Online Nonnegative Matrix Factorization With Robust Stochastic Approximation

Naiyang Guan, Dacheng Tao, *Senior Member, IEEE*, Zhigang Luo, and Bo Yuan

***Abstract*— Nonnegative matrix factorization (NMF) has become a popular dimension-reduction method and has been widely applied to image processing and pattern recognition problems. However, conventional NMF learning methods require the entire dataset to reside in the memory and thus cannot be applied to large-scale or streaming datasets. In this paper, we propose an efficient online RSA-NMF algorithm (OR-NMF) that learns NMF in an incremental fashion and thus solves this problem. In particular, OR-NMF receives one sample or a chunk of samples per step and updates the bases via robust stochastic approximation. Benefitting from the smartly chosen learning rate and averaging technique, OR-NMF converges at the rate of $O(1/\sqrt{k})$ in each update of the bases. Furthermore, we prove that OR-NMF almost surely converges to a local optimal solution by using the quasi-martingale. By using a buffering strategy, we keep both the time and space complexities of one step of the OR-NMF constant and make OR-NMF suitable for large-scale or streaming datasets. Preliminary experimental results on real-world datasets show that OR-NMF outperforms the existing online NMF (ONMF) algorithms in terms of efficiency. Experimental results of face recognition and image annotation on public datasets confirm the effectiveness of OR-NMF compared with the existing ONMF algorithms.**

***Index Terms*— Nonnegative matrix factorization (NMF), online NMF (ONMF), robust stochastic approximation.**

I. Introduction

NONNEGATIVE matrix factorization (NMF) [1] has been widely applied to image processing and pattern recognition applications because these nonnegativity constraints on both the basis vectors and the combination coefficients allow only additive, not subtractive, combinations and lead to naturally sparse and parts-based representations. Since NMF does not allow negative entries in both basis vectors and coefficients, only additive combinations are allowed and such combinations are compatible with the intuitive notion of combining parts to form a whole, i.e., NMF learns parts-based representation. As the basis images are nonglobal and contain several versions of mouths, noses, and other facial parts, the basis images are sparse. Since any given face does not use all the available parts, the coefficients are also sparse. NMF learns sparse representation, but it is different from sparse coding because NMF learns low rank representation while sparse coding usually learns the full rank representation. Recently, several NMF-based dimension reduction algorithms have been proposed [2]–[6] and a unified framework [7] has been proposed to intrinsically understand them.

Several algorithms including the multiplicative update rule [8], [9], projected gradient descent [10], alternating nonnegative least squares [11], active set methods [12], and NeNMF [13] have been utilized to learn NMF. The multiplicative update rule iteratively updates the two matrix factors by the gradient descent method, whereby the learning rates are carefully chosen to guarantee the nonnegativity. Lin [10] applied the projected gradient descent method to NMF, which iteratively optimizes each matrix factor by descending along its projected gradient direction and the learning rate is chosen by Armjo's rule. Berry *et al.* [11] proposed to learn NMF by iteratively projecting the least-squares solution of each matrix factor onto the nonnegative orthant. Kim and Park [12] proposed to learn NMF by utilizing the active set method to iteratively optimize each matrix factor. Recently, Guan *et al.* [13] proposed an efficient NeNMF solver. These algorithms have been widely used to learn NMF and its extensions [8], [5], [14]. However, they share the following drawbacks: 1) the entire dataset resides in memory during the optimization procedure and 2) the time overheads are proportional to the product of the size of the dataset and number of iterations, and the number of iterations usually increases as the size of dataset increases. These drawbacks prohibit the use of NMF with large-scale problems, e.g., image search [15] and multiview learning [16], due to the high computational cost. In addition, these algorithms exclude NMF from streaming datasets because it is necessary to restart NMF on the arrival of new data.

In this paper, we propose an efficient online algorithm to learn NMF on large-scale or streaming datasets. By treating NMF as a stochastic optimization problem, we utilize a robust stochastic approximation (RSA) method to update the bases in an incremental manner. We term this method "online RSA-NMF," or OR-NMF for short in the remainder of this paper. In particular, OR-NMF receives one sample at each step and projects it onto the learned subspace and then updates the

Manuscript received July 23, 2011; accepted April 20, 2012. Date of publication May 22, 2012; date of current version June 8, 2012. This work was supported in part by the Australian Research Council Discovery Project with number DP-120103730, the National Natural Science Foundation of China under Grant 91024030/G03, and the Program for Changjiang Scholars and Innovative Research Team in University under Grant IRT1012.

N. Guan and Z. Luo are with the School of Computer Science, National University of Defense Technology, Changsha 410073, China (e-mail: ny guan@nudt.edu.cn; zgluo@nudt.edu.cn).

D. Tao is with the Center for Quantum Computation & Intelligent Systems and the Faculty of Engineering & Information Technology, University of Technology, Sydney, NSW 2007, Australia (e-mail: dacheng.tao@uts.edu.au).

B. Yuan is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: boyuan@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2197827

bases. Assuming the samples are distributed independently, the objective of updating the bases can be considered as a stochastic optimization problem and the problem is convex. It naturally motivates us to adopt Nemirovski's robust stochastic approximation method [17] to optimize the bases. In operations research, the robust stochastic approximation method [17] solves the stochastic optimization problem and guarantees a convergence rate of $O(1/\sqrt{k})$ by using an averaging technique and smartly choosing the learning rates. By proving minimums of the objective functions to be a quasi-martingale, we prove the convergence of OR-NMF. To keep the space complexity of OR-NMF constant, we introduce a buffer to store a limited number of samples, which makes OR-NMF to update the bases in constant time cost and storage at each step. Therefore, OR-NMF could be applied to large-scale or streaming datasets. We show that OR-NMF can be naturally extended to handle l_1-regularized and l_2-regularized NMF, NMF with the box constraint, and Itakura–Saito divergence-based NMF. Preliminary experiments on real-world datasets show that OR-NMF converges much faster than representative online NMF (ONMF) algorithms. Experimental results of face recognition and image annotation on public datasets show that the performance of OR-NMF is superior to those of other ONMF algorithms.

II. Online RSA-NMF

Given n samples $\{\vec{v}_1, \ldots, \vec{v}_n\} \in \mathbb{R}_+^m$ distributed in the probabilistic space $\mathrm{P} \in \mathbb{R}_+^m$, NMF learns a subspace $\mathrm{Q} \subset \mathrm{P}$ spanned by $r \ll \min\{m, n\}$ bases $\{\vec{w}_1, \ldots, \vec{w}_r\} \in \mathbb{R}_+^m$ to represent these samples, the problem can be written as

$$\min_{W \in \mathbb{R}_+^{m \times r}} f_n(W) = \frac{1}{n} \sum_{i=1}^{n} l(\vec{v}_i, W) \tag{1}$$

where the matrix W contains all the bases and

$$l(\vec{v}_i, W) = \min_{\vec{h}_i \in \mathbb{R}_+^r} \frac{1}{2} \left\| \vec{v}_i - W\vec{h}_i \right\|_2^2 \tag{2}$$

where $\vec{h}_i$ denotes the coefficient of $\vec{v}_i$. According to [18], one is usually not interested in minimizing the empirical cost $f_n(W)$, but instead in minimizing the expected cost

$$\min_{W \in \mathbb{R}_+^{m \times r}} f(W) = E_{\vec{v} \in \mathrm{P}}(l(\vec{v}, W)) \tag{3}$$

where $E_{\vec{v} \in \mathrm{P}}$ denotes the expectation over P.

This section proposes an efficient online algorithm to optimize (3), namely, it receives one sample, or a chunk of samples, at a time and incrementally updates the bases on the arrival of each sample or chunk.

A. Algorithm

Since (3) is nonconvex, it is impractical to solve it directly. Similar to most NMF algorithms, we solve it by recursively optimizing (2) with W fixed and optimizing (3) with $\vec{h}_i$s fixed. In particular, at step $t \geq 1$, on the arrival of sample $\vec{v}^t$, we obtain the corresponding coefficient $\vec{h}^t$ by using

$$\min_{\vec{h}^t \in \mathbb{R}_+^r} \frac{1}{2} \left\| \vec{v}^t - W^{t-1}\vec{h}^t \right\|_2^2 \tag{4}$$

where W^{t-1} is the previous basis matrix and W^0 is randomly initialized, followed by updating W^t

$$W^t = \arg\min_{W \in \mathbb{R}_+^{m \times r}} E_{\vec{v} \in \mathrm{P}_t} \left(\frac{1}{2} \left\| \vec{v} - W\vec{h} \right\|_2^2 \right) \tag{5}$$

where $\mathrm{P}_t \subset \mathrm{P}$ is the probabilistic subspace spanned by the arrived samples $\{\vec{v}^1, \ldots, \vec{v}^t\}$. Note that the corresponding coefficients $\{\vec{h}^1, \ldots, \vec{h}^t\}$ are available in the previous t steps. To simplify the following derivations, we denote $G_t(W, \vec{v}) = (1/2)\|\vec{v} - W\vec{h}\|_2^2$ $(\vec{v} \in \mathrm{P}_t)$, and then the objective function of (5) can be rewritten as

$$E_{\vec{v} \in \mathrm{P}_t} \left(\frac{1}{2} \left\| \vec{v} - W\vec{h} \right\|_2^2 \right) = E_{\vec{v} \in \mathrm{P}_t} \left(G_t(W, \vec{v}) \right) \triangleq g_t(W) \tag{6}$$

where $G_t(W, \vec{v})$ is convex with respect to W and $g_t(W)$ is bounded. According to [19], we have

$$\nabla_W g_t(W) = E_{\vec{v} \in \mathrm{P}_t}(\nabla_W G_t(W, \vec{v})) \tag{7}$$

where $\nabla_W g_t(W)$ and $\nabla_W G_t(W, \vec{v})$ are the gradient of $g_t(W)$ and the subgradient of $G_t(W, \vec{v})$, respectively.

Based on (7), we adopt the robust stochastic approximation (RSA) method to optimize (6). Recent results [17] in operations research show that the robust stochastic approximation method guarantees the convergence rate of $O(1/\sqrt{k})$ by smartly choosing the learning rates and making use of the averaging technique. RSA randomly generates N i.i.d. samples $\{\vec{v}_1, \ldots, \vec{v}_N\}$ and recursively updates W according to

$$W_{k+1} = \Pi_{\mathbf{C}}(W_k - r_k \nabla_W G_t(W_k, \vec{v}_k)), \quad k = 1, \ldots, N \tag{8}$$

where W_k is the generated sequence and $W_1 = W^{t-1}$, r_k is the learning rate, and $\Pi_{\mathbf{C}}(\cdot)$ denotes the orthogonal projector onto the domain $\mathbf{C}$. With the aim of eliminating the invariance of $W\vec{h}$ under the transformation $W \leftarrow W\Lambda$ and $\vec{h} \leftarrow \Lambda^{-1}\vec{h}$, where Λ is any positive diagonal matrix, we constrain each column of W to be a point on the simplex. Thus we have $\mathbf{C} = \{W = [\vec{w}_1, \ldots, \vec{w}_r], \|\vec{w}_j\|_1 = 1, \vec{w}_j \in \mathbb{R}_+^m, j = 1, \ldots, r\}$. Since it is often difficult to obtain such i.i.d samples $\vec{v}_1, \ldots, \vec{v}_N$, they are approximated by cycling on a randomly permuted sequence of the existing samples $\vec{v}^1, \ldots, \vec{v}^t$ in practice, where their coefficients $\vec{h}^1, \ldots, \vec{h}^t$ are obtained in the previous t steps. According to [17], the average solution $\overline{W}_k = \sum_{j=1}^k r_j W_j / \sum_{j=1}^k r_j$ converges almost surely to the optimal solution of (6) by using the following learning rate schema $r_k = \theta^t D_W / M_* \sqrt{k}$, wherein $D_W = \max_{W \in \mathbf{C}} \|W - W_1\|_F$ and $M_* = \sup_{W \in \mathbf{C}} E_{\vec{v}}^{(1/2)}(\|\nabla_W G_t(W, \vec{v})\|_F^2)$, and θ^t is a positive scaling constant of the learning rate at the t-th update. According to [17], D_W is the diameter of $\mathbf{C}$ and M_* is estimated in the following section. The statistical optimization method has been applied to NMF [20] for finding a local optimal solution, but it was designed for the batch NMF. To the best of our knowledge, this is the first attempt to introduce the statistical optimization method to learn NMF in an online fashion.

We summarize the proposed online RSA-NMF algorithm (OR-NMF) in Algorithm 1, and the procedure of updating basis matrix in Algorithm 2. To store the arrived samples and their coefficients, we introduce a set $\mathbf{B}$ in Algorithm 1 and initialize it to be empty (see Statement 1). On the arrival of the

TABLE I
SUMMARY OF ALGORITHMS

Algorithm 1: Online RSA-NMF	**Algorithm 2**: Update basis matrix	**Algorithm 3**: Modified OR-NMF
Input: $\vec{v} \in \mathbb{R}_+^m \sim$ P (samples from P), T (sampling time), r (reduced dimensionality)	**Input**: W^{t-1} (previous basis matrix), θ^t (learning rate scaling constant), $\mathbf{B}$ (sample buffer), τ (tolerance)	**Input**: $\vec{v} \in \mathbb{R}_+^m \sim$ P (samples from P), T (sampling time), r (reduced dimensionality), l (buffer length)
Output: $W \in \mathbb{R}_+^{m\times r}$ (learned basis)	**Output**: $W^t \in \mathbb{R}_+^{m\times r}$ (updated basis matrix)	**Output**: $W \in \mathbb{R}_+^{m\times r}$ (learned basis)
1: $W^0 \in \mathbb{R}_+^{m\times r}$ and $\mathbf{B} \leftarrow \varnothing$	1: $\Sigma_W \leftarrow 0$, $\Sigma \leftarrow 0$, $W_1/W_1^t \leftarrow W^{t-1}$, $k \leftarrow 1$	1: $W^0 \in \mathbb{R}_+^{m\times r}$ and $\mathbf{B} \leftarrow \varnothing$
For $t = 1$ to T do	**Repeat**	**For** $t = 1$ to T do
2: Draw $\vec{v}^t$ from P	2: $r_k = \theta^t D_W / M_* \sqrt{k}$	2: Draw $\vec{v}^t$ from P
3: Calculate $\vec{h}^t$ by (4)	3: $\Sigma_W \leftarrow \Sigma_W + r_k W_k$, $\Sigma \leftarrow \Sigma + r_k$	3: Calculate $\vec{h}^t$ by (4)
4: Add $\vec{v}^t$ to $\mathbf{B}$	4: Draw $\vec{v}_k$ by cycling on permuted $\mathbf{B}$	4: Replace $\vec{v}^{t-l}$ with $\vec{v}^t$ in $\mathbf{B}$ when $t > l$
5: $\theta^t = 0.1\cos\left(\frac{(t-1)\pi}{2T}\right)$	5: $W_{k+1} = \Pi_{\mathbf{C}}(W_k - r_k \nabla_W G_t(W_k, \vec{v}_k))$	5: $\theta^t = 0.1\cos\left(\frac{(t-1)\pi}{2T}\right)$
6: Update W^t with **Algorithm 2**	6: $W_{k+1}^t = \Sigma_W / \Sigma$, $k \leftarrow k+1$	6: Update W^t with **Algorithm 2**
End For	**Until** {Stopping Criterion (9) is Satisfied}	**End For**
7: $W = W^T$	7: $W^t = W_K^t$	7: $W = W^T$

tth sample $\vec{v}^t$, both $\vec{v}^t$ and its coefficient $\vec{h}^t$ are added to the set $\mathbf{B}$ (see Statement 4). The samples in $\mathbf{B}$ will be used to update the basis matrix (Statements 4 and 5 in Algorithm 2). Since all the arrived samples reside in the memory, the space complexity of Algorithm 1 dramatically increases as the sample number increases, we will show how to reduce this complexity in the following section.

In Algorithm 2, the basis matrix is updated by averaging the generated points (see Statement 6) and the updating procedure is stopped at convergence. Since the objective function (5) contains expectation operator, it is hard to check the stopping condition directly. In contrast to [21], which uses $\|W_k^t - W_{k-1}^t\|_F \le \tau$ as a stopping criterion, we consider the following normalized stopping criterion:

$$\frac{\|W_k^t - W_{k-1}^t\|_F}{\|W_{k-1}^t\|_F} \le \tau \tag{9}$$

where W_k^t denotes the kth point generated by Statement 6 and τ is the tolerance which is usually set to a small value, e.g., 10^{-3}. We use the normalized stopping criterion to make it independent of the size of W. In addition, both D_W and M_* are crucial parameters in constructing the learning rate (see Statement 2). According to [17], D_W is equal to the diameter of the domain $\mathbf{C}$, which is $\sqrt{2r}$ because the diameter of the simplex $X = \{\vec{w}, \|\vec{w}\|_1 = 1, \vec{w} \in \mathbb{R}_+^m\}$ is $\sqrt{2}$. By the definition of M_*, i.e., $M_* = \sup_{W\in\mathbf{C}} E_{\vec{v}}^{(1/2)}(\|\nabla_W G_t(W, \vec{v})\|_F^2)$, we adaptively update M_* with the maxima of $\|\nabla_W G_t(W_k, \vec{v}_k)\|_F$ during the optimization procedure in (8). The experimental results show that this strategy works well. Another crucial parameter in Statement 2 is the scaling constant θ^t. According to [17], it brings a constant $\max\{\theta^t, \theta^{-1}\}$ in the convergence rate of Algorithm 2. We adaptively set $\theta^t = 0.1\cos((t-1)\pi/2T)$ in our experiments, and the results show that this strategy performs well.

The main time cost of Algorithm 1 is spent on Statement 6, which updates the basis matrix with Algorithm 2. Since both the gradient $\nabla_W G_t(W_k, \vec{v}_k)$ and the projection operator $\Pi_{\mathbf{C}}$ in (8) can be calculated in $O(mr)$ time, the time complexity of Algorithm 2 is $O(mrK)$, where K is the iteration number. Note that the time complexity of Algorithm 2 stays constant because it has no relation to the sample number t.

B. Convergence Analysis

The convergence of OR-NMF (see Algorithm 1) is proved in Theorem 1. Note that the modified OR-NMF with buffering strategy (see Algorithm 3) can be proved in a similar way.

Theorem 1: The objective function $f(W)$ converges almost surely under Algorithm 1.

Proof: Similar to [22], We prove the convergence of $f(W)$ by using the following three steps: 1) proving that $g_t(W^t)$ converges almost surely; 2) showing that $\|W^{t+1} - W^t\|_F = O(1/t)$; and 3) proving that $f(W^t) - g_t(W^t)$ almost surely converges to zero as t goes to infinity. However, it is worth emphasizing that the proof strategies used in this theorem are very different from those used in [22].

Step 1 is completed by showing that the stochastic positive process, i.e., $g_t(W^t) \ge 0$, is a quasi-martingale. Considering the variations of $g_t(W^t)$, we have

$$g_{t+1}\left(W^{t+1}\right) - g_t\left(W^t\right) = g_{t+1}\left(W^{t+1}\right) - g_{t+1}\left(W^t\right) + g_{t+1}\left(W^t\right) - g_t\left(W^t\right) \tag{10}$$

where $g_t(W^t)$ and $g_{t+1}(W^t)$ are defined in (6). By the law of large numbers, we can rewrite $g_{t+1}(W^t)$ as

$$g_{t+1}\left(W^t\right) = E_{\vec{v}\in \mathrm{P}_{t+1}}\left(G_{t+1}\left(W^t, \vec{v}\right)\right) = \lim_{n\to\infty} \frac{\sum_{k=1}^n \frac{1}{2}\|\vec{v}_k - W^t \vec{h}_k\|_2^2}{n} \quad \text{a.s.} \tag{11}$$

where $\vec{v}_k$ and $\vec{h}_k$ are obtained by randomly cycling on the existing samples $\vec{v}^1, \ldots, \vec{v}^{t+1}$ and the corresponding coefficients $\vec{h}^1, \ldots, \vec{h}^{t+1}$, respectively. Since all the existing samples are fairly chosen in Algorithm 2, (11) is equivalent to

$$g_{t+1}\left(W^t\right) = \frac{t E_{\vec{v}\in \mathrm{P}_t}\left(G_t\left(W^t, \vec{v}\right)\right) + l\left(\vec{v}^{t+1}, W^t\right)}{t+1}$$

$$= \frac{t g_t\left(W^t\right) + l\left(\vec{v}^{t+1}, W^t\right)}{t+1} \quad \text{a.s.} \tag{12}$$

where $\vec{v}^{t+1}$ is the newly arrived sample at the $t+1$th step. By substituting (12) into (10) and using some algebra, we have

$$g_{t+1}\left(W^{t+1}\right) - g_t\left(W^t\right) = g_{t+1}\left(W^{t+1}\right) - g_{t+1}\left(W^t\right) + \frac{l\left(\vec{v}^{t+1}, W^t\right) - g_t\left(W^t\right)}{t+1}. \tag{13}$$

Since W^{t+1} minimizes $g_{t+1}(W)$ on $\mathbf{C}$, $g_{t+1}(W^{t+1}) \leq g_{t+1}(W^t)$. By filtering the past information $F_t = \{\vec{v}^1, \ldots, \vec{v}^t; \vec{h}^1, \ldots, \vec{h}^t\}$ and taking the expectation over both sides of (13), we have

$$\begin{aligned} & E\left[g_{t+1}\left(W^{t+1}\right) - g_t\left(W^t\right) | F_t\right] \\ & \quad \leq \frac{E\left[l\left(\vec{v}^{t+1}, W^t\right) | F_t\right] - g_t\left(W^t\right)}{t+1} \\ & \quad \leq \frac{f\left(W^t\right) - f_t\left(W^t\right)}{t+1} \leq \frac{\left\| f\left(W^t\right) - f_t\left(W^t\right)\right\|_\infty}{t+1} \end{aligned} \tag{14}$$

where $f_t(W^t)$ signifies the empirical approximation of $f(W^t)$, i.e., $f_t(W^t) = (\sum_{i=1}^t l(\vec{v}^i, W^t)/t)$, and the second inequality comes from the fact

$$\begin{aligned} g_t(W^t) &= \lim_{n\to\infty} \frac{\sum_{k=1}^n \frac{1}{2}\left\|\vec{v}_k - W^t\vec{h}_k\right\|_2^2}{n} \\ &\geq \lim_{n\to\infty} \frac{\sum_{k=1}^n l(\vec{v}_k, W^t)}{n} \geq f_t(W^t) \end{aligned}$$

where $(1/2)\|\vec{v}_k - W^t\vec{h}_k\|_2^2 \geq l(\vec{v}_k, W^t)$ according to (4). Since $l(\vec{v}, W)$ is Lipschitz-continuous and bounded and $E_{\vec{v}}[l^2(\vec{v}, W)]$ is uniformly bounded [22], according to the corollary of the Donsker theorem [23], we have

$$E\left[\|\sqrt{t}(f(W^t) - f_t(W^t))\|_\infty\right] = O(1). \tag{15}$$

By taking expectation over both sides of (14) and combining it with (15), there exists a constant $K_1 > 0$ such that

$$E\left[E[g_{t+1}(W^{t+1}) - g_t(W^t)|F_t]^+\right] \leq \frac{K_1}{\sqrt{t}(t+1)}. \tag{16}$$

By letting $t \to \infty$ and summing up all the inequalities derived from (16), we arrive at

$$\begin{aligned} \sum_{t=1}^\infty E & \left[E\left[g_{t+1}\left(W^{t+1}\right) - g_t\left(W^t\right) | F_t\right]^+\right] \\ &= \sum_{t=1}^\infty E\left[\delta_t\left(g_{t+1}\left(W^{t+1}\right) - g_t\left(W^t\right)\right)\right] \\ &= \sum_{t=1}^\infty \frac{K_1}{\sqrt{t}\,(t+1)} < \infty \end{aligned}$$

where δ_t is defined as in [24]. Since $g_t(W^t) > 0$, according to [24], we prove that $g_t(W^t)$ is a quasi-martingale and converges almost surely. As a byproduct, we have

$$\sum_{t=1}^\infty \left| E\left[g_{t+1}\left(W^{t+1}\right) - g_t\left(W^t\right) \middle| F_t\right]\right| < \infty \quad \text{a.s.} \tag{17}$$

In order to prove $f(W^t)$ converges in Step 3, we first prove $\|W^{t+1} - W^t\|_F = O(1/t)$ in this step (2). For the convenience of presentation, we denote the variation of $g_t(W)$ as

$$u_t(W) = g_t(W) - g_{t+1}(W).$$

Since $g_{t+1}(W^{t+1}) \leq g_{t+1}(W^t)$, we have

$$\begin{aligned} & g_t\left(W^{t+1}\right) - g_t\left(W^t\right) \\ & \quad = g_t\left(W^{t+1}\right) - g_{t+1}\left(W^{t+1}\right) + g_{t+1}\left(W^{t+1}\right) - g_{t+1}\left(W^t\right) \\ & \qquad + g_{t+1}\left(W^t\right) - g_t\left(W^t\right) \\ & \quad \leq g_t\left(W^{t+1}\right) - g_{t+1}\left(W^{t+1}\right) + g_{t+1}\left(W^t\right) - g_t\left(W^t\right) \\ & \quad = u_t\left(W^{t+1}\right) - u_t\left(W^t\right). \end{aligned} \tag{18}$$

Considering the gradient of $u_t(W)$

$$\begin{aligned} \nabla_W u_t(W) &= \nabla_W g_t(W) - \nabla_W g_{t+1}(W) \\ &= \frac{1}{t}\left(\frac{\sum_{i=1}^t \vec{v}_i\vec{h}_i^T - t\vec{v}_{t+1}\vec{h}_{t+1}^T}{t+1} \right. \\ &\qquad \left. - W\frac{\sum_{i=1}^t \vec{h}_i\vec{h}_i^T - t\vec{h}_{t+1}\vec{h}_{t+1}^T}{t+1}\right). \end{aligned} \tag{19}$$

Since $W \in \mathbf{C}$, $\|W\|_F < \sqrt{r}$. By the triangle inequality, (19) implies

$$\begin{aligned} \|\nabla_W u_t(W)\|_F \leq \frac{1}{t}\left(\left\|\frac{\sum_{i=1}^t \vec{v}_i\vec{h}_i^T - t\vec{v}_{t+1}\vec{h}_{t+1}^T}{t+1}\right\|_F \right. \\ \left. + \sqrt{r}\left\|\frac{\sum_{i=1}^t \vec{h}_i\vec{h}_i^T - t\vec{h}_{t+1}\vec{h}_{t+1}^T}{t+1}\right\|_F\right) = L_t \end{aligned}$$

where $L_t = O(1/t)$. Therefore, $u_t(W)$ is Lipschitz-continuous with constant L_t. By the triangle inequality, from (18), we have

$$\left\|g_t\left(W^{t+1}\right) - g_t\left(W^t\right)\right\|_F \leq L_t\left\|W^{t+1} - W^t\right\|_F. \tag{20}$$

Note that $g_t(W)$ is convex, it is reasonable to further assume that its Hessian has a low bound K_2

$$\left\|g_t(W^{t+1}) - g_t(W^t)\right\|_F \geq K_2\|W^{t+1} - W^t\|_F^2. \tag{21}$$

By combining (20) and (21), we arrive at

$$\|W^{t+1} - W^t\|_F \leq \frac{L_t}{K_2}. \tag{22}$$

Therefore, we prove that $\|W^{t+1} - W^t\|_F = O(1/t)$.

Step 3 proves that $f(W^t) - g_t(W^t)$ almost surely converges to zero as t goes to infinity. From (13), we get

$$\begin{aligned} \frac{g_t(W^t) - f_t(W^t)}{t+1} &\leq g_t(W^t) - g_{t+1}(W^{t+1}) \\ &\quad + \frac{l(\vec{v}^{t+1}, W^t) - f_t(W^t)}{t+1}. \end{aligned} \tag{23}$$

By taking expectation and absolute over both sides of (23) and summing up all the inequalities derived from (23) as t goes to infinity, we have

$$\sum_{t=1}^\infty \frac{g_t(W^t) - f_t(W^t)}{t+1} < \sum_{t=1}^\infty \left|E\left[g_{t+1}(W^{t+1}) - g_t(W^t)F_t\right]\right|$$

$$+\sum_{t=1}^{\infty}\frac{\|f(W^t)-f_t(W^t)\|_\infty}{t+1} \quad (24)$$

where the inequality comes from the triangle inequality and $g_t(W^t) \geq f_t(W^t)$. From (15), (17), and (24), we conclude that

$$\sum_{t=1}^{\infty}\frac{g_t(W^t)-f_t(W^t)}{t+1} < \infty \quad \text{a.s.} \quad (25)$$

Since both $g_t(W)$ and $f_t(W)$ are Lipschitz continuous, there exists a constant $K_3 > 0$ such that

$$\left|g_{t+1}\left(W^{t+1}\right)-f_{t+1}\left(W^{t+1}\right)-\left(g_t\left(W^t\right)-f_t\left(W^t\right)\right)\right| \leq K_3\|W^{t+1}-W^t\|_F. \quad (26)$$

Since $\|W^{t+1}-W^t\|_F = O(1/t)$ (see Step 2), according to [25], (25) and (26) imply that

$$g_t(W^t)-f_t(W^t)\to 0 \ (t\to\infty), \quad \text{a.s.}$$

Since $f(W^t)-f_t(W^t)\to 0$ $(t\to\infty)$, a.s., we prove that $f(W^t)$ converges almost surely. It completes the proof. ■

C. Buffering Strategy

Since the space complexity of Algorithm 1, i.e., $O(mt + rt + mr)$, increases dramatically as t increases to infinity, it is unacceptable especially for large-scale or streaming datasets because there is insufficient memory in practice to retain the existing samples in set **B**. Therefore, we regard **B** as a buffer and make it store the l most recently arrived samples and thus reduce the space complexity of Algorithm 1. In particular, at step t, we update the buffer **B** by replacing the oldest sample $\vec{v}^{t-l}$ with the new coming sample $\vec{v}^t$ when $t > l$, wherein l is the prefixed buffer length. By using this strategy, we reduce the space complexity of Algorithm 1 to $O(ml + rl + mr)$. We summarize the modified OR-NMF (MOR-NMF) in Algorithm 3 by replacing Statement 4 in Algorithm 1.

As the space complexity of Algorithm 3 stays constant, it performs efficiently in practice especially on large-scale or streaming datasets. Empirical experiments on real-world datasets show that Algorithm 3 converges and works well in various applications. In addition, the buffer length l is a critical parameter in Algorithm 3. It should be selected in a reasonable range for the following two reasons: 1) l cannot be too large because it controls the space complexity of Algorithm 3 and 2) l should be set large enough to make the buffer **B** provide sufficient samples $\vec{v}^{t-l+1},\ldots,\vec{v}^t$ for the permutation procedure in Statement 4 in Algorithm 2. In our experiments, we empirically select l by using the strategy $l = \min\{\lceil(n/10)\rceil, 20\}$, wherein $\lceil x\rceil$ signifies the smallest integer larger than x.

D. Mini-Batch Extension

We can improve the convergence speed of the proposed Algorithm 1 by receiving a chunk of samples instead of a single one, i.e., a total of j samples $\vec{v}^{(t-1)j+1},\ldots,\vec{v}^{tj}$ are drawn from the distribution P in iteration t. Their coefficients $\vec{h}^{(t-1)j+1},\ldots,\vec{h}^{tj}$ are obtained by optimizing the following objective:

$$\min_{H^t\geq 0}\frac{1}{2}\|V^t - W^{t-1}H^t\|_F^2 \quad (27)$$

where $V^t = [\vec{v}^{(t-1)j+1},\ldots,\vec{v}^{tj}]$ and $H^t = [\vec{h}^{(t-1)j+1},\ldots,\vec{h}^{tj}]$. Many methods, such as the projected gradient descent [10] and block principal pivoting [12] whose complexities are not linear in j, can be applied to efficiently solve (27). In this case, the proposed Algorithm 1 can be conveniently extended by calculating the coefficients of the sample chunk by optimizing (27) instead of (4) in Statement 3 and adding this sample chunk to set **B** in Statement 4. We call this extension as "window-based OR-NMF" (WOR-NMF). In addition, Algorithm 3 can also be conveniently extended in this case by replacing the oldest sample chunk $\vec{v}^{j(t-1)-jl+1},\ldots,\vec{v}^{j(t-1)-jl+j}$ with the new coming sample chunk when $t > l$.

III. OR-NMF for NMF Extensions

This section shows that OR-NMF can be conveniently adopted for handling l_1-regularized and l_2-regularized NMF, NMF with box constraint, and Itakura–Saito (IS) divergence-based NMF.

A. l_1-Regularized NMF

Although NMF obtains sparse representation, this sparsity is not explicitly guaranteed. Hoyer [26] proposed incorporating the l_1-regularization on the coefficients. The objective is

$$\min_{W\in\mathbb{R}_+^{m\times r}} f_n(W) = \frac{1}{n}\sum_{i=1}^{n} l_1(\vec{v}_i, W) \quad (28)$$

where $l_1(\vec{v}_i, W) = \min_{\vec{h}\in\mathbb{R}_+^r}(1/2)\|\vec{v}_i - W\vec{h}\|_2^2 + \lambda\|\vec{h}\|_1$ and λ is the tradeoff parameter. The problem (28) can be easily optimized by extending OR-NMF, with Statement 3 in Algorithm 1 replaced by $\vec{h}_t = \arg\min_{\vec{h}\in\mathbb{R}_+^r}(1/2)\|\vec{v}_t - W_t\vec{h}\|_2^2 + \lambda\|\vec{h}\|_1$.

B. l_2-Regularized NMF

The l_2-regularization, i.e., Tikhonov regularization, is usually utilized to control the smoothness of the solution in NMF [14]. The objective function can be written as

$$\min_{W\in\mathbb{R}_+^{m\times r}} f_n(W) = \frac{1}{n}\sum_{i=1}^{n} l_2(\vec{v}_i, W) \quad (29)$$

where $l_2(\vec{v}_i, W) = \min_{\vec{h}\in\mathbb{R}_+^r}((1/2)\|\vec{v}_i - W\vec{h}\|_2^2 + (\alpha/2)\|W\|_F^2 + (\beta/2)\|\vec{h}\|_2^2$, and $\|\cdot\|_F$ is the matrix Frobenius norm and α, β are tradeoff parameters. The problem (29) can be solved by naturally extending OR-NMF. On the arrival of sample $\vec{v}_t$, its coefficient $\vec{h}_t$ can be optimized by $\vec{h}_t = \arg\min_{\vec{h}\in\mathbb{R}_+^r}(1/2)\|\vec{v}_t - W_t\vec{h}\|_2^2 + (\beta/2)\|\vec{h}\|_2^2$. Given samples $\{\vec{v}_1,\ldots,\vec{v}_t\}$ and their corresponding coefficients $\{\vec{h}_1,\ldots,\vec{h}_t\}$, the basis matrix W_{t+1} is optimized by $W^{t+1} = \arg\min_{W\in\mathbb{R}_+^{m\times r}} g_2(W) = E_{\vec{v}}(G_2(W,\vec{v}))$, wherein $G_2(W,\vec{v}) = (1/2)\|\vec{v} - W\vec{h}\|_2^2 + (\alpha/2)\|W\|_F^2$. It is obvious that $G_2(W,\vec{v})$ is convex, and thus OR-NMF can be extended to solve (29) by replacing $\nabla_W G(W_k,\vec{v}_k)$ in Statement 5 in Algorithm 2 with $\nabla_W G_2(W_k,\vec{v}_k)$.

C. NMF With Box Constraint

Although OR-NMF is designed to solve NMF, it can be naturally extended to solve the box-constrained optimization problem [25]

$$\min_{L\le W\le U} f_n(W)=\frac{1}{n}\sum_{i=1}^{n} l_b(\vec{v}_i, W) \tag{30}$$

where both L and U have the same dimensionality as W and the constraint means $L_{ij}\le W_{ij}\le U_{ij}$. By replacing the feasible set $\mathbf{C}$ in (8) with $\mathbf{C}_b=\{W|L_{ij}\le W_{ij}\le U_{ij}, \forall ij\}$, the basis matrix W_{t+1} is obtained by $W^{t+1}=\arg\min_{W\in\mathbf{C}_b} g(W)=E_{\vec{v}}(G(W,\vec{v}))$. It is obvious that $G(W,\vec{v})$ is convex on $\mathbf{C}_b$, and thus OR-NMF can be extended to solve (30) by slightly modifying the projection operator (see Statement 5 in Algorithm 2).

Beside the l_1 and l_2-regularizations and box constraint, OR-NMF can also be extended to handle other regularizations, e.g., manifold regularization [27], [28] and regularizations for transfer learning [29], [31], for data representation. With the aim of acceleration, the active learning based manifold regularization [32] can also be applied to OR-NMF. Due to limitations of space, we postpone these discussions to future publications.

D. IS Divergence-Based NMF

OR-NMF uses the l_2-norm to measure the approximation error. This section shows that OR-NMF can also be extended to optimize the IS divergence-based NMF [33]

$$\min_{W\in\mathbb{R}_+^{m\times r}} \frac{1}{n}\sum_{j=1}^{n} l_{IS}(\vec{v}_j, W) \tag{31}$$

where $l_{IS}(\vec{v}_j, W)=\min_{\vec{h}_j\in\mathbb{R}_+^r} d_{IS}(\vec{v}_j, W\vec{h}_j)$ and the IS divergence is defined by

$$d_{IS}(\vec{x},\vec{y})=\sum_{i=1}^{m}\left(\frac{\vec{x}_i}{\vec{y}_i}-\log\frac{\vec{x}_i}{\vec{y}_i}-1\right). \tag{32}$$

When the sample number increases to infinity, (31) becomes an expectation of the form $\min_{W\in\mathbb{R}_+^{m\times r}} E_{\vec{v}\in\mathrm{P}}(l_{IS}(\vec{v}, W))$, which can be solved in an incremental manner by using OR-NMF.

On the arrival of the sample $\vec{v}_t$, its coefficient $\vec{h}_t$ can be optimized by

$$\vec{h}_t=\arg\min_{\vec{h}_t\in\mathbb{R}_+^r} d_{IS}(\vec{v}_t, W_t\vec{h}_t). \tag{33}$$

According to [33], the problem (33) can be solved by iteratively updated $\vec{h}_t$ from a random initial point through the following multiplicative rule until convergence: $\vec{h}_t\leftarrow\vec{h}_t\otimes(W_t^T\times(\vec{v}_t/(W_t\vec{h}_t)^2))/(W_t^T\times(1/W_t\vec{h}_t))$, where $\otimes$ signifies the element-wise product. Given samples $\{\vec{v}_1,\ldots,\vec{v}_t\}$ and their corresponding coefficients $\{\vec{h}_1,\ldots,\vec{h}_t\}$, the basis matrix is updated by

$$W_{t+1}=\arg\min_{W\in\mathbb{R}_+^{m\times r}} E_{\vec{v}\in\mathrm{P}}(d_{IS}(\vec{v}, W\vec{h})). \tag{34}$$

Since $d_{IS}(\vec{v}, W\vec{h})$ is convex with respect to W, (34) is convex with respect to W, and thus OR-NMF can be extended to solve (34). We call such extended algorithm for IS-NMF as "OR-NMF-IS." In particular, OR-NMF-IS replaces the Statement 4 in Algorithms 1 and 3 with (33), and replaces $\nabla_W G_t(W_t,\vec{v}_t)$ in the Statement 5 in Algorithm 2 with $\nabla_W d_{IS}(\vec{v}_t, W_t\vec{h}_t)=((W_t\vec{h}_t-\vec{v}_t)/(W_t\vec{h}_t)^2)\times\vec{h}_t^T$.

Note that Lefèvre *et al.* [21] proposed the ONMF-IS algorithm to incrementally learn IS-NMF, which will be reviewed in the following section. OR-NMF-IS is different from ONMF-IS and the experimental results show that OR-NMF-IS outperforms ONMF-IS in terms of efficiency.

IV. Related Works

This section briefly reviews the existing ONMF algorithms including ONMF [34], ONMF-IS [21], incremental NMF [35], [36], and online matrix factorization [22], [37] and presents their differences from the proposed OR-NMF.

A. ONMF

Cao *et al.* [34] proposed an ONMF which finds the two matrix factors, i.e., W and H, to approximate the whole data matrix $[V,U]\in\mathbb{R}_+^{m\times n}$, wherein m and n denote the dimensionality and number of samples, respectively, and V and U are the old and new sample matrix, respectively. ONMF assumes that V is approximated by $W_{\mathrm{old}}H_{\mathrm{old}}$. The problem is to find $W\in\mathbb{R}_+^{m\times r}$ and $H\in\mathbb{R}_+^{r\times n}$ such that

$$[V,U]\approx WH\triangleq W[H_1,H_2]. \tag{35}$$

Based on the relation $WH_1\approx W_{\mathrm{old}}H_{\mathrm{old}}$, Cao *et al.* [34] proved that a matrix Γ exists that satisfies $W=W_{\mathrm{old}}\Gamma^{-1}, H_1=\Gamma H_{\mathrm{old}}$. ONMF smartly carries out NMF on $[W_{\mathrm{old}}\Lambda, U]$, wherein Λ is positive diagonal matrix

$$[W_{\mathrm{old}}\Lambda, U]\approx W_{\mathrm{new}}\left[H_{\mathrm{new}}^1, H_{\mathrm{new}}^2\right] \tag{36}$$

which implies $W_{\mathrm{old}}\approx W_{\mathrm{new}}H_{\mathrm{new}}^1\Lambda^{-1}$ and $U\approx W_{\mathrm{new}}H_{\mathrm{new}}^2$. By setting $W=W_{\mathrm{new}}$ and $\Gamma=H_{\mathrm{new}}^1\Lambda^{-1}$, we have

$$W=W_{\mathrm{new}}, H=\left[H_{\mathrm{new}}^1\Lambda^{-1}H_{\mathrm{old}}, H_{\mathrm{new}}^2\right]. \tag{37}$$

It is suggested that Λ be calculated by $\Lambda_{jj}=\|\vec{h}_{\mathrm{old}}^j\|_2$, wherein $\vec{h}_{\mathrm{old}}^j$ is the jth column of H_{old} [34].

Because the time overhead of (36) is much lower than that of (35), ONMF performs well in practice. However, the space complexity of ONMF, $O(mn+rn)$, increases dramatically as n increases. Thus it cannot be applied to large-scale or streaming datasets due to the memory limitation.

B. ONMF-IS

Lefèvre *et al.* [21] proposed ONMF-IS to incrementally learn IS divergence based NMF. The objective function is

$$\min_{W\ge 0} L_t(W)=\frac{1}{t}\sum_{j=1}^{t} d_{IS}(\vec{v}_j, W\vec{h}_j) \tag{38}$$

where d_{IS} is the IS divergence defined in (32), and the infinite sequences $\{\vec{v}_1,\ldots,\vec{v}_t,\ldots\}$ and $\{\vec{h}_1,\ldots,\vec{h}_t,\ldots\}$ contain samples and their corresponding coordinates obtained as $\vec{h}_t=\arg\min_{\vec{h}} d_{IS}(\vec{v}_t, W_t\vec{h})$.

By extending the batch algorithm for IS-NMF, (38) is optimized by recursively updating the following rules:

$$A_t = A_{t-1} + \left(\frac{\vec{v}_t}{(W_{t-1}\vec{h}_t)^2}\vec{h}_t^T\right) \otimes W_{t-1}^2$$

$$B_t = B_{t-1} + \frac{1}{W_{t-1}\vec{h}_t} \times \vec{h}_t^T, \quad W_t = \sqrt{\frac{A_t}{B_t}}$$

where W_0, A_0, and B_0 are randomly initialized. ONMF-IS is simple and scalable for processing long-time audio sequences. However, we are not sure how to use ONMF-IS to optimize the Frobenius-norm-based NMF.

C. Incremental Nonnegative Matrix Factorization (INMF)

Bucak and Gunsel [35] proposed an INMF for learning the NMF problem. On arrival of the $(k+1)$th sample, INMF updates the basis matrix W_{k+1} by optimizing the following objective function:

$$\min_{W_{k+1}\geq 0, \vec{h}_{k+1}\geq 0} \quad S_{\text{old}}\frac{1}{2}\|V_k - W_{k+1}H_k\|_2^2 + S_{\text{new}}\frac{1}{2}\|\vec{v}_{k+1} - W_{k+1}\vec{h}_{k+1}\|_2^2 \tag{39}$$

where $V_k = [\vec{v}_1, \ldots, \vec{v}_k]$ contains the old samples, $H_k = [\vec{h}_1, \ldots, \vec{h}_k]$ contains their coefficients, and $\vec{v}_{k+1}$ denotes the newly arrived sample. As a byproduct, (39) obtains the coefficient $\vec{h}_{k+1}$ of $\vec{v}_{k+1}$. In (39), S_{old} and S_{new} are the weights to balance the contributions of the old samples and new sample, and $S_{\text{old}} + S_{\text{new}} = 1$. To solve (39), INMF iteratively updates $\vec{h}_{k+1}$ and W_{k+1} with the following rule until convergence:

$$\vec{h}_{k+1} \leftarrow \vec{h}_{k+1} \otimes \frac{W_{k+1}^T\vec{v}_{k+1}}{W_{k+1}^T W_{k+1}\vec{h}_{k+1}}$$

$$W_{k+1} \leftarrow W_{k+1} \otimes \frac{S_{\text{old}}V_kH_k^T + S_{\text{new}}\vec{v}_{k+1}\vec{h}_{k+1}^T}{S_{\text{old}}W_{k+1}H_kH_k^T + S_{\text{new}}W_{k+1}\vec{h}_{k+1}\vec{h}_{k+1}^T}. \tag{40}$$

Since both $V_kH_k^T$ and $H_kH_k^T$ can be calculated in an incremental manner as $V_{k+1}H_{k+1}^T = V_kH_k^T + \vec{v}_{k+1}\vec{h}_{k+1}^T$ and $H_{k+1}H_{k+1}^T = H_kH_k^T + \vec{h}_{k+1}\vec{h}_{k+1}^T$, respectively, the space complexity of (40) stays constant. The time complexity of (40) is $O(mr + mr^2) \times K$, where K is the iteration number. Although INMF works well in practice, the multiplicative update rules (40) suffer from slow convergence, which makes (40) time consuming. In addition, it may fail if some elements in the denominators become zero.

D. Incremental Nonnegative Matrix Factorization With Volume Constraint (INMF-VC)

Zhou *et al.* [36] proposed an INMF-VC to control the uniqueness

$$\min_{W\in\mathbf{C}, H\geq 0} D_{t+1} \triangleq \frac{1}{2}\|V - WH\|_F^2 + \mu \ln|\det W|. \tag{41}$$

Minimizing the determinant of W steers (41) toward the unique solution, and the tradeoff parameter μ balances the approximation error of NMF and the volume constraint. In order to steer (41) toward the global solution, INMF-VC sets $\mu = \delta\exp^{-\tau t}$, where δ and τ are both positive constants and μ decreases to zero with increasing sample number t. To learn W and H in an incremental manner, Zhou *et al.* proposed to use the amnesic average method by rewriting (41) into the following two parts given a new sample $\vec{v}_{t+1}$:

$$D_{t+1} \approx \alpha D_t + \beta d_{t+1} \tag{42}$$

where $d_{t+1} = (1/2)\|\vec{v}_{t+1} - W_{t+1}\vec{h}_{t+1}\|_2^2 + \mu\ln|\det W_{t+1}| - \mu\ln|\det W_t|$, and α and β are the moving smooth parameters which are set to $\alpha = 1 - L/t$ and $\beta = L/t$, where $L \in \{1, 2, 3, 4\}$ is the amnesic average parameter. As t tends to infinity, α and β get close to 1 and 0, respectively.

Since (42) is nonconvex, it first updates the coordinate $\vec{h}_{t+1}$ of $\vec{v}_{t+1}$ with the basis W_t fixed, and then updates the basis W_{t+1}. Both $\vec{h}_{t+1}$ and W_{t+1} are updated by using the multiplicative update rule. INMF-VC performs well in blind-source separation and requires W to be square.

E. Online Matrix Factorization Algorithm (OMF)

Recently, Mairal *et al.* [22] proposed an OMF which includes NMF as a special case. Unlike ONMF and INMF, OMF learns the basis matrix W to adapt it to specific data by minimizing an expected cost

$$\min_W f(W) = E_{\vec{v}}(l(\vec{v}, W)) \tag{43}$$

where $l(\vec{v}, W) = \min_{\vec{h}\in\mathbb{R}_+^r}(1/2)\|\vec{v} - W\vec{h}\|_2^2$. OMF efficiently solves (43) by minimizing a quadratic local surrogate $f_t(W) = (1/t)\sum_{i=1}^t l(\vec{v}_i, W)$. In particular, on arrival of the tth sample, OMF updates the basis matrix by

$$W_t = \arg\min_{W\in\mathbf{C}} \frac{1}{t}\left(\frac{1}{2}tr(W^TWA_t) - tr(W^TB_t)\right) \tag{44}$$

where $A_t = \sum_{j=1}^t \vec{h}_j\vec{h}_j^T$ and $B_t = \sum_{j=1}^t \vec{v}_j\vec{h}_j^T$ and they are incrementally updated by $A_t = A_{t-1} + \vec{h}_t\vec{h}_t^T$ and $B_t = B_{t-1} + \vec{v}_t\vec{h}_t^T$, respectively. OMF solves (44) by iteratively updating the columns of W with the following rule until convergence:

$$\vec{w}_j \leftarrow \Pi_{\mathbf{C}}\left(\vec{w}_j - \frac{1}{[A_t]_{jj}}(W\vec{a}_j - \vec{b}_j)\right) \tag{45}$$

where $\vec{a}_j$, $\vec{b}_j$, and $\vec{w}_j$ are the jth column of A_t, B_t, and W, respectively. Note that the learning rate of (45) is the diagonal of the approximated Hessian inverse.

Similar to INMF [35], OMF only stores matrices A_t and B_t. Thus OMF consumes constant memory and scales up gracefully to large-scale datasets. However, (45) converges slowly because the learning rates $[A_t]_{jj}, 1 \leq j \leq r$ ignore most off-diagonal information.

F. OMF-Diagonal Approximation (DA)

Wang *et al.* [37] proposed to use the second-order projected gradient descent method to optimize (44), in which the basis matrix W_t is updated by iterating the following rule until convergence: $W_t^{k+1} = \Pi_C(W_t^k - \mathcal{H}_t^{-1}(W_t^k)\nabla_t(W_t^k))$, wherein $k \geq 1$ is the iteration counter and W_t^k is the kth search point

TABLE II
SUMMARY OF THE USED DATASETS

Datasets	Efficiency comparison				Face recognition			Image annotation			
	m	n	r	sp	#TR	#TS	r	#TR	#TS	#VC	#KD
CBCL	361	500	10/50	.131	30/50/70	470/450/430	10–80	—	—	—	—
ORL	1024	400	10/50	.042	120/200/280	280/200/120	10–150	—	—	—	—
IAPR TC12	100	500	50/80	.745	—	—	—	17 825	1980	291	4.7

m: sample dimensionality. n: sample number. r: reduced dimensionality. sp: sparseness. TR: training set. TS: test set. VC: vocabulary; KD: keywords.

initialized to W_{t-1}, and $\nabla_t(W_t^k)$ and $\mathcal{H}_t(W_t^k)$ are the gradient and the Hessian of (44) at W_t^k, respectively.

Since the exact calculation of the inverse of Hessian matrix is time consuming especially when r increased, Wang *et al.* presented two strategies to approximate the Hessian inverse: 1) DA, which uses only the diagonal of the Hessian matrix to approximate the whole one and 2) conjugate gradient (CG), which approximates the last term, i.e., $\mathcal{H}_t^{-1}(W_t^k)\nabla_t(W_t^k)$, with the least-squares solutions of $\mathcal{H}_t(W_t^k)Q = \nabla_t(W_t^k)$ obtained by using the conjugate gradient method.

DA ignores all of the off-diagonal information in the Hessian matrix, and thus suffers from numerical instability especially when the dataset is sparse. Although CG approximates the Hessian inverse, the conjugate gradient procedure is time consuming. Thus we focused on the DA strategy in this paper and call this method OMF-DA.

The proposed OR-NMF overcomes the aforementioned drawbacks of ONMF, ONMF-IS, INMF, INMF-VC, OMF, and OMF-DA. The experimental results on real-world datasets show that OR-NMF outperforms the existing ONMF algorithms in terms of efficiency, and the experimental results of various applications confirm the effectiveness of OR-NMF.

V. EXPERIMENTS

This section evaluates the efficiency and effectiveness of the OR-NMF by comparing it with representative ONMF algorithms including ONMF [34], ONMF-IS [21], INMF [35], OMF [22], and OMF-DA [37]. The efficiency is evaluated on two face-image datasets, i.e., CBCL [38] and ORL [39], and one popular image dataset, i.e., IAPR TC12 [40]. The effectiveness is evaluated by two applications: 1) face recognition on both CBCL and ORL datasets and 2) image annotation on the IAPR TC12 dataset. We summarize the datasets used in this experiment in Table II and use the following settings.

1) Projection Operator: Based on the definition of the feasible set $\mathbf{C}$ [see (8)], the projection operator $\Pi_{\mathbf{C}}(\cdot)$ plays an important role in Algorithm 1. In all our experiments, we apply the method in [41] to project the columns of W onto the simplex in O(mr) time.

2) Nonnegative Least Squares: To obtain the coefficient of the new sample (see Statement 3 in Algorithm 1), we use the *lsnonneg* method (MATLAB build-in implementation) to solve the nonnegative least-squares problem (4).

3) Algorithmic Schemes: In our experiments, we utilize the well-known multiplicative update rule [8] to obtain the coefficient in INMF. In addition, we set the weights in (39) to $S_{\text{old}} = S_{\text{new}} = 1$ for fair comparison. Note that ONMF receives one chunk of samples at each step, here, we set the chunk size to the reduced dimensionality r.

4) Reduced Dimensionalities: The reduced dimensionality is set to 10, 50, and 80 in the efficiency comparison experiments and varies from 10 to 150 in steps of 10 for evaluating face recognition performance. In addition, the reduced dimensionality for the image annotation experiments varies in a range of $\{10\%, \ldots, 90\%\} \times \min\{m, 200\}$, wherein m is the dimensionality of the visual image features.

A. Efficiency Comparison

To evaluate the efficiency of OR-NMF, we compare its objective values with those of other ONMF algorithms. Since it is difficult to directly calculate the expected cost (3), similar to [22] we instead utilize the empirical cost $f_t(W^t) = (1/t)\sum_{i=1}^{t}(1/2)\|\vec{v}_i - W^t\vec{h}_i\|_2^2$. In the remainder of this section, we compare OR-NMF and MOR-NMF with the existing ONMF algorithms in terms of efficiency on two dense datasets, i.e., CBCL [38] and ORL [39]. For evaluating the proposed algorithms on sparse datasets, we compare them with the existing ONMF algorithms on the sparse dataset, i.e., IAPR TC12 [40]. We compared the efficiency of WOR-NMF with that of ONMF. In this part, all algorithms were applied to two epochs of the permuted training set. We carried out this trial 10 times with different initial points because the objective function (3) is nonconvex. In each trial, all algorithms start from an identical initial point for fair comparison.

1) Dense Datasets: The CBCL dataset [38] contains 2429 face images collected from 10 subjects. Fig. 1(a) (first row) shows some examples of the CBCL dataset. By reshaping a face image to a vector, we get 2429 training samples in $\mathbb{R}^{361}$ and randomly selected 500 samples for online training. Fig. 2 gives the mean and deviation of the objective values versus the sample numbers and CPU seconds of the OR-NMF, MOR-NMF, OMF, OMF-DA, and INMF algorithms. It shows that both OR-NMF and MOR-NMF reduce the objective function faster than other algorithms.

The Cambridge ORL dataset [39] is composed of 400 images collected from 40 individuals. Fig. 1(a) (second row) depicts some examples of the ORL dataset. The training set contains 400 samples in $\mathbb{R}^{1,024}$. Fig. 3 shows that both OR-NMF and MOR-NMF outperform OMF, OMF-DA, and INMF in terms of both objective values and CPU seconds.

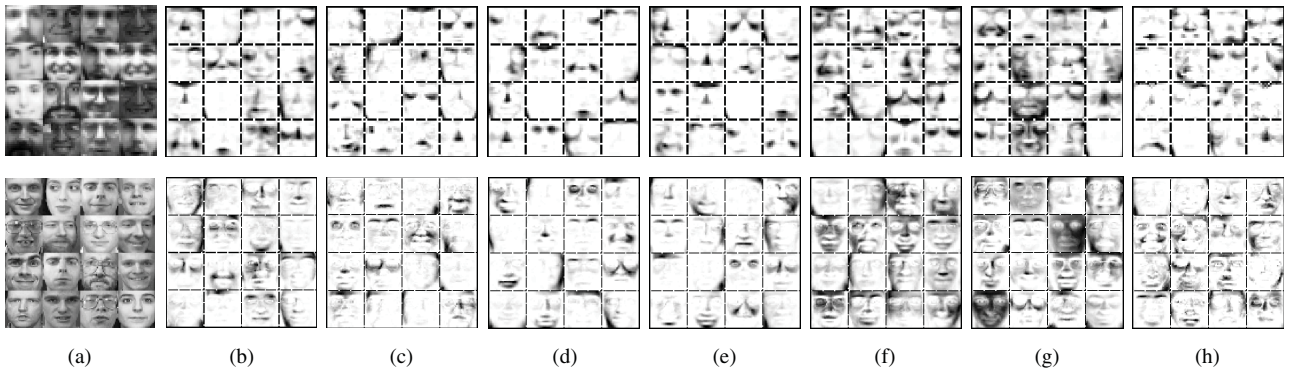

Fig. 1. (a) Image examples, (b) learned bases by OR-NMF, (c) MOR-NMF, (d) OMF, (e) OMF-DA, (f) INMF, (g) WOR-NMF, and (h) ONMF of CBCL (first row) and ORL (second row) dataset.

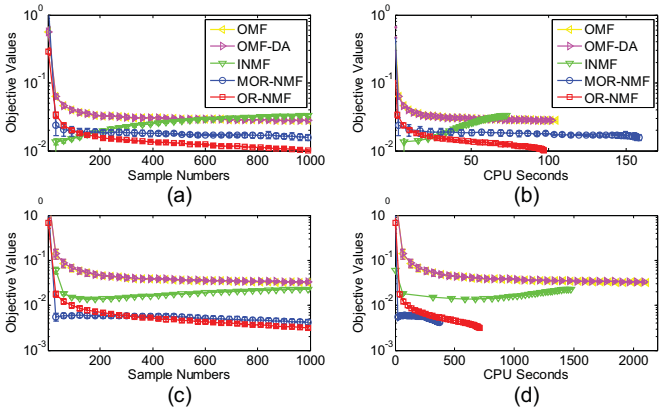

Fig. 2. Objective values versus iteration numbers and CPU seconds of OR-NMF, MOR-NMF, OMF, OMF-DA, and INMF on the CBCL dataset with reduced dimensionality. (a) and (b) $r = 10$. (c) and (d) $r = 50$.

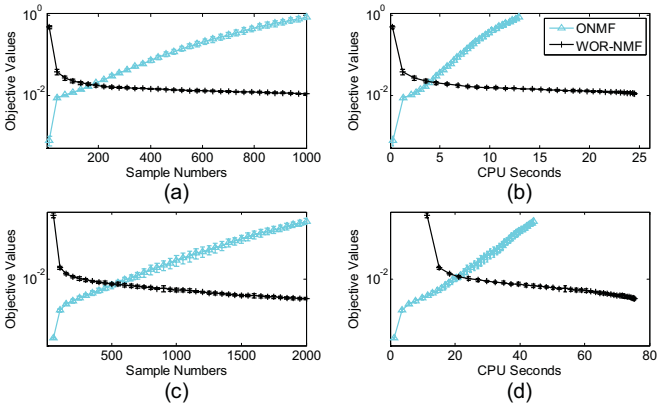

Fig. 4. Objective values versus iteration numbers and CPU seconds of WOR-NMF and ONMF on the CBCL dataset with reduced dimensionality. (a) and (b) $r = 10$. (c) and (d) $r = 50$.

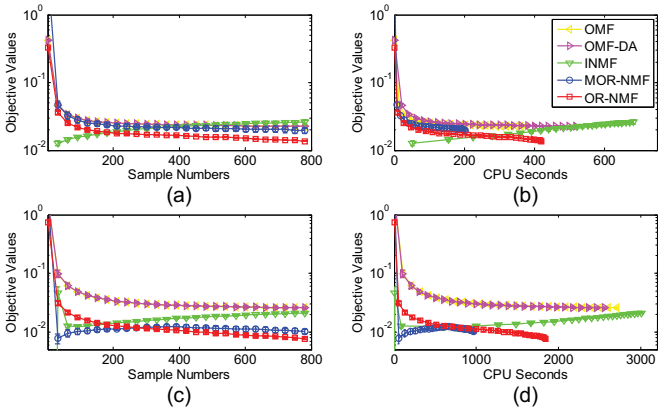

Fig. 3. Objective values versus iteration numbers and CPU seconds of OR-NMF, MOR-NMF, OMF, OMF-DA, and INMF on the ORL dataset with reduced dimensionality. (a) and (b) $r = 10$. (c) and (d) $r = 50$.

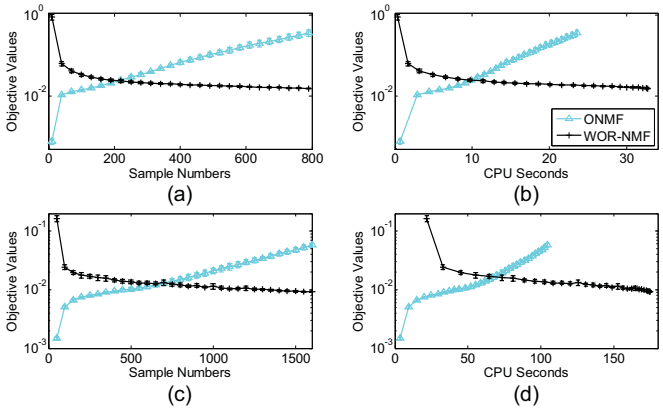

Fig. 5. Objective values versus iteration numbers and CPU seconds of WOR-NMF and ONMF on the ORL dataset with reduced dimensionality. (a) and (b) $r = 10$. (c) and (d) $r = 50$.

Figs. 4 and 5 present the mean and deviation of the objective values versus the sample numbers and CPU seconds of both WOR-NMF and ONMF on both CBCL and ORL datasets. It shows that WOR-NMF outperforms ONMF in terms of both objective values and CPU seconds on these dense datasets.

2) Sparse Datasets: The IAPR TC12 [40] dataset has become popular for image annotation. There are 20 000 images in the IAPR TC12 dataset. In this experiment, the sparse dataset was built by arranging the 100-D "DenseHue" features of 500 randomly selected images into columns. Thus this sparse dataset contains 500 samples in $\mathbb{R}^{100}$. Based on the sparseness defined in [42], we compare the averaged sparseness of the used datasets in Table II. It shows that the IAPR TC12 dataset is much sparser than both the CBCL and ORL datasets.

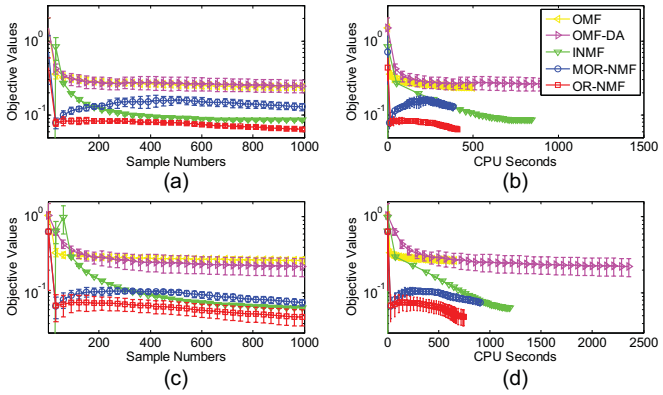

Fig. 6. Objective values versus iteration numbers and CPU seconds of OR-NMF, MOR-NMF, OMF, OMF-DA, and INMF on the IAPR TC12 dataset with reduced dimensionality. (a) and (b) $r = 50$. (c) and (d) $r = 80$.

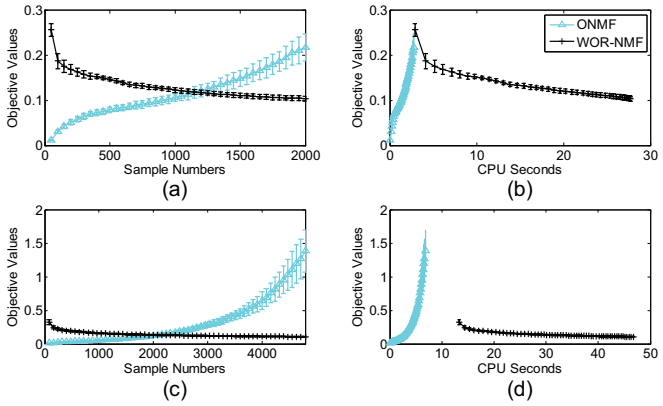

Fig. 7. Objective values versus iteration numbers and CPU seconds of WOR-NMF and ONMF on the IAPR TC12 dataset with reduced dimensionality. (a) and (b) $r = 50$. (c) and (d) $r = 80$.

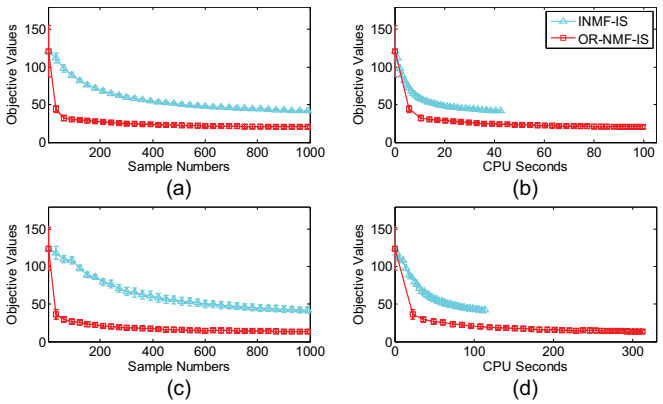

Fig. 8. Objective values versus iteration numbers and CPU seconds of OR-NMF-IS and ONMF-IS on the CBCL dataset with reduced dimensionality. (a) and (b) $r = 10$. (c) and (d) $r = 50$.

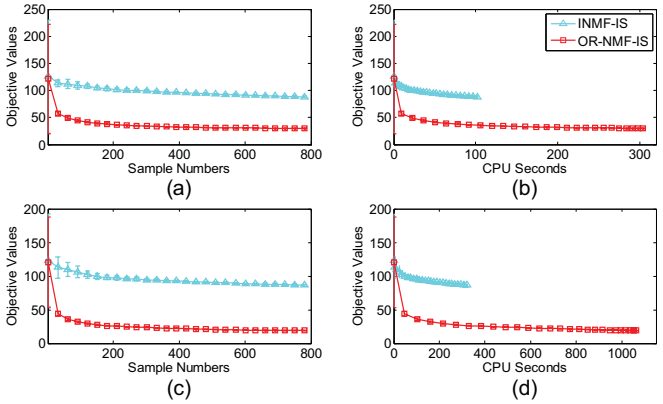

Fig. 9. Objective values versus iteration numbers and CPU seconds of OR-NMF-IS and ONMF-IS on the ORL dataset with reduced dimensionality. (a) and (b) $r = 10$. (c) and (d) $r = 50$.

Fig. 6 gives the mean and deviation of the objective values of OR-NMF, MOR-NMF, OMF, OMF-DA, and INMF. It shows that OR-NMF outperforms OMF, OMF-DA, and INMF in terms of both objective values and CPU seconds. In addition, Fig. 6 shows that MOR-NMF may get higher objective values when the reduced dimensionality is low. That is because MOR-NMF discards old samples in the buffer and thus cannot incorporate sufficient information from such few and sparse samples to update the basis. However, the buffer strategy used in MOR-NMF largely saves storages, and thus it is much more suitable for streaming datasets.

Fig. 7 presents the mean and deviation of the objective values of both MOR-NMF and ONMF. It shows that WOR-NMF outperforms ONMF in terms of objective values. Although ONMF sometimes costs less CPU seconds especially when the reduced dimensionality is relatively high [Fig. 7(b) and (d)], it suffers from a serious nonconvergence problem. Fig. 7 shows that WOR-NMF works well on this sparse dataset.

3) OR-NMF-IS Versus ONMF-IS: For evaluating the capacity of OR-NMF algorithms to optimize IS-NMF [33], we compared OR-NMF-IS with ONMF-IS on the dense datasets. In this experiment, we did not apply these algorithms to the sparse dataset because the IS divergence is ill-posed on that dataset. Figs. 8 and 9 give the mean and deviation of the objective values of both algorithms on the CBCL and ORL datasets. It shows that OR-NMF-IS reduces the objective function more rapidly than ONMF-IS and it confirms that OR-NMF can be applied to incrementally optimizing IS-NMF.

B. Face Recognition

By comparing the bases learned by OR-NMF with those learned by other ONMF algorithms in Fig. 1, it can be concluded that both OR-NMF and MOR-NMF work well in learning parts-based representation on the CBCL [38] and ORL [39] datasets. To further evaluate the effectiveness of such parts-based representation, we compare its face recognition accuracy with those of the existing ONMF algorithms. Different numbers $(3, 5, 7)$ of images were randomly selected from each individual to constitute the training set, and the rest of the images to form the test set. The training set was used to learn basis for the low-dimensional space. The test set was used to report the accuracy in the learned low-dimensional space. The accuracy was calculated as the percentage of samples in the test set that were correctly classified using the NN rule. These trails were independently conducted 10 times and both mean and deviation of accuracy are reported.

1) CBCL Dataset: The CBCL [38] dataset contains 2429 face images taken from 10 subjects. On average, 243 images

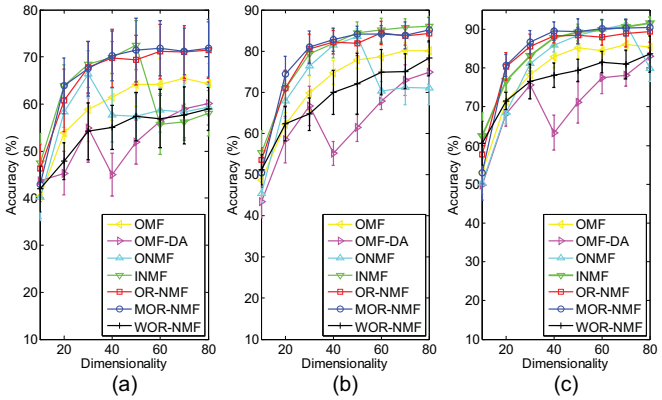

Fig. 10. Mean and deviation of accuracy versus reduced dimensionality of OR-NMF, MOR-NMF, WOR-NMF, OMF, OMF-DA, ONMF, and INMF on the CBCL dataset whereby the training set was composed of (a) three, (b) five, and (c) seven images selected from each individual.

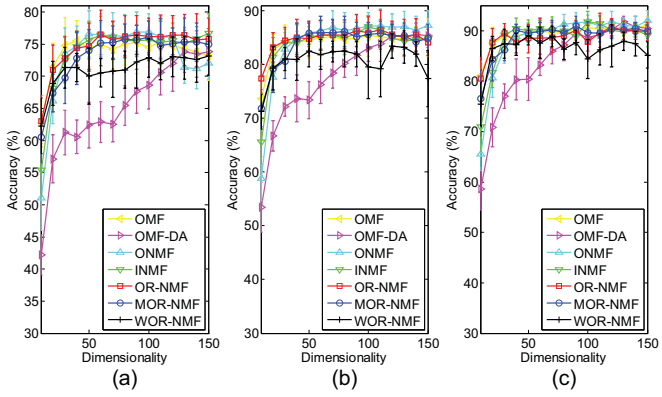

Fig. 11. Mean and deviation of accuracy versus reduced dimensionality of OR-NMF, MOR-NMF, WOR-NMF, OMF, OMF-DA, ONMF, and INMF on the ORL dataset whereby the training set is composed of (a) three, (b) five, and (c) seven images selected from each individual.

were taken for each subject with a variety of illuminations, poses (up to about 30 degrees of rotation in depth), and backgrounds. All face images were aligned according to the eye position and normalized to a 19×19 pixel array. Since the maximum size of the training set in this experiment was 70, we varied the reduced dimensionality from 10 to 80, whereby the step is 10. Fig. 10 depicts the mean and deviation of face recognition accuracies of OR-NMF, MOR-NMF, WOR-NMF, OMF, OMF-DA, ONMF, and INMF on different test sets.

Fig. 10 shows that both OR-NMF and MOR-NMF outperform the other algorithms. Although ONMF and INMF perform well on some reduced dimensionalities, their accuracies decrease when the reduced dimensionality becomes high. OR-NMF, MOR-NMF, and WOR-NMF overcome this problem and perform robustly on all the reduced dimensionalities.

2) ORL Dataset: The Cambridge ORL dataset [39] is composed of 400 images collected from 40 individuals. There are 10 images for each individual with various lighting, facial expressions, and facial details (with glasses or without glasses). All images were taken against the same dark background, and each image was normalized to a 32×32 pixel array and reshaped to a long vector. We varied the reduced dimensionality from 10 to 150, whereby the step is 10. Fig. 11 shows the face recognition accuracies of OR-NMF, MOR-NMF, WOR-NMF, OMF, OMF-DA, ONMF, and INMF on different test sets and it gives the same observation as Fig. 10.

C. Image Annotation

To annotate a given image, the joint equal contribution model (JEC) [43] transfers keywords from the training images as an annotation in a greedy manner. In particular, JEC sorts the training images according to a joint distance calculated by a linear combination of the distances based on different features and denotes these sorted training images as $I_1, \ldots, I_n$. By sorting the keywords of the nearest neighbor I_1 according to their frequency in the training set, it picks up the top k keywords to annotate the given image. If $|I_1| < k$, wherein $|I_1|$ denotes the keyword number in I_1, and JEC sorts the keywords in the remaining neighbors $I_2, \ldots, I_n$ and picks up $k - |I_1|$ top keywords to complete the annotation. Refer to [43] for detailed discussion. In this experiment, we fixed the number of obtained keywords k to 5.

Unlike [43], we project the visual features in both the training set and the test set onto the semantic space learned by OR-NMF, MOR-NMF, WOR-NMF, ONMF, INMF, OMF, and OMF-DA by using $\vec{h}_i = W^{\dagger}\vec{v}_i$, wherein $\vec{v}_i$ denotes the visual feature and replaces the distance between $\vec{v}_i$ and $\vec{v}_j$ with that between $\vec{h}_i$ and $\vec{h}_j$. In this experiment, two types of visual features, i.e., the 100-D "DenseHue" and the 100-D "HarrisHue" were used for the tested image datasets. Note that other visual features, e.g., color [44], luminance [45], and synthetic 3-D features [46], can be used to enhance the performance. Here we use only two of them because this experiment focuses on the effectiveness of OR-NMF compared with those of the competitive ONMF algorithms. Similar to [43], we utilize the greedy label transfer approach to assign keywords to the test image. Three metrics are used to evaluate the performance of the image annotation, namely accuracy (AC), recall (RC), and normalized score (NS) [47] defined as $AC = r/(r + w)$, $RC = r/n$, and $NS = r/n - w/(N - n)$, wherein r and w denote the number of correctly and wrongly annotated keywords, respectively, and n and N the number of keywords in the test image and vocabulary, respectively.

1) IAPR TC12 Dataset: The IAPR TC12 dataset [40] contains 20000 images of natural scenes including different sports and actions, photographs of people, animals, cities, landscapes and many other aspects of life. By extracting keywords with the Tree Tagger part-of-speech tagger, a vocabulary including 291 keywords and an average of 4.7 keywords per image is obtained [43]. The training set contains 17825 images and the test set is formed by the remaining images.

Fig. 12 compares the proposed OR-NMF, MOR-NMF, and WOR-NMF with ONMF, INMF, OMF, and OMF-DA on the IAPR TC12 dataset. It shows that both OR-NMF and WOR-NMF outperform the representative ONMF algorithms, especially when the reduced dimensionality is high.

Table III gives the predicted and manually annotated keywords for example images from the IAPR TC12 dataset. It shows that all the keywords in these images were predicted.

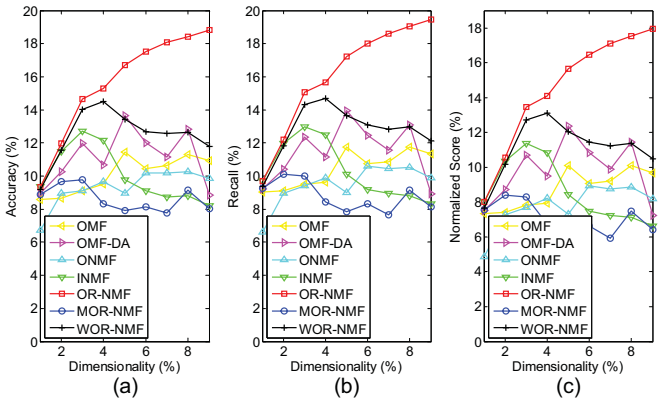

Fig. 12. (a) Accuracy, (b) recall, and (c) normalized score versus reduced dimensionalities of OR-NMF, MOR-NMF, WOR-NMF, ONMF, INMF, OMF and OMF-DA on the IAPR TC12 dataset.

TABLE III
PREDICTED KEYWORDS BY USING OR-NMF VERSUS MANUALLY ANNOTATED KEYWORDS FOR IMAGES IN THE IAPR TC12 DATABASE

					
Predicted Keywords	base, horse, man, statue, building	adult, court, man, player, tennis	forest, sky, snow, tree, railing	sun, range, horizon, landscape, mountain	adult, cloud, grey, sea, sky
Manually Annotated Keywords	base, horse, statue, building	court, man, player, tennis	forest, sky, snow, tree	range, sun, landscape, mountain	cloud, grey, sea, sky

It is interesting that some of the additionally annotated keywords have also semantic representation, e.g., "man," "adult," and "horizon" predicted for the first, second, and fourth column of images, respectively, are meaningful.

Fig. 12 shows that MOR-NMF is not as good as OR-NMF in terms of AC, RC, and NS because it discards old samples in the buffer and thus cannot incorporate sufficient information into the semantic space. We refer to Fig. 6 to compare their objective values on the IAPR TC12 dataset. However, the buffer strategy used in MOR-NMF largely saves storages, and thus it is much more suitable for streaming datasets.

VI. CONCLUSION

This paper proposed an efficient online algorithm termed OR-NMF to learn NMF from large-scale or streaming datasets. In particular, we treated NMF as a stochastic optimization problem and utilized the robust stochastic approximation method (RSA) to update the basis matrix in an online fashion. By smartly choosing the learning rates and using the averaging technique, OR-NMF guarantees convergence at the rate of $O(1/\sqrt{k})$ in updating the basis matrix at each step. To keep the space complexity of OR-NMF constant, we introduced a buffering strategy in optimization. We demonstrated that OR-NMF could be naturally extended to handle l_1-regularized, l_2-regularized NMF, NMF with box constraint, and Itakuro–Saito divergence-based NMF. Preliminary experimental results on real-world datasets showed that OR-NMF could outperform the existing ONMF algorithms in terms of efficiency. The experimental results of face recognition and image annotation on public datasets confirmed that the performance of OR-NMF is superior to other ONMF algorithms.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, D. Liu, the handling associate editor, and anonymous reviewers for their support and constructive comments on this paper.

REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 21, pp. 788–791, 1999.

[2] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 683–695, May 2006.

[3] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 734–749, May 2010.

[4] J. Wang, S. Yan, Y. Fu, X. Li, and T. S. Huang, "Non-negative graph embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 4. Anchorage, AK, Jun. 2008, pp. 1–8.

[5] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[6] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Manifold regularized discriminative non-negative matrix factorization with fast gradient descent," *IEEE Trans. Imag. Process.*, vol. 20, no. 7, pp. 2030–2048, Jul. 2011.

[7] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Non-negative patch alignment framework," *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1218–1230, Aug. 2011.

[8] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, vol. 12. Cambridge, MA: MIT Press, 2000, pp. 556–562.

[9] C. J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1589–1596, Nov. 2007.

[10] C. J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.

[11] M. W. Berry, M. Browne, A. Langville, P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate non-negative matrix factorization," *Comput. Stat. Data Anal.*, vol. 52, no. 1, pp. 155–173, 2007.

[12] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *Proc. IEEE Int. Conf. Data Min.*, Dec. 2008, pp. 353–362.

[13] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for non-negative matrix factorization," *IEEE Trans. Signal Process.*, 2012, DOI: 10.1109/TSP.2012.2190406, to be published.

[14] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, "Text mining using non-negative matrix factorizations," in *Proc. 4th SIAM Int. Conf. Data Min.*, 2004, pp. 452–456.

[15] X. Wang, Z. Li, and D. Tao, "Subspaces indexing model on grassmann manifold for image search," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2627–2635, Sep. 2011.

[16] B. Xie, Y. Mu, D. Tao, and K. Huang, "m-SNE: Multiview stochastic neighbor embedding," *IEEE Trans. Syst. Man Cybern. B*, vol. 41, no. 4, pp. 1088–1096, Aug. 2011.

[17] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, 2009.

[18] L. Bottou, O. Bousquet, and G. Zürich, "The trade-offs of large scale learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20. 2008, pp. 161–168.

[19] V. Strassen, "The existence of probability measures with given marginals," *Ann. Math. Stat.*, vol. 36, no. 2, pp. 423–439, 1965.

[20] A. Korattikara, L. Boyles, M. Welling, J. Kim, and H. Park, "Statistical optimization of non-negative matrix factorization," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, vol. 15. 2011, pp. 1–9.

[21] B. A. Lefèvre, F. Bach, and C. Fèvotte, "Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, New Paltz, NY, Oct. 2011, pp. 313–316.

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[23] A. W. Van der Vaart, *Asymptotic Statistics*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[24] D. L. Fisk, "Quasi-martingales," *Trans. Amer. Math. Soc.*, vol. 120, no. 3, pp. 359–388, 1965.

[25] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Nashua, NH: Athena Scientific, 1999.

[26] P. O. Hoyer, "Non-negative sparse coding," in *Proc. 12th IEEE Workshop Neural Netw. Signal Process.*, Nov. 2002, pp. 557–565.

[27] X. He, D. Cai, Y. Shao, H. Bao, and J. Han, "Laplacian regularized Gaussian mixture model for data clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1406–1418, Sep. 2011.

[28] B. Geng, D. Tao, C. Xu, L. Yang, and X. Hua, "Ensemble manifold regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1227–1233, Jun. 2012.

[29] S. Si, D. Tao, and B. Geng, "Bregman divergence based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.

[30] M. Song, D. Tao, C. Chen, X. Li, and C. Chen, "Color to gray: Visual cue preservation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1537–1552, Sep. 2010.

[31] X. Tian, D. Tao, and Y. Rui, "Sparse transfer learning for interactive video search reranking," *ACM Trans. Multimedia Comput. Commun. Appl.*, 2012, arXiv:1103.2756v3, to be published.

[32] X. He, "Laplacian regularized D-optimal design for active learning and its application to image retrieval," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 254–263, Jan. 2010.

[33] C. Fèvotte, N. Bertin, and J. L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Comput.*, vol. 21, no. 3, pp. 793–830, Mar. 2009.

[34] B. Cao, D. Shen, S. J. Tao, X. Wang, Q. Yang, and Z. Chen, "Detect and track latent factors with online nonnegative matrix factorization," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, San Francisco, CA, 2007, pp. 2689–2694.

[35] S. S. Bucak and B. Gunsel, "Incremental subspace learning via non-negative matrix factorization," *Pattern Recognit.*, vol. 42, no. 5, pp. 788–797, 2009.

[36] G. Zhou, Z. Yang, and S. Xie, "Online blind source separation using incremental nonnegative matrix factorization with volume constraint," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 550–560, Apr. 2011.

[37] F. Wang, C. Tan, A. C. Knoig, and P. Li, "Efficient document clustering via online nonnegative matrix factorization," in *Proc. SIAM Int. Conf. Data Min.*, 2011, pp. 1–12.

[38] B. Weyrauch, J. Huang, B. Heisele, and V. Blanz, "Component-based face recognition with 3-D morphable models," in *Proc. IEEE Workshop Face Process. Video*, Jun. 2004, p. 85.

[39] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Sarasota, FL, Dec. 1994, pp. 138–142.

[40] M. Grubinger, P. D. Clough, M. Henning, and D. Thomas, "The IAPR benchmark: A new evaluation resource for visual information systems," in *Proc. Int. Conf. Lang. Resour. Evaluat.*, Genoa, Italy, 2006, pp. 1–10.

[41] J. Duchi, S. Shalev-Schwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l_1-ball for learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 1–8.

[42] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.

[43] A. Makadia, V. Pavlovic, and S. Kumar, "A new baseline for image annotation," in *Proc. 10th Eur. Conf. Comput. Vis.*, vol. 3. Berlin, Germany, 2008, pp. 316–329.

[44] M. Song, D. Tao, C. Chen, X. Li, and C. Chen, "Color to gray: Visual cue preservation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1537–1552, Sep. 2010.

[45] M. Song, D. Tao, C. Chen, J. Luo, and C. Zhang, "Probabilistic exposure fusion," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 341–357, Jan. 2012.

[46] M. Song, D. Tao, X. Huang, C. Chen, and J. Bu, "Three-dimensional face reconstruction from a single image by a coupled RBF network," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2887–2897, May 2012.

[47] K. Barnard, P. Duygulu, N. Freitas, D. Forsyth, D. Blei, and M. Jordan, "Matching words and pictures," *J. Mach. Learn. Res.*, vol. 3, pp. 1107–1135, Mar. 2003.

[48] P. Duygulu, N. Freitas, K. Barnard, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Proc. 7th Eur. Conf. Comput. Vis.*, vol. 4. 2002, pp. 97–112.

Naiyang Guan received the B.S. and M.S. degrees from the National University of Defense Technology (NUDT), Changsha, China. He is currently pursuing the Ph.D. degree with the School of Computer Science, NUDT.

He was a Visiting Student with the School of Computer Engineering, Nanyang Technological University, Singapore, from October 2009 to October 2010. He is currently a Visiting Scholar with the Centre for Quantum Computation and Information Systems and the Faculty of the Engineering and Information Technology, University of Technology, Sydney, Australia. His current research interests include computer vision, image processing, and convex optimization.

Dacheng Tao (M'07–SM'12) is a Professor of computer science with the Centre for Quantum Computation and Intelligent Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia. He mainly applies statistics and mathematics for data analysis problems in data mining, computer vision, machine learning, multimedia, and video surveillance. He has authored or co-authored more than 100 scientific articles, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, TRANSACTIONS ON IMAGE PROCESSING, *Artificial Intelligence and Statistics, International Conference on Data Mining (ICDM), Computer Vision and Pattern Recognition, and European Conference on Computer Vision*.

He received the Best Theory/Algorithm Paper Runner Up Award in IEEE ICDM in 2007.

Zhigang Luo received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 1981, 1993, and 2000, respectively.

He is currently a Professor with the School of Computer Science, NUDT. His current research interests include parallel computing, computer simulation, and bioinformatics.

Bo Yuan received the Bachelors degree from Peking University Medical School, Beijing, China, in 1983, the M.S. degree in biochemistry and the Ph.D. degree in molecular genetics from the University of Louisville, Louisville, KY, in 1990 and 1995, respectively.

He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China. Before joining SJTU in 2006, he was a tenure-track Assistant Professor with Ohio State University (OSU), Columbus, while serving as a Co-Director for the OSU Program in Pharmacogenomics. At OSU, he was the Founding Director for OSU's Genome Initiative from the early 2000s, leading one of the only three independent efforts in the world (besides the Human Genome Project and the Celera company), having assembled and deciphered the entire human and mouse genomes. His current research interests include biological networks, network evolution, stochastic processes, biologically inspired computing, and bioinformatics, particularly the potential impact of these frameworks on the development of intelligent algorithms and systems.