Evolutionary Nonnegative Matrix Factorization with Adaptive Control of Cluster Quality

Liyun Gong[a], Tingting Mu[b], Meng Wang[c], Hengchang Liu[d], John Y. Goulermas[e]

[a]*School of Computer Science and School of Engineering, The University of Lincoln, Brayford Pool, Lincoln, UK, LN6 7TS*
[b]*School of Computer Science, University of Manchester, Manchester, UK, M1 7DN*
[c]*School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui, P. R. China, 230009*
[d]*School of Computer Science and Technology, University of Science and Technology of China, 188 Ren Ai Road,Suzhou, Jiangsu, P. R. China, 215123*
[e]*School of Electrical Engineering, Electronics and Computer Science, The University of Liverpool, Brownlow Hill, Liverpool, UK, L69 3GJ*

Abstract

Nonnegative matrix factorization (NMF) approximates a given data matrix using linear combinations of a small number of nonnegative basis vectors, weighted by nonnegative encoding coefficients. This enables the exploration of the cluster structure of the data through the examination of the values of the encoding coefficients and therefore, NMF is often used as a popular tool for clustering analysis. However, its encoding coefficients do not always reveal a satisfactory cluster structure. To improve its effectiveness, a novel evolutionary strategy is proposed here to drive the iterative updating scheme of NMF and generate encoding coefficients of higher quality that are capa-

Email addresses: lgong@lincoln.ac.uk (Liyun Gong), tingting.mu@manchester.ac.uk (Tingting Mu), wangmeng@hfut.edu.cn (Meng Wang), hcliu@ustc.edu.cn (Hengchang Liu), j.y.goulermas@liverpool.ac.uk (John Y. Goulermas)

ble of offering more accurate and sharper cluster structures. The proposed hybridization procedure that relies on multiple initializations reinforces the robustness of the solution. Additionally, three evolving rules are designed to simultaneously boost the cluster quality and the reconstruction error during the iterative updates. Any clustering performance measure, such as either an internal one relying on the data itself or an external based on the availability of ground truth information, can be employed to drive the evolving procedure. The effectiveness of the proposed method is demonstrated via careful experimental designs and thorough comparative analyses using multiple benchmark datasets.
Keywords: Nonnegative matrix factorization, clustering, initialization, evolutionary computation.

---

1. Introduction

Non-negative matrix factorization (NMF) has become an increasingly popular data processing tool in the recent years and is widely used by various communities including computer vision, text mining and bioinformatics. It is able to approximate each data sample in a data collection by a linear combination of a set of nonnegative basis vectors weighted by nonnegative weights. This often enables meaningful interpretation of the data, motivates useful insights and facilitates tasks such as dimensionality reduction and subspace learning [3, 29, 28, 49, 6], clustering [37, 31, 5, 12, 38], graph matching [20], etc.

An important group of works in NMF is focused on its optimization strategy and how to find accurate NMF approximations fast for large data sizes.

Typical NMF approximation approaches are reviewed in [1], and include alternating least square algorithms [30], gradient descent [26] and multiplicative update rules based on the creation of an auxiliary function for solving constrained optimization problems [23, 25]. Recent advances in NMF optimization include the use of the projected Newton method [14] and matrix manifold optimization, such as, Stiefel manifold when the extra orthogonality constraint is enforced [45]. Amongst these approaches, the multiplicative update is perhaps the most popular NMF solver, despite the fact that it is very sensitive to initializations. Usually, the simplest NMF initialization setup is to assign random values to the optimizing variables. This is certainly not the most effective strategy, and more sophisticated algorithms have been proposed to improve the convergence rate and the solution quality [22]. These, for example, include the initialization of the factorization matrices based on clustering solutions [41, 50, 34], or the use of data reduction algorithms such as principal component analysis [46] or singular value decomposition [4].

Another major group of NMF research is focused on the study of the NMF variations, so that they can better facilitate a specific data analysis task. For example, the least squares NMF takes into account the uncertainty measurements to better analyze the gene expression data [40]. The weighted-NMF [16] improves the NMF capabilities of representing positive local data for image classification tasks. Also, there are various approaches that introduce extra terms to the original NMF objective function of the reconstruction error by incorporating objectives, such as learning local presentations [24], preserving local data geometries [32, 51], incorporating topographic constraints [42], and enhancing class saparability [29, 39] to better serve a dimensionality re-

duction, clustering or classification task. A thorough survey on such types of approaches can be found in [29].

In this work, we focus particularly on the improvement of the multiplicative NMF update, which is the most commonly used NMF approach, to better serve the very important data analysis task of clustering. Data clustering has been used for decades in many fields, such as image processing and text mining [43, 7, 8], and has benefited more recently the microarray gene expression data analysis in genomic research [21]. Usually, the multiplicative NMF update can result in varying clustering results for the same given dataset due to initialization sensitivity. Moreover, driven by its reliance on the reconstruction error minimization, the resulting factorization matrices may not necessarily indicate the optimal clustering structures. To work on these issues, we propose a novel NMF updating strategy, which takes advantage of the hybridization of different NMF initialization setups and evolves along different directions to produce NMF approximations that suit better the clustering purpose. The effectiveness of the proposed method is demonstrated thoroughly through benchmark testing and comparisons with existing approaches.

The rest of the paper is organized as follows: Section 2 reviews the basic bacground of the multiplicative NMF update rules and the corresponding initializations. The proposed method is described in Section 3. Experimental results and comparative analyses are provided in Section 4 and the work is concluded in Section 5.

2. Background Methodology

2.1. NMF Formulation

Given a $d \times n$ non-negative matrix $\mathbf{X} = [x_{ij}]$ with each element $x_{ij} \geq 0$, its columns represent data points 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 reconstruction error is minimized:

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

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. Here, the number of the basis vectors k implies an upper bound of the rank of the approximated data matrix because $\text{rank}\left(\mathbf{W}\mathbf{H}^T\right) \leq \min\left(\text{rank}(\mathbf{W}), \text{rank}(\mathbf{H})\right) \leq k$. Also, we know the upper bound of the rank of the original data matrix $\mathbf{X}$, given as $\text{rank}(\mathbf{X}) \leq \min(d, n)$. Thus, it is common to set $k \leq \min(d, n)$ [37] so that the factorized matrix $\mathbf{W}\mathbf{H}^T$ is able to provide a low-rank approximation to the original data matrix $\mathbf{X}$ with the benefit of noise and data redundancy reduction. When the number of the basis vectors is set as the expected cluster number, each element of $\mathbf{H}$ can be viewed as the confidence value a data point belonging a data cluster. The ith data point is assigned to the jth cluster when $j = \text{argmax}_{l=1}^{k} h_{il}$. In addition to the Frobenius norm as used in Eq.(1), other metrics for evaluating the distance between the original data matrix $\mathbf{X}$ and the approximated one $\mathbf{W}\mathbf{H}^T$ can be used, such as Kullback-Leibler (KL) divergence [23] and earth mover's distance [35].

2.2. Multiplicative NMF Update Rules and Initialization

The solution to the constrained optimization problem in Eq.(1) can be approximated iteratively by the following multiplicative update rules [23]:

$$\mathbf{H}_{t+1} = \mathbf{H}_t \circ \left(\mathbf{X}^T\mathbf{W}_t\right) \oslash \left(\mathbf{H}_t\mathbf{W}_t^T\mathbf{W}_t\right), \quad (2)$$

$$\mathbf{W}_{t+1} = \mathbf{W}_t \circ \left(\mathbf{X}\mathbf{H}_t\right) \oslash \left(\mathbf{W}_t\mathbf{H}_t^T\mathbf{H}_t\right), \quad (3)$$

where $\circ$ denotes the Hadamard product and $\oslash$ the Hadamard division of two matrices of the same size, $\mathbf{W}_t$ and $\mathbf{H}_t$ denote the computed basis and encoding coefficient matrices at the tth iteration. Different setups of $\mathbf{W}_0$ and $\mathbf{H}_0$ may lead to different factorization results. One traditional strategy is random initialization (RI), which generates elements of $\mathbf{W}_0$ and $\mathbf{H}_0$ in a completely random manner [23]. To increase the convergence rate, more advanced initialization approaches have been developed, such as random Acol initialization (RAI) and clustering-based initialization (CI), which are described below.

2.2.1. Random Acol Initialization

Instead of forming completely random basis vectors of $\mathbf{W}_0$, the RAI method [22] forms each column of $\mathbf{W}_0$ by averaging p randomly selected columns of the data matrix $\mathbf{X}$. For example,

$$[\mathbf{W}_0]_i = \frac{1}{p} \sum_{j \in N_p^{(i)}} [\mathbf{X}]_j, \quad (4)$$

where $N_p^{(i)}$ denotes a set of p random integers between 1 and n generated for the ith column of $\mathbf{W}_0$, and $[\cdot]_i$ denotes the ith column of an input matrix. Then, $\mathbf{H}_0$ is computed by a least square computation [22] .

2.2.2. Clustering-based Initialization

When the output of NMF is used to facilitate the cluster exploration, it is natural to conduct the initialization by linking to a clustering algorithm. For example, it is possible to set $\mathbf{H}_0$ as the $n \times k$ cluster membership matrix $\mathbf{M} = [m_{ij}]$ obtained by a clustering algorithm [50, 34], where

$$m_{ij} = \begin{cases} 1, & \text{if the } i\text{th data point belongs to the } j\text{th cluster,} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

and set $\mathbf{W}_0$ as the $d \times k$ cluster centroid matrix $\mathbf{C}$ [50] that can be computed from $\mathbf{M}$ and the original data matrix $\mathbf{X}$ by

$$[\mathbf{C}]_j = \frac{1}{n_j} \sum_{m_{ij}=1} [\mathbf{X}]_i. \tag{6}$$

Here, n_j denotes the total number of data points belonging to the jth cluster.

3. Proposed Method

In this work, we propose an evolutionary strategy to improve the iterative updating procedure of NMF, refereed to as ENMF. It aims at producing higher quality basis and encoding coefficient matrices $\mathbf{W}$ and $\mathbf{H}$ that are more suitable for data clustering tasks. The algorithm starts from multiple pairs of initialization matrices of the basis vectors and encoding coefficients. These matrix pairs form an initial candidate set denoted as $S_0 = \left\{\left(\mathbf{W}_0^i, \mathbf{H}_0^i\right)\right\}_{i=1}^{m}$, where $\left\{\mathbf{W}_0^i\right\}_{i=1}^{m}$ and $\left\{\mathbf{H}_0^i\right\}_{i=1}^{m}$ are referred to as the seed matrices. The algorithm evolves creating an updated candidate set at each iteration, denoted as $S_t = \left\{\left(\mathbf{W}_t^i, \mathbf{H}_t^i\right)\right\}_{i=1}^{m_t}$ for the tth iteration with m_t denoting the new candidate number. The proposed updating rules result in an updated candidate number of $m_t = 3m + 1$ in each iteration, which we will discuss in detail in

Section 3.2. In the end, the optimal encoding coefficient matrix and its corresponding basis matrix are selected from the finally evolved candidate set based on a score function formulated to suit the data clustering task.

3.1. 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, that contains various seed matrices of the basis and encoding coefficient:

- The CI approach is first conducted via performing the k-means clustering [11]. The resulting binary cluster membership matrix $\mathbf{M}$ is used as $\mathbf{H}_0^1$, and the resulting clustering centroid matrix $\mathbf{C}$ as $\mathbf{W}_0^1$.

- Similar CI approach is conducted again but based on the fuzzy c-means (FCM) clustering [2]. The obtained cluster membership and centroid matrices $\mathbf{M}$ and $\mathbf{C}$ are used as $\mathbf{H}_0^2$ and $\mathbf{W}_0^2$, respectively. In addition, one more candidate is generated by setting the $n \times k$ member degree matrix $\mathbf{U} = [u_{ij}]$ of FCM as the $\mathbf{H}_0^3$ and the same centroid matrix $\mathbf{C}$ as $\mathbf{W}_0^3$. Here, the degree value u_{ij} represents the confidence value the ith data point belonging to the $j$the cluster and satisfies the conditions of $0 \leq u_{ij} \leq 1$ and $\sum_{j=1}^{k} u_{ij} = 1$.

- The RI and RAI approaches are used to generate the two candidates of $\left(\mathbf{W}_0^4, \mathbf{H}_0^4\right)$ and $\left(\mathbf{W}_0^5, \mathbf{H}_0^5\right)$.

In this case, a total number of $m = 5$ seed matrices are generated. However, it is worth to note that the proposed NMF updating algorithm is a general

method and the users could choose any type and number of initial candidates to suit their needs besides the above setup.

3.2. Evolving Strategy

In each iteration, three new subsets of candidates $S_{t+1}^{(M)}$, $S_{t+1}^{(S)}$ and $S_{t+1}^{(F)}$ are generated from the previous set S_t, according to three types of evolving rules proposed, which correspondingly are the multiplicative rule, the survival of the fittest rule and the firefly rule. The three subsets together constitute the updated set $S_{t+1} = S_{t+1}^{(M)} \cup S_{t+1}^{(S)} \cup S_{t+1}^{(F)}$ for the $(t+1)$th iteration. In the following, we explain these rules in detail.

3.2.1. Multiplicative Rule

The multiplicative rule is constructed to take advantage of the classical multiplicative update rule for NMF approximation. It generates the new candidate subset by

$$S_1^{(M)} = \Phi_M\left(S_0, \mathbf{X}\right), \tag{7}$$

for the first iteration and

$$S_{t+1}^{(M)} = \Phi_M\left(S_t^{(M)}, \mathbf{X}\right), \tag{8}$$

for the $(t+1)$th iteration $(t \geq 1)$. The operation $S' = \Phi_M(S, \mathbf{X})$ takes one set of matrix pairs $S = \{(\mathbf{W}_i, \mathbf{H}_i)\}_{i=1}^m$ and one $d \times n$ data matrix $\mathbf{X}$ as the input, where each matrix pair in S includes one $d \times k$ matrix $\mathbf{W}_i$ and one $n \times k$ matrix $\mathbf{H}_i$. It outputs a set of matrix pairs denoted as $S' = \{(\mathbf{W}_i', \mathbf{H}_i')\}_{i=1}^m$, which are formulated based on Eqs.(2, 3) and are as follows

$$\mathbf{H}_i' = \mathbf{H}_i \circ \left(\mathbf{X}^T\mathbf{W}_i\right) \oslash \left(\mathbf{H}_i\mathbf{W}_i^T\mathbf{W}_i\right), \tag{9}$$

$$\mathbf{W}_i' = \mathbf{W}_i \circ \left(\mathbf{X}\mathbf{H}_i\right) \oslash \left(\mathbf{W}_i\mathbf{H}_i^T\mathbf{H}_i\right). \tag{10}$$

This rule enables the inclusion of multiple NMF solutions obtained by the multiplicative update rules. Each solution provides an approximated optimal solution to the minimization problem of the reconstruction error. Also, each solution is obtained through a different way of initialisation such as the random and clustering-based ones. Given a number of m initial candidates for the algorithm to start, there are always m candidates generated by the multiplicative rule in each iteration.

3.2.2. Survival of the Fittest Rule

The survival of the fittest rule is designed to ensure the inclusion of the most competitive candidates that contain the best encoding coefficient matrix suitable for the clustering task in each iteration. In the first iteration, after applying the multiplicative rule to the initial candidate set S_0, the candidate population is enlarged to a combined set of $S_1^{(M)} \cup S_0$. A best encoding coefficient matrix $\mathbf{H}_0^*$ is selected from those contained in $S_1^{(M)} \cup S_0$ according to a predefined score function $O(\cdot)$ that assesses the quality of the input encoding coefficient matrix in terms of its clustering performance. This selection procedure can be formulated as

$$\mathbf{H}_0^* = \underset{(\mathbf{W},\mathbf{H})\in S_0 \cup S_1^{(M)}}{\arg\max} O(\mathbf{H}). \tag{11}$$

The survival of the fittest rule further generates a new candidate subset $S_1^{(S)}$ by modifying the matrices contained in $S_1^{(M)}$ based on $\mathbf{H}_0^*$, such that

$$S_1^{(S)} = \Phi_S\left(S_1^{(M)}, \mathbf{H}_0^*\right). \tag{12}$$

The operation $S' = \Phi_S(S, \mathbf{A})$ creates $m+1$ matrix paris $S' = \{(\mathbf{W}_i', \mathbf{H}_i')\}_{i=1}^{m+1}$ from the input set $S = \{(\mathbf{W}_i, \mathbf{H}_i)\}_{i=1}^{m}$ and the matrix $\mathbf{A}$. Specifically, all the

encoding coefficient matrices $\{\mathbf{H}'_i\}_{i=1}^{m+1}$ contained in S' are set as

$$\mathbf{H}'_i = \mathbf{A}, \quad i = 1, 2, \ldots, m, \tag{13}$$

while the basis matrices are generated by

$$\mathbf{W}'_i = \mathbf{W}_i, \quad \text{for } i = 1, 2, \ldots, m, \tag{14}$$

$$\mathbf{W}'_{m+1} = \max\left(0, \mathbf{X}\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A}\right), \tag{15}$$

with the operation $\max(0, \cdot)$ truncating all the negative elements of the input matrix to zero. This survival of the fittest operation $\Phi_S\left(S_1^{(M)}, \mathbf{H}_0^*\right)$ inherits the best encoding coefficient matrix $\mathbf{H}_0^*$ in all the $(m+1)$ newly generated candidates, among which the first m candidates keep the m basis matrices contained in $S_1^{(M)}$ as shown in Eq.(14) and the last candidate uses a new basis matrix generated according to $\mathbf{H}_0^*$ as shown in Eq.(15). The design of Eq.(15) is based on the alternating least squares algorithm for NMF [9, 22, 36], where $\mathbf{X}\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A}$ provides a least square estimation to a matrix $\mathbf{W}$ so that the distance between $\mathbf{W}\mathbf{A}^T$ and $\mathbf{X}$ is minimized. After applying the thresholding operation $\max(0, \cdot)$ to maintain only the positive elements in the estimated matrix, a basis matrix $\mathbf{W}'_{m+1}$ is generated containing non-negative elements to suit the purpose of NMF and meanwhile offering smaller reconstruction error when combined with $\mathbf{H}_0^*$.

By incorporating $S_1^{(M)} = \Phi_M(S_0, \mathbf{X})$ into Eqs.(12) and (11), we re-express the proposed update for the first iteration as

$$S_1^{(S)} = \Phi_S\left(\Phi_M(S_0, \mathbf{X}), \mathbf{H}_0^*\right), \tag{16}$$

$$\text{where} \quad \mathbf{H}_0^* = \underset{(\mathbf{W}, \mathbf{H}) \in S_0 \cup \Phi_M(S_0, \mathbf{X})}{\arg\max} O(\mathbf{H}). \tag{17}$$

The above update can be viewed as an operation on S_0 with the assistance of the two sub-operations of Φ_M and Φ_S. We follow a strategy similar to Eq.(16) to formulate the survival of the fittest rule for the $(t+1)$th iteration ($t \geq 1$), but let the update operate on the candidate subset $S_t^{(S)}$ generated by the survival of the fittest rule in the previous iteration, other than S_0. Replacing the set S_0 with $S_t^{(S)}$ in Eq.(16) and replacing $\mathbf{H}_0^*$ with $\mathbf{H}_t^*$, it gives

$$S_{t+1}^{(S)} = \Phi_S\left(\Phi_M\left(S_t^{(S)}, \mathbf{X}\right), \mathbf{H}_t^*\right). \tag{18}$$

In order to maintain a non-decreasing excellence of the new population, we select the most competitive candidate $\mathbf{H}_t^*$ from a combined set formed based on all the candidates generated in the previous iteration. By replacing the set S_0 with S_t in Eq.(17), we have

$$\mathbf{H}_t^* = \underset{(\mathbf{W},\mathbf{H})\in S_t \cup \Phi_M(S_t,\mathbf{X})}{\arg\max} O(\mathbf{H}). \tag{19}$$

Here, the set $S_t = S_t^{(M)} \cup S_t^{(S)} \cup S_t^{(F)}$ includes all the candidates generated by all the three proposed rules in the tth iteration.

To summarize, the survival of the fittest rule generates $m+1$ candidates in each iteration by combining the best encoding coefficient matrix $\mathbf{H}_t^*$ selected in each iteration with various basis matrices. This is equivalent to forcing all the weaker encoding coefficient matrices to eliminate themselves but let the best one to survive; thus, the rule is termed the survival of the fittest. The proposed rule combines $\mathbf{H}_t^*$ with different basis matrices as shown in Eq.(14) in addition to a computed one providing smaller reconstruction error as in Eq.(15). It attempts to introduce new candidates by altering the basis matrix of the strongest one to avoid being trapped in a local optimum.

3.2.3. Firefly Rule

The firefly rule is designed to generate candidates with the potential of providing higher clustering performance. The core difference between the firefly and the survival of the fittest rules is that the firefly one aims at generating new encoding coefficient matrices of higher quality, while the survival of the fittest at keeping the best encoding coefficient matrix from the previous iteration. In Section 3.2.2 its was explained how to select the best encoding coefficient matrix in each iteration in Eqs.(11) and (19), which result in $\{\mathbf{H}_0^*, \mathbf{H}_1^*, \ldots \mathbf{H}_t^*, \ldots\}$. In the following, we show how the firefly rule generates the new candidates by modifying the current candidates using $\mathbf{H}_t^*$.

We design the firefly rule so that it possesses a matching structure as the the survival of the fittest rule. In the first iteration, both rules modify the same candidate subset $S_1^{(M)}$ generated by the multiplicative rule but using different operations. Thus, the firefly rule can be expressed as follows by replacing the Φ_S operation in Eq.(12) with Φ_F

$$S_1^{(F)} = \Phi_F\left(S_1^{(M)}, \mathbf{H}_0^*\right) = \Phi_F\left(\Phi_M\left(S_0, \mathbf{X}\right), \mathbf{H}_0^*\right). \tag{20}$$

The $n \times k$ matrix $\mathbf{H}_0^*$ is selected from the encoding coefficient matrices contained in the combined candidate set $S_1^{(M)} \cup S_0$. The operation $S' = \Phi_F(S, \mathbf{A})$ takes a set $S = \{(\mathbf{W}_i, \mathbf{H}_i)\}_{i=1}^m$ and an $n \times k$ matrix $\mathbf{A}$ as input, while outputs a new set $S' = \{(\mathbf{W}'_i, \mathbf{H}'_i)\}_{i=1}^m$. Specifically, the relationship between the input and output of Φ_F is defined as

$$\mathbf{H}'_i = \mathbf{H}_i + \beta e^{-\gamma\|\mathbf{A}-\mathbf{H}_i\|_F^2}\left(\mathbf{A} - \mathbf{H}_i\right), \tag{21}$$

$$\mathbf{W}'_i = \begin{cases} \tilde{\mathbf{W}}_i, & \text{if } \left\|\mathbf{X} - \tilde{\mathbf{W}}_i\mathbf{H}_i'^T\right\|_F^2 < \left\|\mathbf{X} - \mathbf{W}_i\mathbf{H}_i'^T\right\|_F^2, \\ \mathbf{W}_i, & \text{otherwise,} \end{cases} \tag{22}$$

where the matrix $\tilde{\mathbf{W}}_i$ is computed by

$$\tilde{\mathbf{W}}_i = \max\left(0, \mathbf{X}\left(\mathbf{H}_i'\mathbf{H}_i'^T\right)^{-1}\mathbf{H}_i'\right), \tag{23}$$

with $0 < \beta \leq 1$ and $\gamma > 0$ being the user selected parameters. From the $(t+1)$th iteration ($t \geq 1$), the firefly rule starts to generate the new candidate subset $S_{t+1}^{(F)}$ from the previous subset $S_t^{(F)}$. By replacing the Φ_S operation in Eq.(18) with Φ_F and the subset $S_t^{(S)}$ with $S_t^{(F)}$, we obtain the matching formulation of the firefly update for the $(t+1)$th iteration ($t \geq 1$)

$$S_{t+1}^{(F)} = \Phi_F\left(\Phi_M\left(S_t^{(F)}, \mathbf{X}\right), \mathbf{H}_t^*\right), \tag{24}$$

where the $n \times k$ matrix $\mathbf{H}_t^*$ is selected from the combined set of $S_t \cup \Phi_M\left(S_t, \mathbf{X}\right)$. In the following, we explain the core ideas behind the firefly operation Φ_S as formulated in Eqs.(21) and (22).

The design of Eq.(21) is motivated by the recent evolutionary optimization algorithm inspired by the flashing behavior of firefly, known as the firefly algorithm [44]. The algorithm assumes that attractiveness between fireflies is proportional to their brightness, thus, given any two fireflies, one will move towards the other that glows brighter. However, such attractiveness decreases when the distance between two fireflies increases. Following Eq.(21), the encoding coefficient matrix of each candidate in either $S_1^{(M)}$ for the first iteration or $\Phi_M\left(S_t^{(F)}, \mathbf{X}\right)$ for the tth ($t > 1$) iteration is viewed as a firefly. Its quality is evaluated by the score function $O(\cdot)$, representing the brightness degree of the firefly. By rewriting Eq.(21) as

$$\mathbf{H}_i' = \left(1 - \beta e^{-\gamma\|\mathbf{A}-\mathbf{H}_i\|_F^2}\right)\mathbf{H}_i + \beta e^{-\gamma\|\mathbf{A}-\mathbf{H}_i\|_F^2}\mathbf{A}, \tag{25}$$

it can be seen that the newly generated encoding coefficient matrix is a mixture of the one generated by the multiplicative rule ($\mathbf{H}_i$) and the pre-selected

one with the best clustering performance ($\mathbf{A} = \mathbf{H}_t^*$). This is equivalent to moving the fireflies towards the brightest firefly. In Eq.(21) the exponential term $\beta e^{-\gamma \|\mathbf{A}-\mathbf{H}_i\|_F^2}$ that determines how much $\mathbf{H}_i$ should be moved towards $\mathbf{A} = \mathbf{H}_t^*$ is directly controlled by the distance between the two matrices. This is equivalent to forcing the attractiveness towards the brightest firefly to decrease as the relative distance increases. The parameter $0 < \beta \leq 1$ adjusts the contribution of $\mathbf{A}$ to the construction of $\mathbf{H}_i'$, controlling the dominating degree of the brightest firefly to determine the positions of the other ones. The parameter $\gamma > 0$ controls how much the distance $\|\mathbf{A} - \mathbf{H}_i\|_F^2$ affects the contribution of $\mathbf{A}$ in the construction of $\mathbf{H}_i'$, determining the decaying degree of the attractiveness between fireflies against their distance.

Eq.(22) updates the basis matrix for each of the encoding coefficient matrix $\mathbf{H}_i'$. The design is based on the alternating least squares algorithm for NMF [9, 22, 36], which updates the basis matrix based on the current encoding coefficient matrix through first solving the unconstrained reconstruction error minimization problem of

$$\min_{\mathbf{W}} \left\| \mathbf{X} - \mathbf{W}\mathbf{H}_i'^T \right\|_F^2, \tag{26}$$

by setting its derivative to zero and then modifying the resulting matrix by converting all its negative elements to zero. This procedure gives the matrix $\tilde{\mathbf{W}}_i$. However, the modification step of converting the negative elements to zero potentially raises the risk of obtaining undesired reconstruction error. An alternative setup of $\mathbf{W}_i'$ is to employ the original one $\mathbf{W}_i$ as generated by the multiplicative rule, given the fact that the basis matrix does not affect directly the data cluster structure. In Eq.(22), between $\mathbf{W}_i'$ and $\mathbf{W}_i$ we choose the one possessing the smaller reconstruction error in order to

prevent the proposed evolving procedure from sacrificing the data representation accuracy to compensate for the cluster quality. Here, $\tilde{\mathbf{W}}_i$ is always nonnegative. Also, according to Eq.(25), it is obvious that, when $\mathbf{H}_i$ and $\mathbf{A}$ are both non-negative, $\mathbf{H}'_i$ is non-negative. These guarantee that the matrix pairs $(\mathbf{W}'_i, \mathbf{H}'_i)$ generated by the firefly rule are eligible as NMF candidates.

The firefly rules generates a total of m candidates in each iteration. As shown in Eqs.(16) and (18) for the survival of the fittest rule and Eqs.(20) and (24) for the firefly rule, instead of directly updating S_0 and $S_t^{(S)}$ (or $S_t^{(F)}$) with Φ_S (or Φ_F), the multiplicative operation Φ_M is first used to smoothen out the given candidates, which may potentially reduce the reconstruction error. The mixture of Φ_M and Φ_S (or Φ_F) attempts to evolve matrix pairs offering good quality of encoding coefficient matrix while alternatively ensuring the joint quality of the basis and encoding coefficient matrices.

3.3. Score Function

Since the primary goal of this work is to improve NMF so that it can serve better the data clustering task, it is natural to formulate the score function as a cluster validity measure [10, 27] that assesses the cluster quality. When the internal evaluation is used, the assessment is based on the data matrix $\mathbf{X}$ itself. The cluster structure possessing higher within-cluster similarity and lower between-cluster similarity is of better quality. In this case, the Dunn induex (DI) [19] for example can be directly used as the score function. When the external evaluation is used, the assessment compares the clustering results with the ground truth partition of the data, for which the cluster structure that better matches the ground truth partition is of higher quality. For example, the RAND index [33] can be used as the score function for the

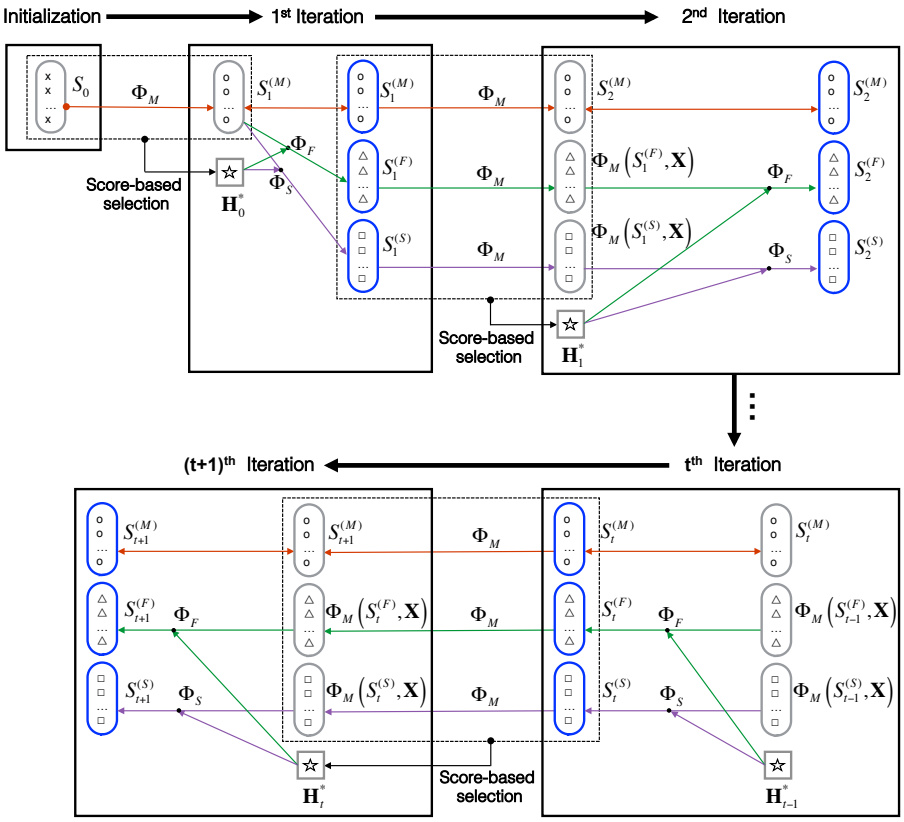


Figure 1: Data flow of the proposed ENMF. The circle, triangle and rectangle symbols represent candidates derived during the generation of the $S_t^{(M)}$, $S_t^{(F)}$ and $S_t^{(S)}$ subsets, respectively. The lines of different shades represent data flows from the three rules.

external evaluation, to compute the percentage of the correct data partitions offered by the clustering result as compared to the ground truth.

The overall data flow of the proposed ENMF strategy is shown in Figure 1. In the first iteration, starting from the m pairs of seed matrices included in the initial candidate set S_0, different candidate subsets are generated from S_0 by following the three proposed rules. These lead to three evolved subsets $S_1^{(M)}$, $S_1^{(S)}$ and $S_1^{(F)}$ containing m, m+1 and m candidates, respectively, constituting the new population S_1 of the first iteration. From the second iteration,

different candidate fractions in the population are updated by different rules. For example, $S_t^{(M)}$ is updated by the multiplicative rule, $S_t^{(S)}$ by the survival of the fittest rule and $S_t^{(F)}$ by the firefly rule. The updated subsets $S_{t+1}^{(M)}$, $S_{t+1}^{(S)}$ and $S_{t+1}^{(F)}$ include m, $m+1$ and m candidates, respectively, constituting the new population S_{t+1}. These updating rules result in a fixed population size of $m_t = 3m + 1$ during each iteration. The motivation behind these rules are summarized as follows. The operation Φ_M based on the multiplicative rule aims at generating candidates that are able to converge to an NMF solution driven by the reconstruction error minimization. The operations of Φ_S and Φ_F based on the survival of the fittest and firefly rules aim at the local improvement of the candidates to produce better quality of data clusters within each iteration. Specifically, Φ_F moves the the encoding coefficient matrices generated by the multiplicative rule towards a best one selected, based on a pre-defined score function, while Φ_S ensures the best selected encoding coefficient matrix is included in the updated candidate set. The score function assesses the cluster quality so that the output of NMF is able to serve better a data clustering task.

4. Experimental results and analysis

As explained in Section 2.1, the factorization of a $d \times n$ data matrix $\mathbf{X}$ into one $d \times k$ basis matrix $\mathbf{W}$ and one $n \times k$ encoding coefficient matrix $\mathbf{H} = [h_{il}]$ can be directly used to discover the data cluster structure, by setting the number of the basis vectors k as the number of the expected clusters and by assigning the ith data point to the jth cluster through $j = \text{argmax}_{l=1}^{k} h_{il}$. To evaluate the matrix factorization output in terms of its corresponding clustering performance, we conduct experiments using ten benchmark classification

datasets from the UCI machine learning repository, including balance scale, breast tissue, breast cancer Wisconsin diagnostic (WDBC), breast cancer Wisconsin original (BCWO), dermatology, glass identification, Haberman's survival, iris, thyroid and red wine quality (winered). The characteristics of these datasets are summarized in Table 1, where the first word or the abbreviation of the data name is used to refer to each dataset. For data preprocessing, a scalar $|\min_{i,j} x_{ij}|$ is added to the input data matrix $\mathbf{X}$ when it contains negative elements.

4.1. Experimental Setup

We compare the proposed ENMF updating strategy with the classical multiplicative update [23]. We also examine effects of different factorization initialization approaches including RI, RAI and the three types of CI based on the membership matrix of the k-means clustering (CI1), the membership matrix of the FCM clustering (CI2) and the member degree matrix of the FCM clustering (CI3). Each initialization approach is run for five times, generating five pairs of encoding coefficient and basis matrices. For ENMF, these five pairs are used as the initial candidates to start the algorithm, which means that the initial candidate population size of the ENMF is $m = 5$ and a candidate population size of $m_t = 3m + 1 = 16$ is maintained during the evolving iterations. Apart from these, we also examine the results of the ENMF by using a mixture of all five initialization approaches (MIX). In this case, five pairs of initial coefficient and basis matrices are generated by running each of the five initialization approaches once, which again leads to an initial candidate population size of $m = 5$ and a population size of $m_t = 16$ during the iterations. Both internal evaluation based on DI index

and external evaluation based on RAND index are experimented. When external evaluation is used as the score function, the evolving procedure of the ENMF becomes supervised due to the involvement of the ground truth class information, for which we split the dataset into two separate sets for the training and test purposes. The RAND index computed with the training set is used as the score function to drive the evolving of the ENMF[1]. For the standard NMF based on the multiplicative update, the same five seed matrix pairs that are used as the initial candidates of the ENMF are used to initialize the updating procedure of the standard NMF. This leads to five solutions of the standard NMF and the one possessing the best clustering performance is reported. All the experiments are repeated five times and the averaged clustering performance is reported. In all experiments, the number of the basis vectors k is set as the cluster number for both NMF and ENMF. For ENMF, we adapt the recommended parameter setting $\gamma = \beta = 1$ for the firefly rule as suggested by [44]. The iteration numbers for the NMF and ENMF updates are both fixed as 500 in all the experiments.

4.2. Results and Analysis

Table 2 compares the standard NMF with the multiplicative update and the proposed ENMF with a mixture of three updates under the different initialization setups of CI1, CI2, CI3, RI, RAI and MIX as explained in the previous section. For a more clear identification of the performance improvement of ENMF over NMF, we also summarize the performance difference between ENMF and NMF under different initialization setups in the same

[1]A four-fold cross validation (CV) is performed to report the RAND performance.

table. It can be seen that for most cases ENMF leads to an improved clustering performance as compared to standard NMF. This shows effectiveness of the proposed factorization strategy that is particularly designed to suit better the clustering purpose. Particularly, ENMF is more forceful than NMF to produce a cluster structure that is more compatible to a ground truth partition associated with the data, indicated by the significant performance improvement in terms of the external evaluation measure of RAND index. Under the internal evaluation based on DI index, ENMF provides more satisfactory improvement over NMF for the case of random initialization than clustering based intialization. This is because when the output of a clustering approach such as k-means and FCM is employed to initialize the factorization, the matrix $\mathbf{H}$ has already contained a cluster structure possessing good DI value due to the nature of the clustering algorithm, and thus the improvement over the later involving procedure can be in a smaller scale, or the performance can remain the same for some datasets.

In Figures 2 and 3, we compare the convergence of ENMF with mixed initialization and that of NMF multiplicative update initialized by different approaches of RI, RAI and the best one from CI1 to CI3 (referred to as CI in the figure) for different datasets, based on RAND and DI indices, respectively. It can be observed that, clustering performance obtained by NMF initialized with the output of a clustering algorithm does not improve much over iterations for most datasets. The combination of NMF and clustering based initialization is only worthy when the later NMF update is able to improve the cluster quality. Otherwise, the clustering algorithm on its own can be directly applied to save the extra computational cost consumed by

the NMF update. Differently, the clustering performance obtained based on the ENMF initialized by a mixture of different approaches can significantly and rapidly improve over iterations for most datasets.

Overall, as shown in Table 2 and Figures 2 and 3, the performance of the ENMF and NMF is affected by the employed initialization approach. There is no superiority of one initialization approach over another given different datasets. For many datasets, a mixed initialization leads to better performance, indicating the effectiveness of hybridization. When ENMF employs similar seed matrices, e.g., those obtained by CI1 and CI2, they do not motivate the evolving procedure to generate better candidates than NMF, as exemplified by the zero improvements in Table 2. Differently, random initialization, such as RAI and RI, offers solutions of varying quality. It helps to avoid the local optimum by preventing the generation of candidates that are too similar, but may lead to unsatisfactory convergence without sufficient number of iterations due to the lack of seed quality control. By initializing the ENMF with a mixture of different types of seed matrices, their quality and diversity are balanced, and have the potential to generate solutions providing higher quality of clusters and converge in less iterations.

In the above experiments, we adopt the recommended setting of $\beta = \gamma = 1$ for the firefly rule as suggested by [44] and use an initial population size of $m = 5$ for the ENMF determined by the number of used initialization approaches. Here, we further conduct some experiments to investigate the impact of different values of β, γ and m. By letting the firefly parameters vary within the ranges of $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ and $\gamma \in \{0.001, 0.01, 0.1, 1, 10, 100\}$, we record the clustering performance of the ENMF at its 100th and 500th

iterations for different parameter combinations, and demonstrate the corresponding RAND and DI performance in Figures 4 and 5 using the iris and Haberman's survival datasets. It can be seen from the left plot of each subfigure that the clustering performance of the ENMF reaches its best at different parameter values for different datasets. It can also be seen from the right plot of each subfigure that the adopted parameter setting of $\beta = \gamma = 1$, although is not the best choice for individual cases, it appears as an above average choice for most cases in both the early stage (100th iteration) and the end (500th iteration) of the evolving procedure. When the DI index is used as the score function, the performance is less sensitive to parameter setting than using the RAND index. In general, it is reasonable to adopt the setting choice of $\beta = \gamma = 1$ to save the extra effort on performing parameter selection. About the initial population size, in Section 3.1 we propose to generate five candidates using five different NMF initialization approaches. It is of course possible to generate more than one candidate using each approach, which leads to a larger population size. To investigate the impact of the population size, we run the ENMF with the random initialization approach RAI under different initial population sizes of $m = 3$ and $m = 10$ using the RAND index as the score function, for which the initial candidates are generated by running the RAI approach three and ten times, respectively, using the iris and Haberman's survival datasets under four random training-test partitions. It can be seen from Figure 6, among seven trials of the two datasets, the smaller population size ($m = 3$) actually provides higher clustering performance, while only in one trial of the iris data, the larger population size ($m = 10$) performs better. We also conduct the same experiments for the

clustering based initialization approach CI and have observed that different population sizes lead to almost the same performance of the ENMF. Since a larger population size does not necessarily offer better performance for the random initialization approach and different population sizes usually offer similar performance for the clustering based initialization, we include only one candidate for each initialization to improve the algorithm efficiency.

5. Conclusion

We have proposed a new strategy for conducting NMF, so that the resulting encoding coefficient matrix $\mathbf{H}$ is capable of representing better quality of clustering structures. Three rules have been designed, of which the first rule inherits the classical multiplicative update, while the other two rules are driven by the preservation of stronger candidates offering higher clustering quality and are inspired by the evolutionary optimization algorithm of firefly. Any measure for assessing the clustering performance can be used as the score function to control the evolving procedure. The proposed framework is general and can also be applied to improve NMF applications for other data analysis tasks by setting appropriate score functions. For example, measures of compression rates, data sparsity and reconstruction errors can be used for data compression tasks. Experimental results have demonstrated the superior performance of the proposed method over existing ones for the data clustering task evaluated with ten benchmark datasets.

For the future work, we will investigate the application of the current algorithm in the sparse coding as mentioned in [48] [47] and [17]. Extra terms will be added in the optimisation function (as in [48] and [17]) to control the sparsity of the encoding vectors, while the proposed algorithm will be applied

for solving the new proposed optimisation function to obtain sparse data representations. Besides,not only limited to the clustering, we will apply the developed algorithm for other applications, i.e., data compressions for high-dimensionality data such as image and video ([15], [18],[13], [48] and [17]).

References

[1] Berrya, M. W., Amy, M. B., Langvilleb, N., Paucac, V. P., and Plemmonsc, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, **52**, 155–173.

[2] Bezdek, J. C., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Comput. Geosoci, vol. 10, no. 2-3, pp.191-203*.

[3] Bin, G., Woo, W. L., and Ling, B. W. K. (2010). Improving pomdp tractability via belief compression and clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **40**(1), 125–136.

[4] Boutsidis, C. and E.Gallopoulos (2008). Svd based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition 41, ELSEVIER, pp. 1350-1362*.

[5] Brunet, J. P., Tamayo, P., Golub, T. R., and Mesirov, J. P. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences of the United States of America (PNAS), vol. 101(12), pp. 4164- 4173*.

[6] Bucak, S. S. and Gunsel, B. (2009). Incremental subspace learning via non-negative matrix factorization. *Pattern Recognition*, **42**(5), 788–797.

[7] Cao, J., Wu, Z., Wu, J., and Xiong, H. (2013). SAIL: Summation-based incremental learning for information-theoretic text clustering. *IEEE Transactions on Cybernetics*, **43**, 570–584.

[8] Chen, L., C, C., and Lu, M. (2011). A multiple-kernel fuzzy c-means algorithm for image segmentation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **41**(5), 1263 –1274.

[9] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S. I. (2009). *Non-negative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. John Wiley.

[10] de L. Balaguer, M. A. and Williams, C. M. (2013). A cluster validity framework based on induced partition dissimilarity. *IEEE Transactions on Cybernetics*, **43**, 308–320.

[11] Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, **21**, 768–769.

[12] Gao, Y. and Church, G. (2005). Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics, vol. 21, pp. 3970-3975*.

[13] Gong, L., Mu, T., and Goulermas, J. Y. (2015). Evolutionary nonnegative matrix factorization for data compression. *In: Huang DS., Bevilacqua V., Premaratne P. (eds) Intelligent Computing Theories and Methodologies. ICIC 2015. Lecture Notes in Computer Science, vol 9225. Springer*.

[14] Gong, P. and Zhang, C. (2012). Efficient nonnegative matrix factorization via projected newton method. *Pattern Recognition*, **45**(9), 3557–3565.

[15] Guan, Z., Zhang, L., Peng, J., and Fan, J. (2015). Multi-view concept learning for data representation. *Transactions on Knowledge and Data Engineering, vol. 27, No. 11*, pages 3016–3028.

[16] Guillamet, D., Vitria, J., and Scheile, B. (2003). Introducing a weighted nonnegative matrix factorization for image classification. *Pattern Recognition Letters,vol. 24, pp. 2447-2454*.

[17] Hong, C., Yu, J., Tao, D., and Wang, M. (2015a). Image-based 3d human pose recovery by multi-view locality sensitive sparse retrieval. *IEEE Transactions on Industrial Electronics*, **62**(6), 3742–3751.

[18] Hong, C., Yu, J., You, J., Chen, X., and Tao, D. (2015b). Multi-view ensemble manifold regularization for 3d object recognition. *Information Sciences, vol. 320, No. 1*, pages 395–405.

[19] J, D. (1974). Well separated clusters and optimal fuzzy partitions. *Journal Cybernetics*, **4**(1), 95–104.

[20] Jiang, B., Zhao, H., Tang, J., and Luo, B. (2014). A sparse nonnegative matrix factorization technique for graph matching problems. *Pattern Recognition*, **47**(2), 736–747.

[21] Jiang, D. X., Tang, C., and Zhang, A. D. (2004). Cluster analysis for gene expression data: A survey. *IEEE Trans. Know. and Data Eng., vol. 16, no. 11, pp. 1370-1386*.

[22] Langville, A. N., Meyer, C. D., and Albright, R. (2006). Initializations for the nonnegative matrix factorization. In *Proceeding of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[23] Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Adv. Neural Inf. Process. Systems 13 (2000)*, pages 556–562.

[24] Li, S. Z., Hou, X., Zhang, H., and Cheng, Q. (2001). Learning spatially localized parts-based representation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 207–212.

[25] Lin, C. (2007a). On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, **18**(6), 1589–1596.

[26] Lin, C. (2007b). Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, **19**(10), 2756–2779.

[27] Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J., and Wu, S. (2013). Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics*, **43**, 982–994.

[28] Nikitidis, S., Tefas, A., Nikolaidis, N., and Pitas, I. (2012). Subclass discriminant nonnegative matrix factorization for facial image analysis. *Pattern Recognition*, **45**(12), 4080–4091.

[29] Nikitidis, S., Tefas, A., and Pitas, I. (2014). Projected gradients for

subclass discriminant nonnegative subspace learning. *IEEE Transactions on Cybernetics*. In press, DOI: 10.1109/TCYB.2014.2317174.

[30] Paatero, P. and Tapper, U. (1994). Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, **5**, 111–126.

[31] Pascual-Montano, A., Carmona-Saez, P., Chagoyen, M., Tirado, F., Carazo, J. M., and Pascual-Marqui, R. D. (2006). bionmf: a versatile tool for nonnegative matrix factorization in biology. *BMC Bioinformatics, vol. 7(1), pp. 366-374*.

[32] Pei, X., Wu, T., and Chen, C. (2014). Automated graph regularized projective nonnegative matrix factorization for document clustering. *IEEE Transactions on Cybernetics*. In Press, DOI:10.1109/TCYB.2013.2296117.

[33] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, Theory and Methods Section, vol. 66, pp. 846-850*.

[34] Rezaei, M., Boostani, R., and Rezaei, M. (2011). An efficient initialization method for nonnegative matrix factorization. *Journal of Applied Sciences, vol. 11, pp. 354-359*.

[35] Sandler, R. and Lindenbaum, M. (2011). Nonnegative matrix factorization with earth mover's distance metric for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(8), 1590–1602.

[36] Schachtner, R. (2010). *Extensions of non-negative matrix factorization*

and their application to the analysis of wafer test data. Ph.D. thesis, University of Regensburg.

[37] Shahnaza, F., Berrya, M. W., Paucab, V. P., and Plemmonsb, R. J. (2006). Document clustering using nonnegative matrix factorization. *Information Processing and Management*, **42**(2), 373–386.

[38] Shang, F., Jiao, L., and Wang, F. (2012). Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognition*, **45**(6), 2237–2250.

[39] S.Zafeiriou, Tefas, A., Buciu, I., and Pitas, I. (2006). Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification. *IEEE Transactions on Neural Networks*, **13**(3), 683–695.

[40] Wang, G., Kossenkov, A. V., and Ochs, M. F. (2006). LS-NMF: a modified nonnegative matrix factorization algorithm utilizing uncertainty estimates. *BMC Bioinformatics, 7:175, doi: 10.1186/1471-2105-7-175*.

[41] Wild, S. (2003). *Seeding non-negative matrix factorizations with the spherical k-means clustering*. Master's thesis, Master of Science Thesis, University of Colorado.

[42] Xiao, Y., Zhu, Z., Zhao, Y., Wei, Y., Wei, S., and Li, X. (2014). Topographic NMFfor data representation. *IEEE Transactions on Cybernetics*. In Press, DOI:10.1109/TCYB.2013.2294215.

[43] Xu, R. and Wunsch, D. I. (2005). Survey of clustering algorithms. *IEEE Trans. Neural Networks,vol. 16, no. 3, pp. 645 - 678*.

[44] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimization. *Int. J. bio-inspired computation*.

[45] Yoo, J. and Choi, S. (2010). Nonnegative matrix factorization with orthogonality constraints. *Journal of Computing Science and Engineering*, **4**(2), 97–109.

[46] Zhao, L., Zhuang, G., and Xu, X. (2008). Facial expression recognition based on pca and nmf. In *Proceedings of the 7th World Congress on Intelligent Control and Automation*, pages 6826–6829.

[47] Zhao, W., Liu, Z., Guan, Z., Lin, B., and Cai, D. (2016a). Orthogonal projective spare coding for image representation. *Neurocomputing, vol. 173*, pages 270–277.

[48] Zhao, W., Liu, Z., Guan, Z., Lin, B., and Cai, D. (2016b). Orthogonal projective sparse coding for image representation. *Neurocomputing*, **173**(2), 270–277.

[49] Zheng, W., Lai, J., Liao, S., and He, R. (2012). Extracting non-negative basis images using pixel dispersion penalty. *Pattern Recognition*, **45**(8), 2912–2926.

[50] Zheng, Z., Yang, J., and Zhu, Y. (2007). Initialization enhancer for non-negative matrix factorization. *Engineering Applications of Artificial Intelligence,vol. 20, pp. 101-110*.

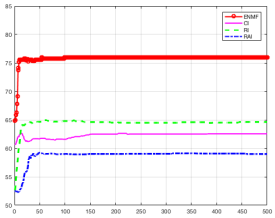
[51] Zhi, R., Flierl, M., Ruan, Q., and Kleijn, W. (2011). Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition. *IEEE Transactions on Cybernetics*, **41**, 38–52.

Datasets	No. of Instances (n)	No. of Features (d)	No. of Classes (k)
Balance	625	4	3
Breast	106	9	6
WDBC	569	30	2
BCWO	683	9	2
Dermatology	358	34	6
Glass	214	9	6
Haberman	306	3	2
Iris	150	4	3
Thyroid	215	5	3
Winered	1599	11	6

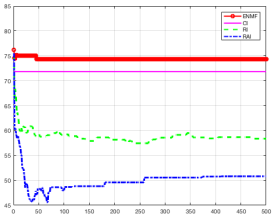
Table 1: Summary of dataset characteristics.

Method	Balance		Breast		WDBC		BCWO		Dermatology		Glass		Haberman		Iris		Thyroid		Winered	
Index	R (%)	DI	R (%)	DI	R (%)	DI	R (%)	DI	R (%)	DI	R (%)	DI	R (%)	DI	R (%)	DI	R (%)	DI	R (%)	DI
NMF-CI1	63.8	**1.14**	72.0	0.42	75.2	0.92	92.4	**1.11**	71.1	**1.13**	70.4	0.82	50.4	1.18	88.2	**1.81**	78.1	**1.22**	59.3	0.20
ENMF-CI1	66.0	**1.14**	72.0	0.42	75.2	0.92	92.4	**1.11**	73.3	**1.13**	70.7	0.82	50.4	1.18	88.2	**1.81**	78.1	**1.22**	58.7	0.20
Improvment	+2.2	0	0	0	0	0	0	0	+2.2	0	+0.3	0	0	0	0	0	0	0	-0.6	0
NMF-CI2	65.9	1.11	68.7	**1.25**	75.2	0.92	91.6	**1.11**	70.8	0.96	71.8	0.56	50.7	1.18	88.3	1.78	73.6	0.85	59.1	0.21
ENMF-CI2	67.3	1.11	68.7	**1.25**	75.2	0.92	91.6	**1.11**	70.8	0.96	71.8	0.56	50.7	1.18	88.3	1.78	73.6	0.85	58.5	0.21
Improvment	+1.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.6	0
NMF-CI3	65.6	1.00	69.6	0.68	72.7	1.15	63.9	1.00	80.7	0.55	72.5	0.52	50.9	1.18	93.5	1.26	76.4	0.82	59.4	0.28
ENMF-CI3	**81.2**	**1.14**	67.9	0.68	76.7	**1.16**	**93.5**	**1.11**	82.0	0.99	72.3	0.61	50.5	1.18	93.5	1.78	**83.5**	1.06	58.7	0.28
Improvment	+15.6	+0.14	-1.7	0	+4.0	+0.01	+29.6	+0.11	+1.3	+0.44	-0.2	+0.09	-0.4	0	0	+0.52	+7.1	+0.25	-0.7	0
NMF-RI	65.7	1.00	64.5	0.88	71.1	1.15	68.3	1.00	85.6	0.88	69.5	0.74	50.4	1.18	81.5	1.01	76.2	0.82	59.0	0.54
ENMF-RI	73.9	1.11	**75.2**	1.00	74.3	**1.16**	71.1	1.01	85.5	1.00	69.7	0.94	54.7	1.18	86.2	1.27	79.5	1.03	59.7	**0.61**
Improvment	+8.2	+0.11	+10.7	+0.11	+3.2	+0.01	+2.8	+0.01	-0.1	+0.13	+0.2	+0.20	+4.3	0	+4.7	+0.26	+3.3	+0.22	0.7	+0.06
NMF-RAI	64.8	1.00	57.6	0.94	69.2	1.15	68.0	1.00	86.1	0.87	69.4	0.70	51.9	1.18	81.1	1.01	74.6	0.82	58.6	0.50
ENMF-RAI	73.2	1.06	73.9	0.99	69.3	**1.16**	70.1	1.02	**86.9**	0.97	69.3	0.91	54.0	**1.20**	85.6	1.27	77.5	1.14	59.6	0.59
Improvment	+8.4	+0.06	+16.3	+0.05	+0.1	+0.01	+2.1	+0.02	+0.8	+0.10	-0.1	+0.21	+2.1	+0.02	+4.5	+0.26	+2.9	+0.33	+1	+0.09
NMF-MIX	65.9	1.08	71.4	0.83	75.2	1.15	92.4	**1.11**	85.9	0.96	72.5	0.65	51.8	1.18	93.5	**1.81**	78.2	0.94	59.3	0.51
ENMF-MIX	76.0	1.11	72.9	1.02	**77.1**	**1.16**	**93.5**	**1.11**	86.2	1.04	**72.7**	**0.96**	**55.0**	1.18	**95.6**	**1.81**	**83.5**	1.03	**59.8**	0.60
Improvment	+10.1	+0.03	+1.5	+0.19	+1.9	+0.01	+1.1	0	+0.3	+0.08	+0.2	+0.31	+3.2	0	+2.1	0	+5.3	+0.09	+0.5	+0.08

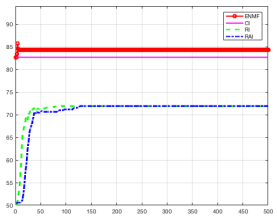
Table 2: Clustering performance comparison for different datasets in terms of the RAND index in percentage (denoted by R) and the DI index. The best performance is highlighted in bold. The performance improvement of ENMF over NMF is reported for each initialization setup.



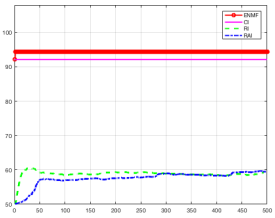
(a) Balance



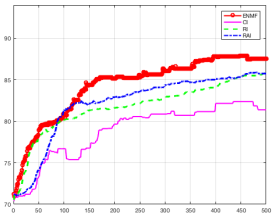
(b) Breast



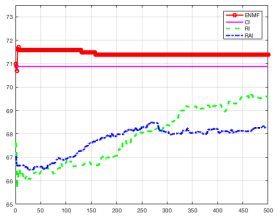
(c) WDBC



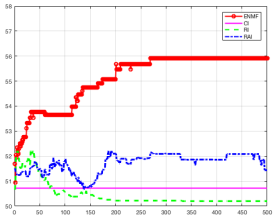
(d) BCWO



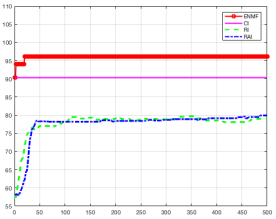
(e) Dermatology



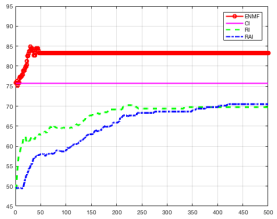
(f) Glass



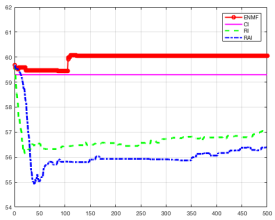
(g) Haberman



(h) Iris

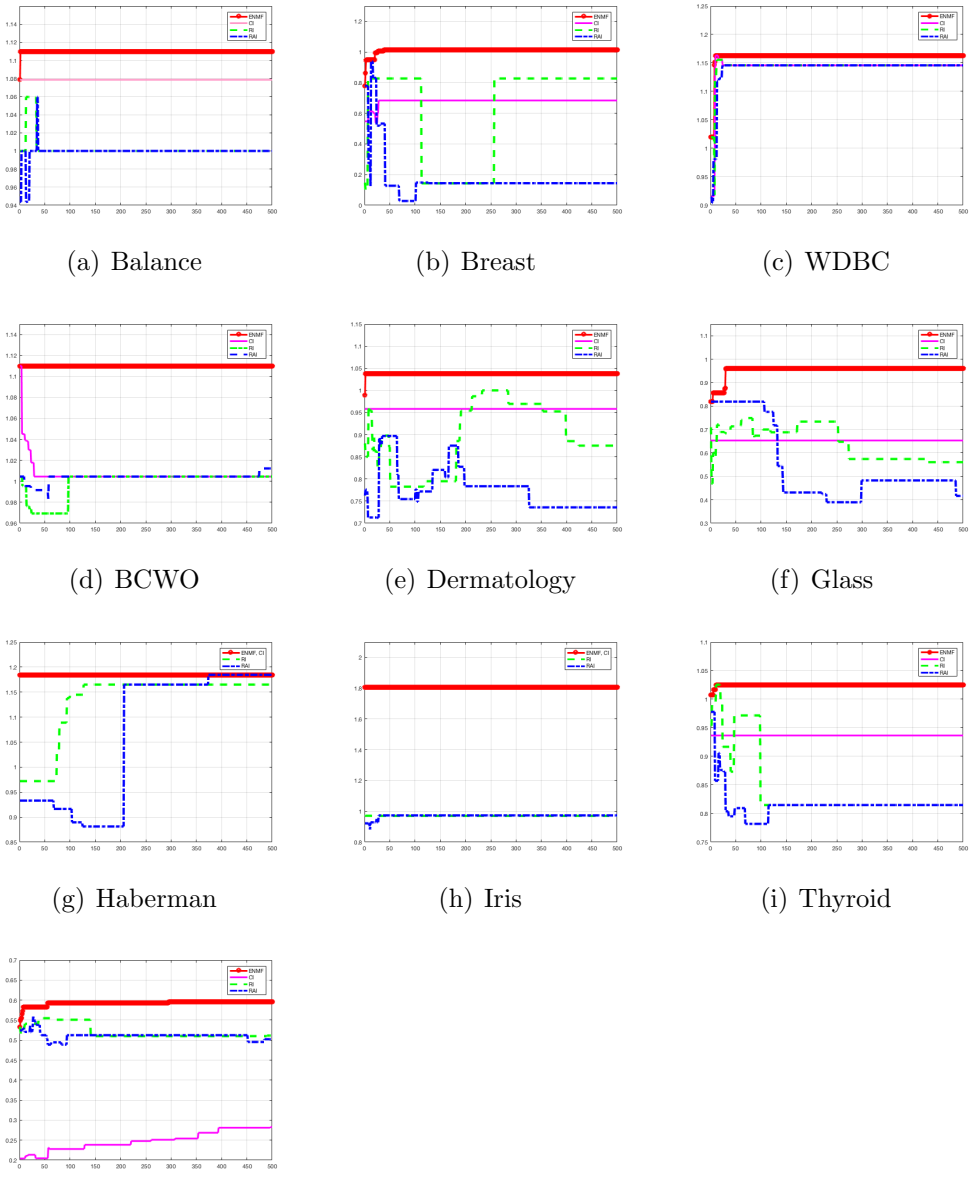


(i) Thyroid



(j) Winered

Figure 2: Comparison of the RAND performance between the ENMF under the mixed initialization and the NMF multiplicative update initialized by RI, RAI and CI for various datasets.



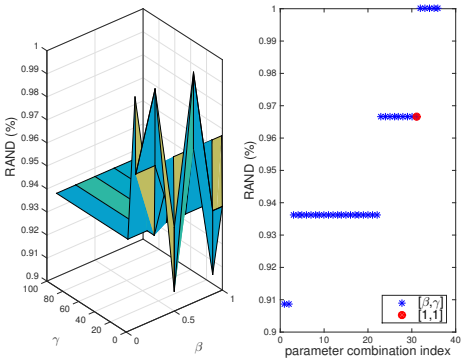
(a) Balance (b) Breast (c) WDBC

(d) BCWO (e) Dermatology (f) Glass

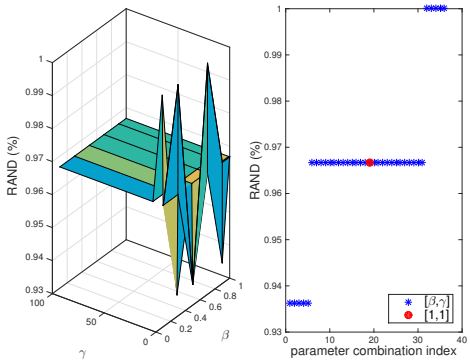
(g) Haberman (h) Iris (i) Thyroid

(j) Winered

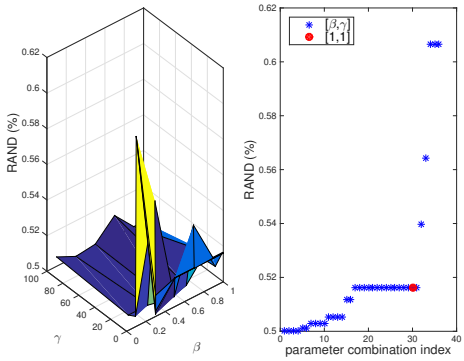
Figure 3: Comparison of the DI performance between the ENMF under the mixed initialization and the NMF multiplicative update initialized by RI, RAI and CI for various datasets.



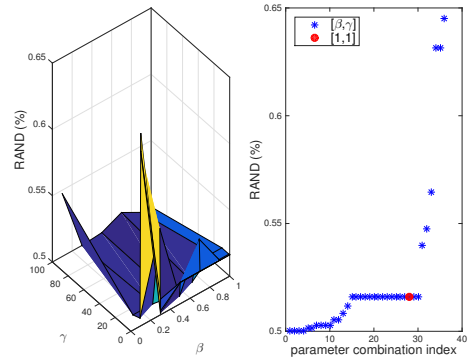
(a) Iris data at the 100th iteration.



(b) Iris data at the 500th iteration.

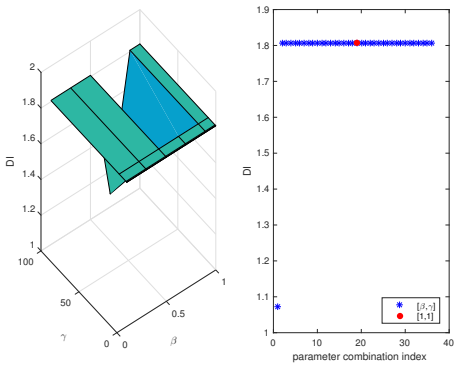


(c) Haberman data at the 100th iteration.

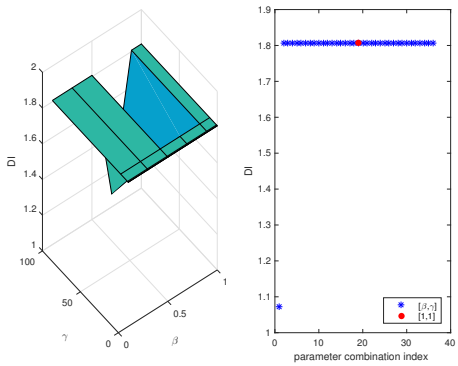


(d) Haberman data at the 500th iteration.

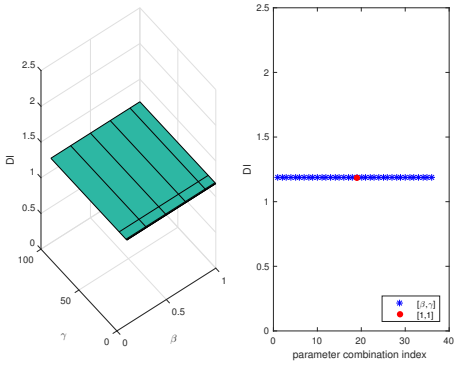
Figure 4: The left plot in each subfigure demonstrates the RAND performance of the ENMF at the 100th or 500th iteration under different parameter settings of $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ and $\gamma \in \{0.001, 0.01, 0.1, 1, 10, 100\}$ leading to a total of 36 parameter combinations. The right plot of each subfigure demonstrates the RAND performance distributions of all the 36 parameter combinations of $[\beta, \gamma]$. The marker "$\bullet$" indicates the parameter combination of $\beta = \gamma = 1$, while the markers "$*$" indicate the remaining combinations.



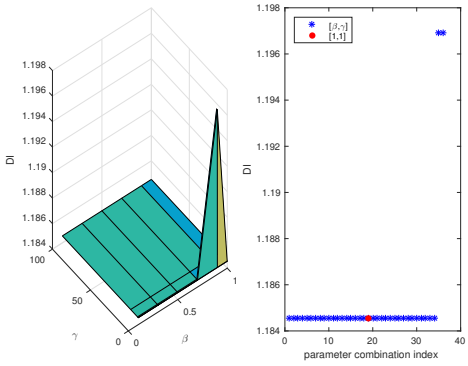
(a) Iris data at the 100th iteration.



(b) Iris data at the 500th iteration.

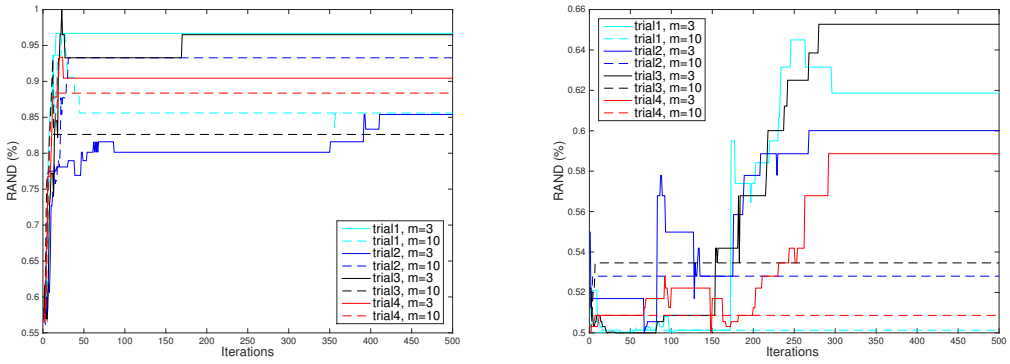


(c) Haberman data at the 100th iteration.



(d) Haberman data at the 500th iteration.

Figure 5: The left plot in each subfigure demonstrates the DI performance of the ENMF at the 100th or 500th iteration under different parameter settings of $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ and $\gamma \in \{0.001, 0.01, 0.1, 1, 10, 100\}$ leading to a total of 36 parameter combinations. The right plot of each subfigure demonstrates the DI performance distributions of all the 36 parameter combinations of $[\beta, \gamma]$. The marker "$\bullet$" indicates the parameter combination of $\beta = \gamma = 1$, while the markers "$*$" indicate the remaining combinations.



(a) Iris (b) Haberman

Figure 6: Comparison of the convergence rates of the ENMF under different initial population sizes of $m = 3$ and $m = 10$ using the RAI initialization for the iris and Haberman datasets. Four different training-test partitions are used, each referred to as a trial.