

# Evolutionary Nonnegative Matrix Factorization for Data Compression

Liyun Gong<sup>1</sup>, Tingting Mu<sup>2</sup>, and John Y. Goulermas<sup>2</sup>

1. School of Computer Science and School of Engineering, The University of Lincoln, Lincoln, UK, LN6 7TS

2. Electrical Engineering, Electronics and Computer Science, The University of Liverpool, Brownlow Hill, Liverpool, UK, L69 3GJ  
lgong@lincoln.ac.uk; {t.mu; j.y.goulermas}@liverpool.ac.uk

**Abstract.** This paper aims at improving non-negative matrix factorization (NMF) to facilitate data compression. An evolutionary updating strategy is proposed to solve the NMF problem iteratively based on three sets of updating rules including multiplicative, firefly and survival of the fittest rules. For data compression application, the quality of the factorized matrices can be evaluated by measurements such as sparsity, orthogonality and factorization error to assess compression quality in terms of storage space consumption, redundancy in data matrix and data approximation accuracy. Thus, the fitness score function that drives the evolving procedure is designed as a composite score that takes into account all these measurements. A hybrid initialization scheme is performed to improve the rate of convergence, allowing multiple initial candidates generated by different types of NMF initialization approaches. Effectiveness of the proposed method is demonstrated using Yale and ORL image datasets.

**Keywords:** Non-negative matrix factorization; data compression, evolutionary computation

## 1 Introduction

Non-negative matrix factorization (NMF) is an algorithm based on decomposition by parts the input data matrix, which can reduce the dimensionality of the datasets while keep the most information about the datasets. It is suitable for redundancy reduction in image data, known as image compression, to optimize storage space and increase transmission rate. Different from compression methods such as principal component analysis (PCA) and independent component analysis (ICA), NMF introduces non-negative constraints which offer more clear interpretation. Recent advances on NMF are focused on formulating more sophisticated objective function for NMF to better serve a dimensionality reduction, clustering or classification task, by incorporating extra concerns into the original reconstruction error such as preservation of local data geometry [12] and enhancement of class separation[5].

In traditional NMF, the initial values of the factorization variables of NMF are usually set as random values. However, this is not the most effective setup. Many algorithms have been proposed to obtain the initial values of the factorization variables in a more sophisticated way, in order to improve the rate of convergence. For example, spherical k-means clustering is used to initialize one factorization matrix, then nonnegative least square is used to derive the other factorization matrix [8]. Principal component analysis (PCA) can also be used to perform such initialization, for which non-negativity of the factorization variables can be enforced by either converting all the negative elements of the PCA output to zero [11] or keeping the absolute values [10]. Another way is to utilize fuzzy c-means clustering (FCM), where the FCM cluster centroids can be used as one factorization matrix while the cluster membership degrees to derive the other factorization matrix [11, 7]. Performances of six initialization methods of random, centroid, singular value decomposition (SVD) centroid, random acol, random C and co-occurrence based ones is compared in [3]. When applying NMF to data compression, an appropriate initialization method has the potential to enhance both convergence rate and compression performance. On the other hand, sensitivity of the NMF performance to different initialization methods makes it challenging for the user to choose an appropriate one for a given task.

Instead of choosing one particular initialization scheme, we propose an evolutionary NMF updating procedure, which learns from multiple seed candidates initialized in the solution space, and effectively updates the candidate set along multiple directions in order to obtain better quality of factorization matrices to facilitate data compression. The proposed method is general, and there is no limitation on the used number and types of the initialization methods. The major contributions of the proposed design include to improve the the most commonly used multiplicative NMF update [4] so that it can serve better the data compression task, and to take advantage of the hybrid of different NMF initialization setups to produce NMF approximations that suits better the data compression purpose and save the users' effort on initialization determination. Effectiveness of the proposed method will be demonstrated thoroughly through benchmark testing and comparison with existing approaches using Yale and ORL image datasets.

The rest of the paper is organized as follows: Section 2 explains the proposed method. Performance evaluation and comparative analysis are conducted in Section 3 by experimenting with the proposed and state-of-the-art methods. Finally, a conclusion is drawn in Section 4.

## 2 Proposed Method

Given a  $d \times n$  non-negative matrix  $\mathbf{X} = [x_{ij}]$  with each element  $x_{ij} \geq 0$ , its columns represent images to be analyzed. NMF seeks two non-negative matrices, a  $d \times k$  one  $\mathbf{W} = [w_{ij}]$  and an  $n \times k$  one  $\mathbf{H} = [h_{ij}]$ , so that the following

factorization error is minimized:

$$\min_{\substack{w_{ij} \geq 0, \\ h_{ij} \geq 0}} \left\| \mathbf{X} - \mathbf{W}\mathbf{H}^T \right\|_F^2, \quad (1)$$

where  $\|\cdot\|_F$  denotes the Frobenius norm. Each column of  $\mathbf{W}$  is known as the basic vector, while each column of  $\mathbf{H}$  as the encoding coefficient vector. For the image compression analysis, each column of  $\mathbf{W}$  indicates one basic image and  $k \leq \min(d, n)$  is often assumed as the number of the basic images determined by the user or the specific purpose.

We propose an evolutionary strategy to improve the iterative updating procedure of NMF, named as ENMF. It aims at producing higher-quality basis and encoding coefficient matrices  $\mathbf{W}$  and  $\mathbf{H}$  to suit the data compression purpose. The algorithm starts from multiple pairs of initialization matrices for the basis and encoding coefficient matrices, which form an initial candidate set denoted as  $S_0 = \{(\mathbf{W}_0^i, \mathbf{H}_0^i)\}_{i=1}^m$  where  $\{\mathbf{W}_0^i\}_{i=1}^m$  and  $\{\mathbf{H}_0^i\}_{i=1}^m$  are referred as the seed matrices. The algorithm then evolves, creating an updated candidate set at each iteration, denoted as  $S_t = \{(\mathbf{W}_t^i, \mathbf{H}_t^i)\}_{i=1}^{m_t}$  for the  $t$ th iteration with  $m_t$  denoting the new candidate number. In the end, the optimal basis matrix and its corresponding encoding coefficient matrix are selected from the finally evolved candidate set based on a score function formulated to suit data compression.

## 2.1 Seed Matrix Generation

To take advantage of the state-of-the-art NMF initialization strategies and to achieve local improvement of the optimal solution, multiple NMF initialization approaches are utilized to construct the initial candidate set, which contains various seed matrices of the basis and encoding coefficient ones: (1) The clustering-based initialization (CI) approach is first conducted via performing k-means clustering [2]. The resulting binary cluster membership matrix is used as  $\mathbf{H}_0^1$ , and the resulting clustering centroid matrix as  $\mathbf{W}_0^1$ . (2) A similar CI approach is conducted again but via FCM clustering [1]. The obtained cluster membership and centroid matrices are used as  $\mathbf{H}_0^2$  and  $\mathbf{W}_0^2$ , respectively. (3) The random initialization (RI) [4] and random acol initialization (RAI) [3] are used to generate the two candidates of  $(\mathbf{W}_0^3, \mathbf{H}_0^3)$  and  $(\mathbf{W}_0^4, \mathbf{H}_0^4)$ . It is worth to note that the proposed NMF updating algorithm is a general method. The users can freely include any type and any number of initial candidates to suit their needs apart from the above ones.

## 2.2 Evolving Strategy

In each iteration, three new subsets of candidates  $S_{t+1}^{(M)}$ ,  $S_{t+1}^{(F)}$  and  $S_{t+1}^{(S)}$  are generated from the previous set  $S_t$ , according to three types of evolving rules proposed, including the multiplicative, firefly and the survival of the fittest rules. The three subsets together constitute the updated set  $S_{t+1} = S_{t+1}^{(M)} \cup S_{t+1}^{(F)} \cup S_{t+1}^{(S)}$  at the  $(t+1)$ th iteration, from which the best candidate is selected as the final output in the last iteration. In the following, we explain the three rules in detail.

**Multiplicative Rule:** This rule is constructed to take advantage of the classical multiplicative update rules for NMF approximation [4]. It generates the new candidate subset by

$$S_1^{(M)} = \Phi_1(S_0, \mathbf{X}), \quad (2)$$

for the first iteration and

$$S_{t+1}^{(M)} = \Phi_1(S_t^{(M)}, \mathbf{X}), \quad (3)$$

for the  $(t + 1)$ th iteration ( $t \geq 1$ ). The operation  $S' = \Phi_1(S, \mathbf{X})$  takes one set of matrix pairs  $S = \{(\mathbf{W}_i, \mathbf{H}_i)\}_{i=1}^m$  and one  $d \times n$  matrix  $\mathbf{X}$  as the input, outputs a set of matrix pairs denoted as  $S' = \{(\mathbf{W}'_i, \mathbf{H}'_i)\}_{i=1}^m$ , and is formulated as

$$\mathbf{H}'_i = \mathbf{H}_i \circ (\mathbf{X}^T \mathbf{W}_i) \oslash (\mathbf{H}_i \mathbf{W}_i^T \mathbf{W}_i), \quad (4)$$

$$\mathbf{W}'_i = \mathbf{W}_i \circ (\mathbf{X} \mathbf{H}_i) \oslash (\mathbf{W}_i \mathbf{H}_i^T \mathbf{H}_i). \quad (5)$$

where  $\circ$  and  $\oslash$  denote the Hadamard product and division, respectively. This rule updates the candidates separately from the other rules in order to enable the inclusion of multiple NMF solutions obtained by the multiplicative update rules to the final evolved candidate set. These solutions are driven by the same factorization error minimization but initialized through different ways.

**Firefly Rule:** This rule encourages the generation of new candidate matrix pairs that may contain higher quality of basic matrix than those obtained by the previous multiplicative rule, in order to facilitate the data compression task more effectively.

In the first iteration, the firefly rule operates on the candidate subset  $S_1^{(M)}$  of the multiplicative rule, and further creates another candidate subset by

$$S_1^{(F)} = \Phi_2(S_1^{(M)}, \mathbf{W}_1^*). \quad (6)$$

The operation  $S' = \Phi_2(S, \mathbf{A})$  takes a set  $S = \{(\mathbf{W}_i, \mathbf{H}_i)\}_{i=1}^m$  and an  $d \times k$  matrix  $\mathbf{A}$  as input, while outputs a new set  $S' = \{(\mathbf{W}'_i, \mathbf{H}'_i)\}_{i=1}^m$ . The corresponding relationship between its input and output is defined by

$$\mathbf{W}'_i = \mathbf{W}_i + \beta (\mathbf{A} - \mathbf{W}_i), \quad (7)$$

$$\mathbf{H}'_i = \mathbf{H}_i, \quad (8)$$

where  $0 < \beta \leq 1$  is set by the user. It is obvious that, given  $0 < \beta \leq 1$ ,  $\mathbf{W}'_i$  is always non-negative when  $\mathbf{W}_i$  and  $\mathbf{A}$  are both non-negative. These guarantee that the generated matrix pairs  $(\mathbf{W}'_i, \mathbf{H}'_i)$  are eligible to be used as NMF candidates. The matrix  $\mathbf{W}_1^*$  used in Eq. (6) is selected through searching within the combined set of  $S_1^{(M)} \cup S_0$  based on a predefined score function  $O(\cdot)$  for assessing the compression quality, which will be explained in Section 2.3.

Eq. (7) drives  $\{\mathbf{W}_i\}_{i=1}^m$  generated by the multiplicative rule to move towards a pre-selected optimal basis matrix  $\mathbf{W}_1^*$ . This design is motivated by a recent evolutionary optimization algorithm inspired by the flashing behaviour of firefly, known as firefly algorithm [9]. It assumes that attractiveness between fireflies is proportional to their brightness, thus, given any two fireflies, one will move towards the other that glows brighter. Following Eq. (7), each candidate in  $S_1^{(M)}$  is viewed as a firefly. The quality of the basic matrix for each candidate, evaluated by the score function  $O(\cdot)$ , represents the brightness degree of the firefly. The evolving rule is constructed by letting all the fireflies move towards the brightest one in each iteration. This procedure offers an opportunity to evolve higher quality of basic matrices to better serve the data compression task.

From the second iteration, the firefly rule starts to create new candidate subset  $S_{t+1}^{(F)}$  by operating on its previously generated subset  $S_t^{(F)}$ . It first modifies  $S_t^{(F)}$  by the multiplicative rule  $\Phi_1$ , and then updates the resulting set based on the firefly operation  $\Phi_2$ . This gives the following new candidate subset for the  $(t + 1)$ th iteration ( $t \geq 0$ ):

$$S_{t+1}^{(F)} = \Phi_2 \left( \Phi_1 \left( S_t^{(F)}, \mathbf{X} \right), \mathbf{W}_t^* \right), \quad (9)$$

where  $\mathbf{W}_t^*$  is selected according to the score function  $O(\cdot)$ , through searching within not only the whole previous candidate set but also its update via multiplicative rule  $S_t \cup \Phi_1(S_t, \mathbf{X})$  to maintain its quality. Instead of directly updating  $S_t^{(F)}$  with  $\Phi_2$ , Eq. (9) uses the multiplicative rule to smoothen out the given candidates, which may potentially reduce the factorization error. The mixed application of  $\Phi_1$  and  $\Phi_2$  attempts to evolve matrix pairs offering good quality of basic matrix while alternatively ensuring the joint quality of the basis and encoding coefficient matrices.

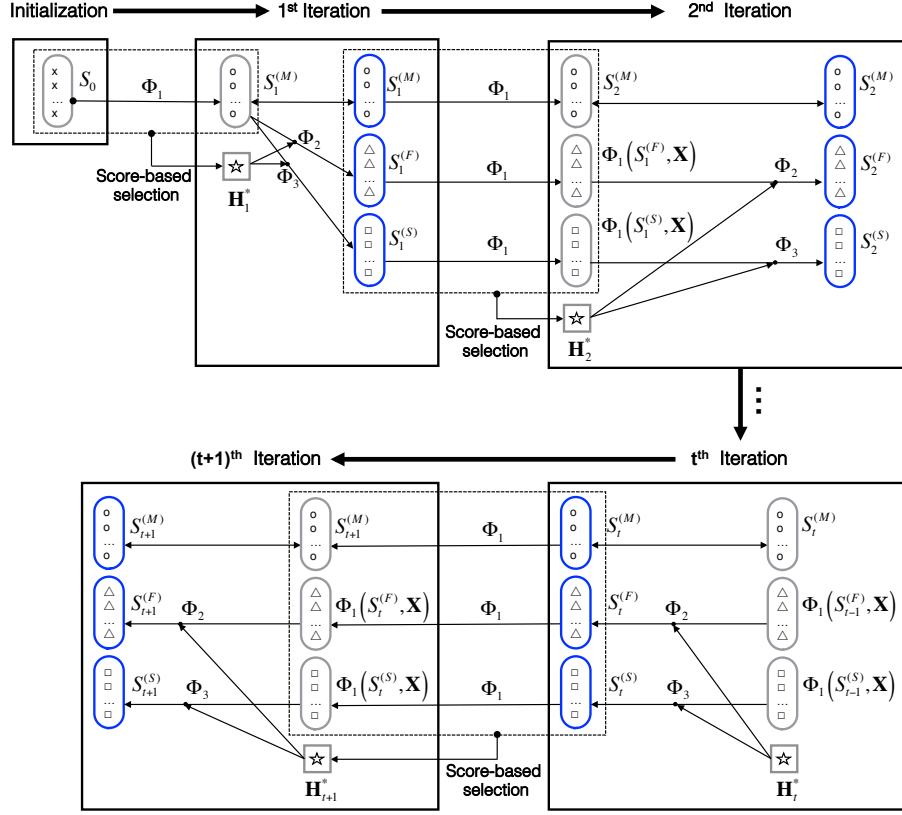
**Survival of the Fittest Rule:** This rule ensures the candidates containing the best basic matrix are always included in the evolved set. At the first iteration, the candidate subset  $S_1^{(S)}$  is generated by

$$S_1^{(S)} = \Phi_3 \left( S_1^{(M)}, \mathbf{W}_1^* \right), \quad (10)$$

After that, it modifies its previously generated subset  $S_t^{(S)}$  by

$$S_{t+1}^{(S)} = \Phi_3 \left( \Phi_1 \left( S_t^{(S)}, \mathbf{X} \right), \mathbf{W}_t^* \right). \quad (11)$$

Here, the operation  $S' = \Phi_3(S, \mathbf{A})$  creates  $m$  matrix pairs  $S' = \{(\mathbf{W}'_i, \mathbf{H}'_i)\}_{i=1}^m$  from the input set  $S = \{(\mathbf{W}_i, \mathbf{H}_i)\}_{i=1}^m$ , and is formulated as  $\mathbf{W}'_i = \mathbf{A}$  and  $\mathbf{H}'_i = \mathbf{H}_i$ . It combines the best basic matrix  $\mathbf{W}_t^*$  selected in each iteration with various encoding coefficient matrices. The use of  $\mathbf{W}'_i = \mathbf{A}$  can be viewed as a special case of Eq. (7) with the fixed parameter  $\beta = 1$ , equivalent to forcing all the weaker fireflies to eliminate themselves but let the brightest one survive. Thus, this rule is named as the survival of the fittest.



**Fig. 1.** Data flow of the proposed ENMF. The circle, triangle and rectangle symbols represent candidates derived during the generation of the  $S_t^{(M)}$ ,  $S_t^{(F)}$  and  $S_t^{(S)}$  subsets, respectively.

### 2.3 Score Function

Since the primary goal of this work is to improve NMF so that it can serve better the data compression task, it is important to design an appropriate score function to assess compression quality. Usually, in addition to factorization error, data compression performance is also indicated by sparsity and orthogonality of the resulting basis matrix. Given the data matrix  $\mathbf{X}$  and the basis matrix  $\mathbf{W}$ , the following measurements are usually computed [11]:

$$\text{error}(\mathbf{W}) = \frac{\|\mathbf{X} - \mathbf{W}\mathbf{H}_W^T\|_F}{\|\mathbf{X}\|_F}, \quad (12)$$

$$\text{sparsity}(\mathbf{W}) = \frac{\sum_{i=1}^d \sum_{j=1}^k |w_{ij}|}{k}, \quad (13)$$

$$\text{orthogonality}(\mathbf{W}) = \sqrt{\frac{d \sum_{i,j=1, i \neq j}^k \mathbf{w}_i^T \mathbf{w}_j}{k(k-1)/2}}, \quad (14)$$

where  $w_{ij}$  denotes the  $ij$ -th element of the basis matrix  $\mathbf{W}$  while the vector  $\mathbf{w}_i$  denotes its  $i$ th column, and the encoding matrix  $\mathbf{H}_W^T$  is computed from  $\mathbf{W}$  by applying non-negative least square analysis [4]. Based on the formulations of all the three measurements, the lower value they possess, the better compression quality they indicate. To take into account all these three measurements, the following composite score function is proposed, given as

$$O(\mathbf{W}) = \begin{cases} \frac{\text{sparsity}(\mathbf{W}) + \text{orthogonality}(\mathbf{W})}{2}, & \text{if } \text{error}(\mathbf{W}) \leq \alpha, \\ \frac{\text{sparsity}(\mathbf{W}) + \text{orthogonality}(\mathbf{W})}{2\beta}, & \text{otherwise.} \end{cases} \quad (15)$$

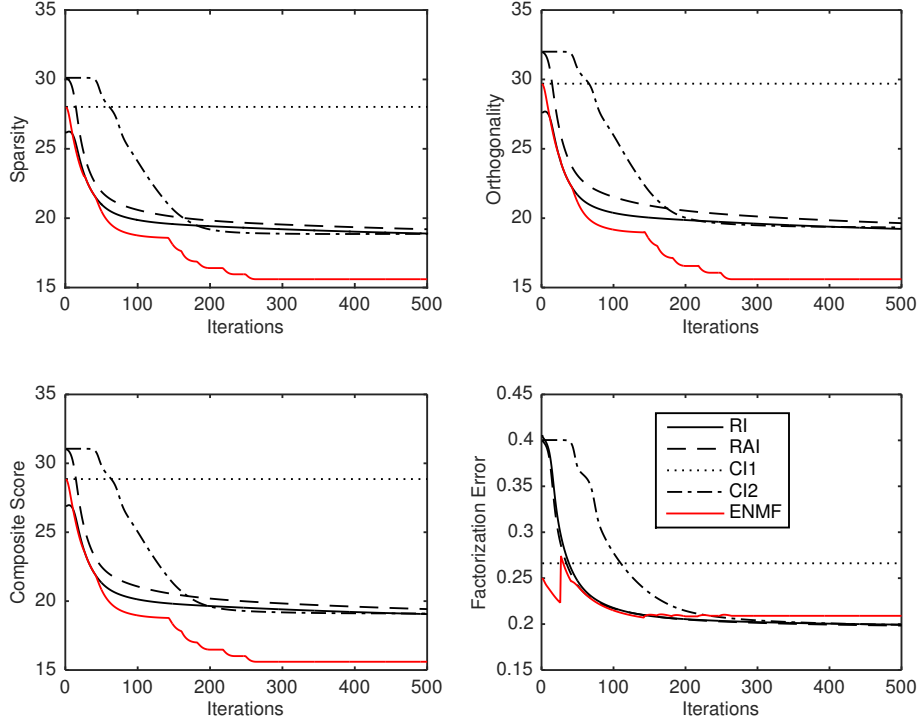
The parameter  $0 < \alpha < 1$  is defined by the user, providing a threshold of the minimum allowed factorization error rate. This proposed score examines the averaged sparsity and orthogonality performance within a pre-defined factorization error range. When the allowed error threshold  $\alpha$  is exceeded, the other two quantities are heavily penalized by  $\frac{1}{\beta}$ , where  $\beta$  can be set as a very small positive value to impose high penalty, e.g.,  $10^{-6}$ . The overall data flow of the proposed ENMF is shown in Fig. 1.

### 3 Experimental Results and Analysis

Two face image datasets of Yale and ORL are used to examine the performance. Yale contains grayscale images representing 15 subjects with 11 images per subject including center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised and wink. ORL contains 400 images representing 40 distinct subjects, with 10 images per subject taken at different times varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). Each image is scaled by its maximum pixel value so that the resulting data matrix  $\mathbf{X}$  possesses elements between 0

**Table 1.** Performance comparison for different methods.

Datasets	Sparsity	Orthogonality	Factorization Error	RAND
Yale	RI: 18.8	RI:19.2	RI:0.2	RI:91.9
	RAI:19.3	RAI:19.8	RAI:0.2	RAI:91.8
	CI1:28.1	CI1:29.8	CI1:0.3	CI1:93.1
	CI2:18.6	CI2:19.0	CI2:0.2	CI2:91.6
	ENMF: <b>15.3</b>	ENMF: <b>15.4</b>	ENMF: <b>0.2</b>	ENMF: 90.8
ORL	RI: 22.9	RI:22.9	RI:0.1	RI:95.0
	RAI:24.1	RAI:24.2	RAI:0.1	RAI:95.1
	CI1:31.0	CI1:31.5	CI1:0.2	CI1:95.1
	CI2:22.5	CI2:22.6	CI2:0.1	CI2:94.9
	ENMF: <b>20.6</b>	ENMF: <b>20.4</b>	ENMF: <b>0.1</b>	ENMF: 94.1

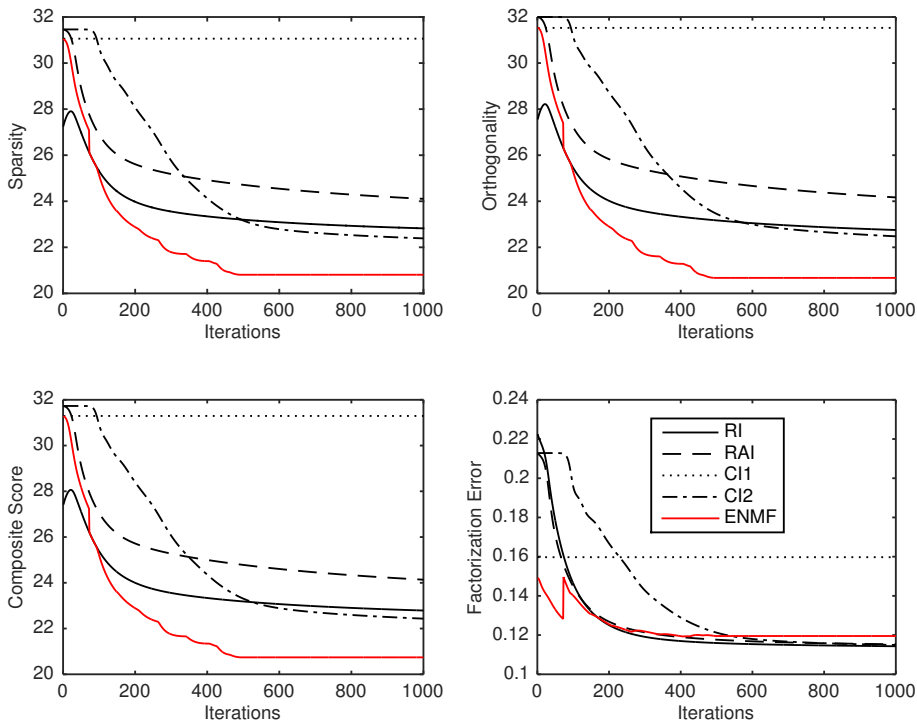


**Fig. 2.** Convergence and compression performance comparison for different methods with Yale in terms of different measurements

and 1. The images are compressed by  $k = 25$ , which was empirically observed to be sufficient for representing image data in our study. The proposed ENMF is compared with the most commonly used multiplicative NMF approach based on four different initialization methods including RI, RAI, CI by k-means (CI1) and CI by FCM (CI2). The error threshold  $\alpha$  is set as 0.3 and 0.15 for Yale and ORL, respectively, which was chosen by comparing the output sample images with the original ones given different threshold values. The maximum iteration number for ENMF is set as 500 for Yale and 1000 for ORL to allow good convergence, for which we have observed that the score measure converges approximately after 200 interactions for Yale while after 500 iterations for ORL (see Figs. 2 and 3). Each algorithm is run five times for each dataset and the averaged performance is reported. In addition to the three measurements for data compression, we also report the RAND index [6], which evaluates the quality of the encoding matrix  $\mathbf{H}$  to see how well it preserves the ground truth partition of the data.

In Figs. 2 and 3, we compare the convergence of the proposed ENMF with the competing methods, in terms of the three quality measurements of sparsity, orthogonality and factorization error, as well as the proposed composite score function. The x-axis of each plot represents the iteration number while the y-axis



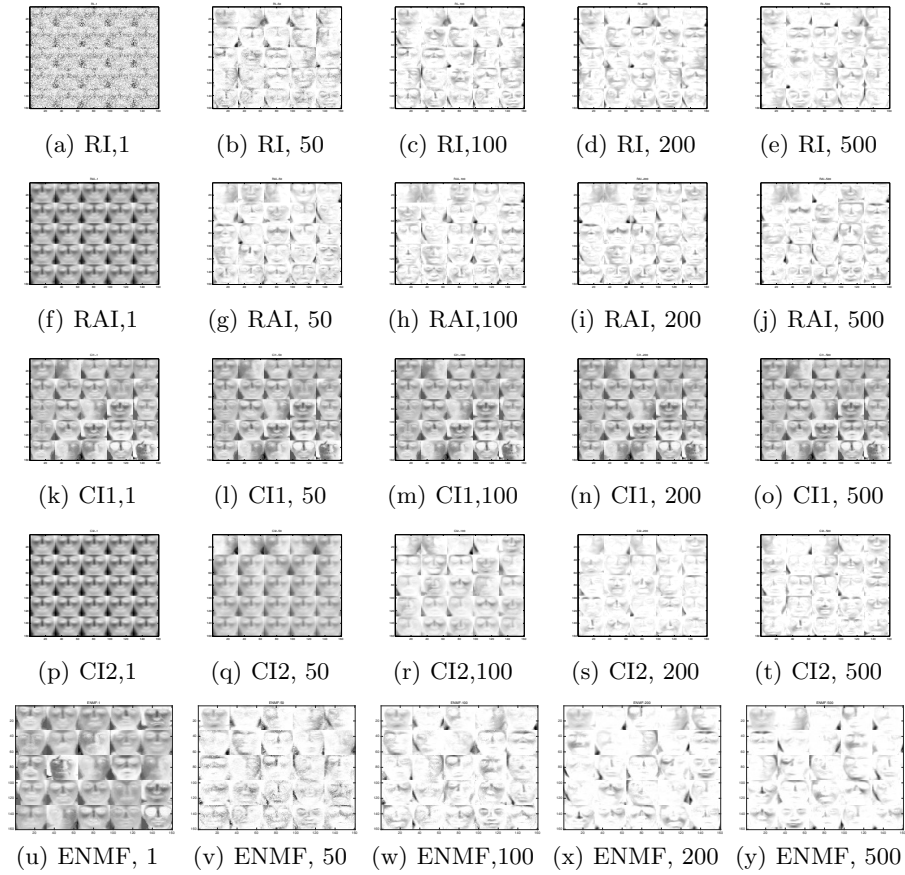


**Fig. 3.** Convergence and compression performance comparison for different methods with ORL in terms of different measurements.

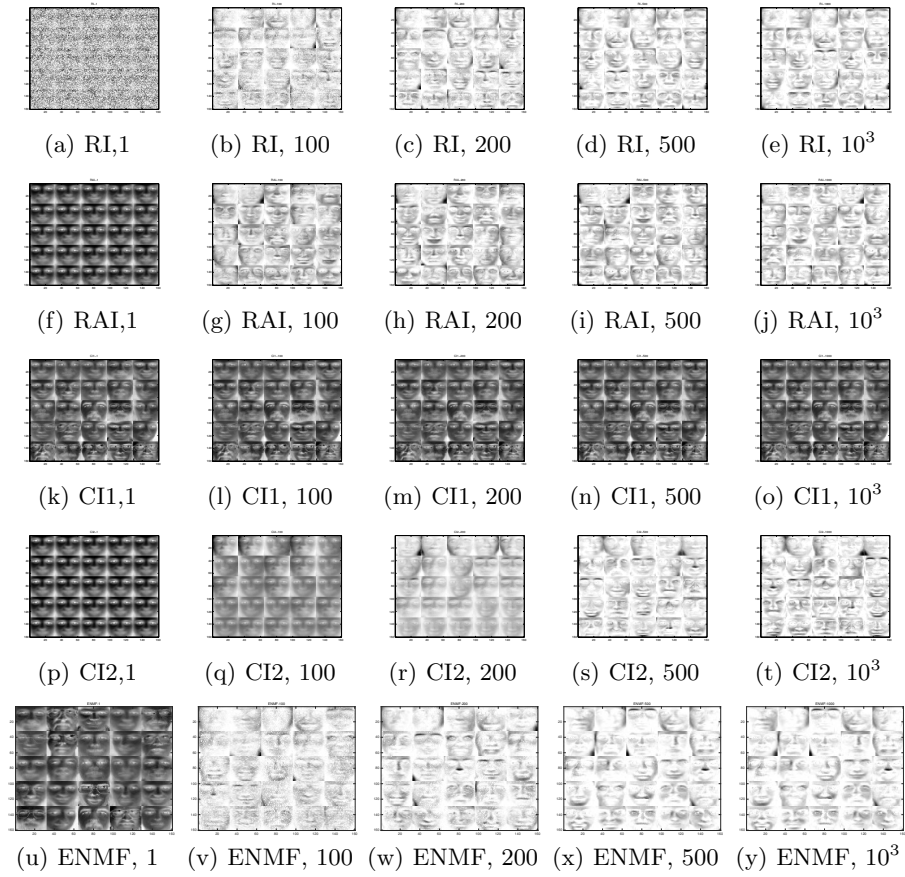
represents the values of the relevant score function as indicated in the figures. It can be seen that ENMF offers the fastest convergence and the best data compression performance including much better sparsity and orthogonality and equally low factorization error. We also provide examples of the learned basis images by different methods for Yale and ORL in Figs. 4 and 5, respectively. It can be seen that ENMF achieves higher sparsity than the others after 50 iterations for Yale and after 100 iterations for ORL. In general, the basis matrix computed by CI1 possesses pretty low sparsity. Also, observing in detail each column of the learned basis matrix that is corresponding to one patch of the  $5 \times 5$  face patches in each subfigure, ENMF offers more distinct face patches that indicate good orthogonality between the columns of the basis matrix.

Table 1 reports the averaged performance of the proposed ENMF and the competing methods by repeating the experiments five times for each method. Sparsity, orthogonality and factorization error evaluate the quality of the basis matrix  $\mathbf{W}$  that is important for data compression, while the RAND index measures the possible information loss of the encoding matrix  $\mathbf{H}$  in terms of cluster structure preservation. It can be seen that the proposed method possesses significantly better sparsity and orthogonality, meanwhile comparative factorization

error and RAND index as compared to the others. The competitive performance of the proposed method demonstrated in Fig. 2, Fig. 3 and Table 1 benefits from its three-rule-driven update procedure. It inherits good quality of candidates derived by the classical multiplicative rule, while introduces more diversified offsprings generated from the strongest parents (the brightest fireflies) to avoid local optimum but maintain competent searching direction.



**Fig. 4.** Demonstration of the basis images learned by different methods for Yale. The rows represent different methods including multiplicative NMF with RI, RAI, CI1 and CI2 initializations and ENMF from top to bottom. The columns represent the compared iteration numbers of 1, 50, 100, 200 and 500 from left to right.



**Fig. 5.** Demonstration of the basis images learned by different methods for ORL. The rows represent different methods including multiplicative NMF with RI, RAI, CI1 and CI2 initializations and ENMF from top to bottom. The columns represent the compared iteration numbers of 1, 100, 200, 500 and 1000 from left to right.

## 4 Conclusions

We have proposed a novel evolutionary updating strategy to improve NMF for data compression. The hybrid initialization takes advantages of multiple initialized seed candidates in the solution space and saves the users' effort on selection of an appropriate initialization scheme. Three sets of updating rules including multiplicative, firefly and survival of the fittest rules have been proposed driven by a composite score function for assessing data compression quality in terms of sparsity, orthogonality and factorization error. This enables effective searching of an optimal candidate along multiple directions directly controlled by the performance of the targeted data compression task. Experiments have been carried out using two image datasets of Yale and ORL and the results have shown that the proposed ENMF outperforms the most popular multiplicative NMF methods for data compression in terms of both convergence and compression performance.

## References

1. James C. Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Comput. Geosci.*, vol. 10, no. 2-3, pp.191-203, 1984.
2. E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768-769, 1965.
3. A N Langville, C D Meyer, and R Albright. Initializations for the nonnegative matrix factorization. In *Proceeding of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.
4. C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, 1974.
5. S. Nikitidis, A. Tefas, and I. Pitas. Projected gradients for subclass discriminant nonnegative subspace learning. *IEEE Transactions on Cybernetics*, 2014. In press, DOI: 10.1109/TCYB.2014.2317174.
6. W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, Theory and Methods Section*, vol. 66, pp. 846-850, 1971.
7. M Rezaei, R Boostani, and M Rezaei. An efficient initialization method for non-negative matrix factorization. *Journal of Applied Sciences*, vol. 11, pp. 354-359, 2011.
8. S Wild. Seeding non-negative matrix factorizations with the spherical k-means clustering. Master's thesis, Master of Science Thesis, University of Colorado, 2003.
9. Yang X.S. Firefly algorithm, stochastic test functions and design optimization. *Int. J. bio-inspired computation*, 2010.
10. L Zhao, G Zhuang, and X Xu. Facial expression recognition based on pca and nmf. In *Proceedings of the 7th World Congress on Intelligent Control and Automation*, pages 6826-6829, 2008.
11. Z Zheng, J Yang, and Y Zhu. Initialization enhancer for non-negative matrix factorization. *Engineering Applications of Artificial Intelligence*, vol. 20, pp. 101-110, 2007.
12. Ruicong Zhi, M. Flierl, Qiuqi Ruan, and W.B. Kleijn. Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition. *IEEE Transactions on Cybernetics*, 41:38-52, 2011.