{8}$$

$$\textit{average entropy} = \frac{1}{k}\sum_{j=1}^{k}\left(-\sum_{i=1}^{m} w_{ij}\log w_{ij}\right), \ \|W(:,j)\|_2 = 1 \ \forall j, \tag{9}$$

where $\mathbf{v}_i$ is the ith component of $\mathbf{v}$, $\mathbf{v} = W(:)$ vectorizes W and $\mathrm{Nr}_W = \texttt{normc}(W)$ is a MATLAB command which takes W as an input and normalizes the columns to a length of one.

According to [15], *Hoyer's sparseness* ranges in between 0 and 1. The sparsest possible vectors, those having only one nonzero entry, has the extreme value 1. The other extreme value, 0, is associated with the densest possible vectors, those where all entries have the same value. The impressive performance of the α-HPNMF algorithm is advocated by the corresponding large values of *Hoyer's sparseness*. Therefore, we can conclude that the new hybrid algorithms are capable of generating sparser factors than the other methods.

We also used the τ*-measure* as defined in (8) to measure the orthogonality of the basis matrices generated by the new hybrid algorithms and the other tested algorithms. As stated in [6], values of τ closer to 1 are indications of a high level of orthogonality. Maintaining approximate orthogonality among the columns of a basis matrix is very important in the tasks of acquiring sparse factors that ensure minimum overlap among basis images. The values of τ corresponding to our proposed algorithm, namely α-HPNMF, are shown to be much higher and closer to 1 than those of α-NMF and α-PNMF. This witnesses that the new algorithms are capable of providing highly orthogonal and sparser factors, thereby generating highly independent basis elements and saving a great deal of storage space than their counterparts.

We used (9) to quantify the *entropy* of the various α-divergence-based algorithms. As mentioned in [7], *entropy* is a measure of randomness. Here, we note that algorithms that score small *entropy* values enable the extraction of more localized

Table 3
Comparison of performances (in terms of time, sparsity, orthogonality and entropy) of the new and the existing α-divergence based algorithms. The maximum number of iterations was set to 200 for all data sets and the rank k was set to 16, 49, 25 and 9 for ORL, MIT, Georgia Tech (GT) and Yale data sets, respectively. In each row the best results are highlighted in bold.

Data	Measure	α-HPNMF		α-PNMF		α-NMF	
		$\alpha = 2$	$\alpha = \frac{1}{2}$	$\alpha = 2$	$\alpha = \frac{1}{2}$	$\alpha = 2$	$\alpha = \frac{1}{2}$
ORL	Time	27.30	41.70	25.55	39.66	**7.15**	28.48
	Hoyer's sp.	**0.71**	0.69	0.47	0.31	0.34	0.30
	τ	**0.99**	**0.99**	0.97	0.96	0.96	0.96
	Av. entropy	**17.16**	19.25	45.07	57.42	50.36	53.54
MIT	Time	54.21	102.54	51.61	99.70	**34.12**	108.55
	Hoyer's sp.	**0.84**	0.83	0.64	0.41	0.55	0.48
	τ	**0.99**	**0.99**	**0.99**	0.98	**0.99**	**0.99**
	Av. entropy	**4.29**	5.20	23.57	35.93	22.60	26.82
GT	Time	11.18	21.28	9.95	20.12	**6.20**	22.21
	Hoyer's sp.	**0.78**	0.77	0.47	0.37	0.41	0.37
	τ	**0.99**	**0.99**	0.97	0.97	0.98	0.98
	Av. entropy	**5.86**	6.52	26.29	29.96	24.19	26.49
Yale	Time	14.82	20.38	13.60	16.71	**3.19**	6.50
	Hoyer's sp.	**0.64**	0.62	0.49	0.41	0.28	0.25
	τ	**0.99**	0.98	0.95	0.94	0.93	0.92
	Av. entropy	**22.41**	24.95	39.23	47.66	55.39	58.27

Table 4
Summary of used datasets from the UCI repository.

Dataset	Classes	m	n
Iris	3	150	4
Ecoli	7	327	7
Pima	2	768	8

features and ensure the rise of more “orthogonal” and sparser basis factors. One can observe from the tabular results in Table 3 that the *entropy* values corresponding to the proposed new hybrid algorithms are smaller than those of α-NMF and α-PNMF. Hence, one can say that the α-HPNMF algorithms (especially for $\alpha = 2$) are capable of extracting better localized basis features than the other methods.

4.5. Data clustering

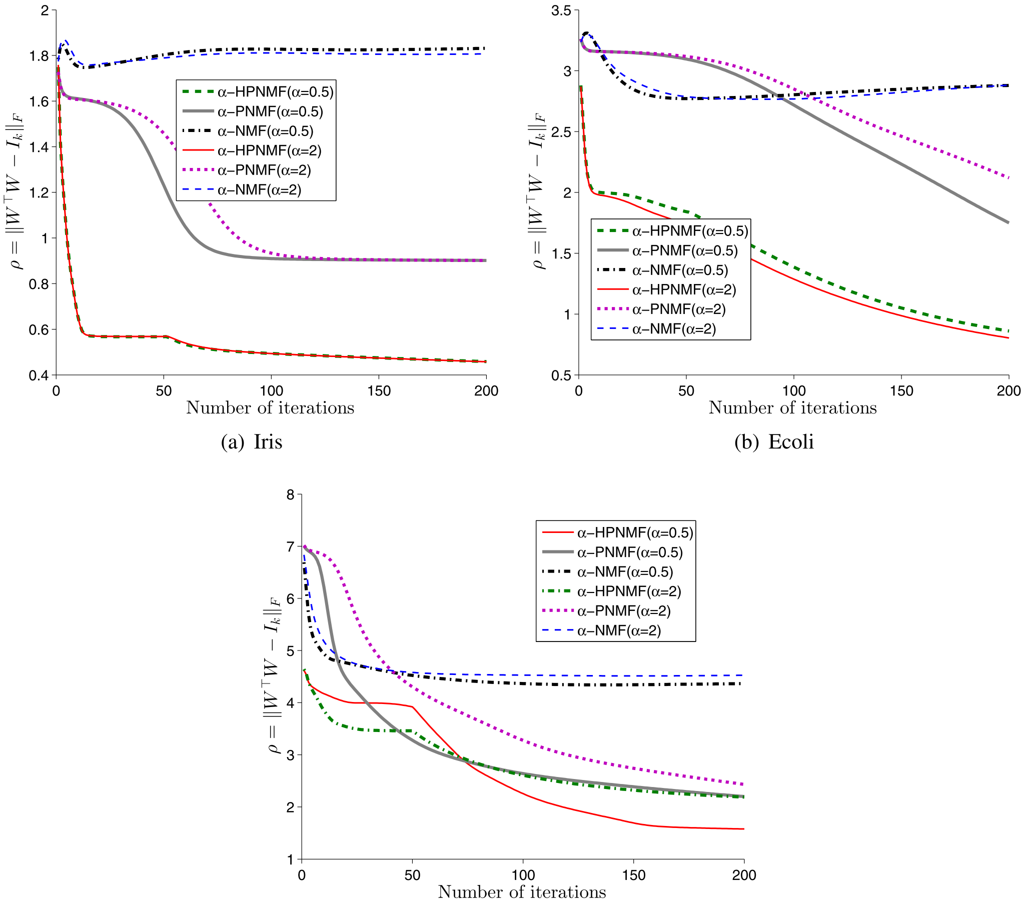
In this section we analyze the numerical results obtained from applying the new and existing α-divergence-based algorithms on three real-world data sets from the UCI repository. First, we present the description of these data sets.

4.5.1. Description of data sets

The following data sets obtained from the University of California at Irvine (UCI) machine learning repository were used for conducting experiments about data clustering.

- The *Iris plants data set (Iris)* [16] is a data set which contains $m = 150$ instances of $n = 4$ positive-real-valued attributes. The samples belong to three iris classes, ‘Setosa’, ‘Versicolour’, and ‘Virginica’, each consisting of 50 instances; altogether they form a $m \times n = 150 \times 4$ data matrix Y. Here rank $k = 3$ was used.
- The *Ecoli data set (Ecoli)* [17] contains $m = 327$ protein samples of the *Escherichia coli (E. coli)* bacteria categorized in $n = 7$ different classes. The values (positive-real-valued attributes between 0 and 1) are arranged in the form of a $m \times n = 327 \times 7$ data matrix Y. For convenience only the samples coming from the first 5 largest classes were considered. We also used $k = 5$.
- The *Pima Indian diabetes data set (Pima)* [18] contains some information about diabetes patients who are of Pima Indian heritage. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females who are at least 21 years of age. There are two classes in this data set which are labeled as ‘1’ and ‘0’, where ‘1’ means a positive test for diabetes and ‘0’ stands for a negative test. There are a total of eight clinical findings for the 268 cases in class ‘1’ and the 500 cases in class ‘0’. The sample values are arranged in a matrix of size $m \times n = 768 \times 8$. Usually the feature subspace dimension (i.e., k) is assigned the same value as the number of classes, but for the sake of computational convenience we set the former to $k = 10$ for this particular data set.

We summarize the data sets described above in Table 4.



(a) Iris

(b) Ecoli

(c) Pima

Fig. 14. Comparison of the orthogonality of the basis factors of the various α-divergence based algorithms discussed in this paper. We plotted the values of the ρ-measure defined by $\rho = \|W^\top W - I\|_F$, $\|W(:, j)\|_2 = 1$, $\forall j$. The smaller the ρ, the higher the orthogonality is.

4.5.2. Orthogonality and clustering

Empirical and theoretical studies [9,15,19] show that NMF has clear clustering effects and NMF with orthogonality constraints is equivalent to k-means clustering. Once again we used the ρ-measure defined in Section 4 to measure and compare the orthogonality of the basis matrices provided by α-HPNMF, α-NMF and α-PNMF. The experimental results in Fig. 14 indicate that α-HPNMF manages to achieve ρ values which are quite smaller than those of α-NMF and α-PNMF on all data sets. This implies that the proposed new hybrid algorithms outperform their counterparts by providing approximately orthogonal basis factors, keeping the overlap between the basis columns to a minimum and giving rise to sparser factors. Hence, we can conclude that α-HPNMF (especially for $\alpha = 2$) possesses a much better clustering performance than α-NMF and α-PNMF.

4.5.3. Sparsity, purity and entropy

We used *purity* and *entropy* as defined in [20,21] to compare the clustering performances of several α-divergence-based NMF algorithms. In addition, the measurement *Hoyer's sparseness*, as defined in (7), is employed to compare the sparsity of the basis factors corresponding to these methods. First, we give the definitions of *purity* and *entropy*:

$$purity = \frac{1}{n}\sum_{i=1}^{k}\max_{1\le j\le q} n_i^j, \tag{10}$$

$$entropy = -\frac{1}{n\log_2 q}\sum_{i=1}^{k}\sum_{j=1}^{q} n_i^j \log_2 \frac{n_i^j}{n_i}, \tag{11}$$

where q is the number of classes, n_i^j is the number of samples in the cluster i that belong to the original class j and $n_i = \sum_j n_i^j$.

Table 5
Comparison of performances (in terms of sparsity, purity and entropy) of the tested algorithms: α-NMF, α-PNMF and α-HPNMF. We set *maxiter* to 200 and recorded the mean results of the measurements for 100 different random initializations. The best results are highlighted in bold.

Data	Measure	α-HPNMF		α-PNMF		α-NMF	
		$\alpha = 2$	$\alpha = \frac{1}{2}$	$\alpha = 2$	$\alpha = \frac{1}{2}$	$\alpha = 2$	$\alpha = \frac{1}{2}$
Iris$_{k=3}$	Hoyer	**0.39**	**0.39**	0.34	0.33	0.17	0.17
	Purity	**0.81**	**0.81**	0.72	0.72	0.78	0.76
	Entropy	**0.35**	**0.35**	0.40	0.40	0.40	0.41
Ecoli5$_{k=5}$	Hoyer	**0.48**	0.47	0.35	0.32	0.17	0.17
	Purity	0.71	**0.73**	0.62	0.62	0.68	0.68
	Entropy	0.41	**0.39**	0.54	0.54	0.47	0.48
Pima$_{k=10}$	Hoyer	**0.66**	0.62	0.60	0.57	0.37	0.37
	Purity	0.65	0.65	0.65	0.65	0.65	0.65
	Entropy	0.27	0.27	**0.26**	**0.26**	**0.26**	0.27

In NMF and other multivariate data analysis techniques *purity* and *entropy* are used quite often to compare the clustering performances of different algorithms for the reason that they provide fair comparison. In fact, they quantify clustering performances by using ground truth class information irrespective of the assumptions of the cluster distributions as well as the type of algorithm being used [8]. *Entropy*, whose values range in between ‘0’ and ‘1’, measures how the various classes are distributed within each cluster. Perfect clustering, a clustering that leads to clusters that contain information from only a single class, is attained when the value of the *entropy* is ‘0’. In general, small *entropy* values (those close to 0) are associated with better clustering performances [21]. The other useful clustering performance measure is *purity* whose values also lie in the interval [0,1], with ‘1’ standing for perfect clustering. Usually, large *purity* values (those close to 1) indicate remarkable achievement in clustering [8,20].

In Table 5, we presented the mean results of the various measurements defined in (7), (10), and (11) for 10 different random initializations. As reported in this table, the new algorithms score the highest values on *Hoyer’s sparseness* measure for all the data sets, thereby guaranteeing the sparsest basis factors and saving a great deal of storage space.

It is mentioned earlier that *purity* is one of the measures commonly used for comparing clustering performances of different algorithms under the notion that the higher the *purity* value, the better the clustering performance is. According to the results recorded in Table 5, the proposed new hybrid algorithms score the highest *purity* values on 2 out of 3 data sets while all algorithms have an identical score on the third one, namely the Pima data set. In general, we can say that α-HPNMF has better clustering performance than α-NMF and α-PNMF (as evidenced by the corresponding high *purity* values).

The *Entropy* measure, defined by (11), is also used to assess the clustering performances of the various algorithms discussed in this paper. By referring to the results in Table 5, we can see that the *entropy* values corresponding to the new hybrid algorithm (α-HPNMF) are smaller than those of α-NMF and α-PNMF for 2 out of 3 data sets. Therefore, we can conclude that α-HPNMF has a much better clustering performance on the majority of the data sets than its counterparts.

5. Conclusion

In this paper, we proposed a new class of generalized hybrid algorithm called α-HPNMF, by combining the basic ALS algorithm with the multiplicative update rule of the α-PNMF method. Different kinds of measures including *Hoyer’s sparseness*, the τ*-measure, purity* and *entropy* were employed to assess the performances of the new and several existing algorithms. For feature extraction and image analysis, we used four well-known data sets of faces and came to realize that the new hybrid algorithm is capable of extracting much better localized facial parts and providing highly orthogonal and sparser basis factors than the other algorithms. We also realized that the choice $\alpha = 2$ for the α-HPNMF algorithm provides better results. The clustering performances of the new and existing α-divergence based algorithm was tested on three commonly used data sets from the UCI repository. The numerical experiments on the UCI data sets also confirm that the proposed new hybrid algorithm exhibits higher clustering performances, give rise to approximately orthogonal and highly sparse basis factors, and guarantee less overlap among basis elements, thereby minimizing the redundancy of information to a greater extent.

References

[1] D. Lee, S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 556–562.

[2] D. Lee, S. Seung, Algorithms for non-negative matrix factorization, Adv. Neural Inf. Process. Syst. 13 (2001) 788–791.

[3] M. Berry, M. Browne, A. Langville, V. Pauca, R. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization, Comput. Stat. Data Anal. 52 (2004) 155–173.

[4] A. Cichocki, R. Zdunek, A. Phan, S. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation, Wiley, UK, 2009.

[5] A. Cichocki, H. Lee, Y. Kim, S. Choi, Nonnegative matrix factorization with α-divergence, Pattern Recognit. Lett. 29 (2008) 1433–1440.

[6] Z. Yang, E. Oja, Projective Nonnegative Matrix Factorization Based on α-Divergence, 2009.

[7] Z. Yuan, Advances in Independent Component Analysis and Nonnegative Matrix Factorization, Ph.D. thesis, Helsinki University of Technology, Dissertations in Information and Computer Sciene, Finland, 2009.

[8] Z. Yang, E. Oja, Linear and nonlinear projective nonnegative matrix factorization, IEEE Trans. Neural Netw. 21 (2010) 734–749.

[9] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix tri-factorizations for clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Springer, 2006, pp. 126–135.

[10] F. Pompili, N. Gillis, P.-A. Absil, F. Glineur, Two algorithms for orthogonal nonnegative matrix factorization with application to clustering, Neurocomputing 141 (2014) 15–25.

[11] F. Samaria, A. Harter, Parameterization of a stochastic model for human face identification, in: Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, Sarasota Florida, 1994, pp. 138–142.

[12] K. Sung, Learning and Example Selection for Object and Pattern Recognition, Ph.D. thesis, MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Cambridge, MA, 1996.

[13] A. Nefian, A Hidden Markov Model Based Approach for Face Detection and Recognition, Georgia Institute of Technology, Atlanta, 1999 Ph.D. thesis.

[14] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 711–720.

[15] P. Hoyer, Nonnegative matrix factorization with sparseness constraints, J. Mach. Learn. Res. 5 (2004) 1457–1469.

[16] R. Fisher, The use of multiple measurements in taxonomic problems, Annu. Eugen. 7 (1936) 179–188.

[17] P. Horton, K. Nakai, A probablistic classification system for predicting the cellular localization sites of proteins, in: Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology, 1996, pp. 109–115.

[18] J. Smith, J. Everhart, W. Dickson, W. Knowler, R. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the 1988 Symposium on Computer Applications and Medical Care, IEEE Computer Society Press, 1988, pp. 261–265.

[19] S. Li, X. Hou, H. Zhang, Q. Cheng, Learning spatially localized, parts-based representation, in: Proceedings of the 2001 IEEE Computer Vision and Pattern Recognition, IEEE, 2001, pp. 207–212.

[20] H. Kim, H. Park, Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares, Bioinformatics 23 (2007) 1495–1502.

[21] P. Zhao, G. Karypis, Empirical and theoretical comparisons of selected criterion functions for document clustering, Kluwer Academic Publishers, Mach. Learn. 55 (2004) 311–331.