ONLINE NONNEGATIVE MATRIX FACTORIZATION WITH OUTLIERS

Renbo Zhao, Vincent Y. F. Tan

Department of Electrical and Computer Engineering, National University of Singapore

ABSTRACT

We propose an optimization framework for performing online Nonnegative Matrix Factorization (NMF) in the presence of outliers, based on ℓ_1 regularization and stochastic approximation. Due to the online nature of the algorithm, the proposed method has extremely low computational and storage complexity and is thus particularly applicable in this age of BigData. Furthermore, our algorithm shows promising performance in dealing with outliers, which previous online NMF algorithms fail to cope with. Convergence analysis shows the dictionary learned by our algorithm converges to that learned by its batch counterpart almost surely, as data size tends to infinity. We show numerically on a range of face datasets that our algorithm is superior to the state-of-the-art NMF algorithms in terms of running time, basis representations and reconstruction of original images. We also observe that our algorithm performs well even when the density of outliers reaches 40%. We provide explanations behind this seemingly surprising result.

Index Terms— Online Learning, Nonnegative Matrix Factorization, Scalable Methods, Dimensionality Reduction

1. INTRODUCTION

Nonnegative Matrix Factorization (NMF) is a popular dimensionality reduction [1] and data clustering method [2], due to its parts-based, non-subtractive interpretation of the learned basis [3]. Many algorithms have been proposed for NMF, including multiplicative updates [3], block principal pivoting [4], projected gradient descent [5] and alternating nonnegative least squares [6]. However, all these algorithms fall in the class of *batch* NMF algorithms. This class of algorithms has two major limitations. First, in this age of BigData, real-world datasets are high-dimensional and contain a large number of samples. Thus, the computational time and storage space incurred by batch algorithms are prohibitive. In addition, when new data points arrive batch algorithms have to perform computation from scratch whereas it makes more sense to simply update the solutions obtained previously. Second, when data are contaminated by outliers (for example, intense salt noise in images due to acquisition imperfections), reasonable basis vectors cannot be learned in general. Thus the underlying subspace cannot be recovered reliably.

To overcome these two limitations, in recent years, algorithms have been proposed to tackle each limitation separately. For the first limitation (large-scale datasets), many *online* NMF (ONMF) algorithms have been proposed, including [7–12]. These algorithms achieve successes in different applications, such as visual tracking [8, 12] and document clustering [11]. For the second limitation (outliers), (batch) *robust* NMF (RNMF) algorithms [13–17] have been developed to simultaneously remove outliers and learn basis representations from the recovered subspace. Although each aforementioned algorithm, be it online NMF or robust NMF, solves one problem, as we show later, it will suffer from the other problem. Thus, we need to devise a new and unified algorithm that can overcome both limitations simultaneously.

In this paper, we introduce such an algorithm called *online* NMF in the presence of *outliers* (ONMFO). This algorithm aims to remove the outliers while performing online learning, so we are able to learn the parts-based basis representation as if we have uncorrupted data. Convergence analysis shows the dictionary learned by our algorithm converges to that learned by its batch counterpart almost surely, as data size tends to infinity. To the best of our knowledge, thus far, online NMF algorithms specifically designed to handle outliers have not been considered in any previous works. We show, through extensive numerical simulations on three well-known face datasets, that (i) the running time of ONMFO is significantly less than that of RNMF; (ii) the learned basis vectors of ONMFO are comparable to that learned by RNMF, and are intuitively more representative of the parts of human faces than ONMF and PCA-based online algorithms; and (iii) the image reconstruction results of ONMFO are slightly inferior to that of RNMF, but are superior to other algorithms. Moreover, ONMFO performs well even when the outlier density reaches 40%. We provide explanations for this seemingly surprising result.

2. PROBLEM FORMULATION

2.1. Notations

In the following, we use capital boldface letters to denote matrices. In particular, we use $\mathbf{V}$, $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{R}$ to denote data matrix, dictionary matrix, coefficient matrix and outlier matrix respectively, such that $\mathbf{V} \approx \mathbf{W}\mathbf{H} + \mathbf{R}$. $\mathbf{V}$, $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{R}$ are all nonnegative with dimensions $F \times N$, $F \times K$, $K \times N$ and $F \times N$ respectively. Here F, K and N denote the observed dimension, the (known) latent dimension and the number of data samples respectively. We use lower-case boldface letters to denote vectors. Specifically, we use $\mathbf{v}$, $\mathbf{w}$, $\mathbf{h}$ and $\mathbf{r}$ to denote the columns of $\mathbf{V}$, $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{R}$, respectively. Given a nonnegative matrix $\mathbf{X}$, its i-th column is denoted by $\mathbf{x}_i$ and (i,j)-th entry by $\mathbf{X}_{i,j}$. Moreover, we denote its Frobenius norm by $\|\mathbf{X}\|_F$, and $\ell_{1,1}$ norm by $\|\mathbf{X}\|_{1,1} = \sum_{i,j} \mathbf{X}_{i,j}$. Inequality $\mathbf{x} \geq 0$ or $\mathbf{X} \geq 0$ denotes entry-wise nonnegativity. For arithmetic operations, we use $\cdot$ to denote the Hadamard product and $/$ to denote entry-wise quotient (division) between matrices.

2.2. Cost Functions

In this work, we intend to develop a scalable, online method to learn a reasonable basis representation of streaming data contaminated by entry-wise outliers. In other words, we aim to learn the basis representation (matrix) $\mathbf{W}$ by minimizing the effects of outliers. In this work, an outlier is defined to be an entry corrupted by a *gross additive nonnegative component*. Let t be a time index. At time t, our

optimization problem is formulated as follows:

$$\min \quad \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\mathbf{v}_i-\mathbf{W}_t\mathbf{h}_i-\mathbf{r}_i\right\|_F^2+\lambda\left\|\mathbf{r}_i\right\|_1$$
$$\text{subject to} \quad \mathbf{W}_t\geq 0,\mathbf{h}_1,\ldots,\mathbf{h}_t\geq 0,\mathbf{r}_1,\ldots,\mathbf{r}_t\geq 0, \quad (1)$$

where λ denotes the regularization weight. In (1), $\mathbf{W}_t$ denotes the updated dictionary matrix at time t, and $\mathbf{h}_i$ and $\mathbf{r}_i$ denote learned coefficients and outliers at time i . We use ℓ_1 regularization on $\mathbf{r}_i$ to enforce it to be sparse.

Equivalently, (1) can be solved in two steps

$$\mathbf{W}_t=\arg\min_{\mathbf{W}\geq 0} f_t(\mathbf{W}),\ \text{where}\ f_t(\mathbf{W})=\frac{1}{t}\sum_{i=1}^{t}\ell(\mathbf{v}_i,\mathbf{W}), \quad (2)$$

and

$$\ell(\mathbf{v}_i,\mathbf{W})=\min_{\mathbf{h},\mathbf{r}\geq 0}\frac{1}{2}\left\|\mathbf{v}_i-\mathbf{W}\mathbf{h}-\mathbf{r}\right\|_2^2+\lambda\left\|\mathbf{r}\right\|_1. \quad (3)$$

We develop algorithms in Section 3.2 to minimize the empirical cost $f_t(\mathbf{W})$. However, for the purpose of analysis (in Section 3.3), we focus on the expected cost over the distribution $\mathcal{P}$ of i.i.d. data samples, i.e.,

$$\min_{\mathbf{W}} f(\mathbf{W})=\mathbb{E}_{\mathbf{v}\sim\mathcal{P}}[\ell(\mathbf{v},\mathbf{W})]=\lim_{t\to\infty} f_t(\mathbf{W})\ \text{w.p. 1}. \quad (4)$$

We show in Theorem 3 as $t\to\infty$, $\mathbf{W}_t$ learned by our algorithm will converge to the optimal solution of (4) almost surely.

3. OPTIMIZATION ALGORITHMS

In this section we first derive a batch algorithm for NMF with $\ell_{1,1}$ regularization. Based on it, we derive a corresponding online algorithm using ideas from Section 2.2. We also show the proposed online algorithm has nice convergence properties, i.e., the dictionary matrix learned by it converges asymptotically to the optimal solution of (4) almost surely, as data size tends to infinity.

3.1. Batch Optimization Algorithms

In this section we derive the optimization algorithms for minimizing the following problem

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0,\mathbf{R}\geq 0}\frac{1}{2}\left\|\mathbf{V}-\mathbf{W}\mathbf{H}-\mathbf{R}\right\|_F^2+\lambda\left\|\mathbf{R}\right\|_{1,1}. \quad (5)$$

In the literature of batch robust NMF, only [17] considered a similar formulation as (5). However, in this work we impose nonnegativity constraint on $\mathbf{R}$ and derive a new unified multiplicative update algorithm for $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{R}$. This algorithm is different from the method used in [17]. Our strategy is to first hypothesize a multiplicative update algorithm using a heuristic approach as in [3]. Then we prove the cost function in (5) is nonincreasing under the updates.

First, we propose the following multiplicative updates using the heuristic approach in [3] and *block coordinate descent* framework:

$$\mathbf{W}=\tilde{\mathbf{W}}\cdot\frac{\mathbf{V}\mathbf{H}^\top}{(\tilde{\mathbf{W}}\mathbf{H}+\mathbf{R})\mathbf{H}^\top}, \quad (6)$$

$$\mathbf{H}=\tilde{\mathbf{H}}\cdot\frac{\mathbf{W}^\top\mathbf{V}}{\mathbf{W}^\top(\mathbf{W}\tilde{\mathbf{H}}+\mathbf{R})}, \quad (7)$$

$$\mathbf{R}=\tilde{\mathbf{R}}\cdot\frac{\mathbf{V}}{\mathbf{W}\mathbf{H}+\tilde{\mathbf{R}}+\boldsymbol{\lambda}_{F\times N}}, \quad (8)$$

Algorithm 1 Online NMF with outliers (ONMFO)

Input: Data matrix $\mathbf{V}$, regularization weight λ, initial dictionary matrix $\mathbf{W}_0$

for t = 1 to N **do**

1) Observe sample $\mathbf{v}_t$.

2) Learn coefficient vector $\mathbf{h}_t$ and outlier vector $\mathbf{r}_t$ based on $\mathbf{W}_{t-1}$ iteratively, using (16) and (17)

$$\{\mathbf{h}_t,\mathbf{r}_t\}=\arg\min_{\mathbf{h}\geq 0,\mathbf{r}\geq 0}\frac{1}{2}\left\|\mathbf{v}_t-\mathbf{W}_{t-1}\mathbf{h}-\mathbf{r}\right\|_F^2+\lambda\left\|\mathbf{r}\right\|_1. \quad (11)$$

3) Update the sufficient statistics

$$\mathbf{A}_t:=\mathbf{A}_{t-1}+\mathbf{v}_t\mathbf{h}_t^\top, \quad (12)$$
$$\mathbf{B}_t:=\mathbf{B}_{t-1}+\mathbf{h}_t\mathbf{h}_t^\top, \quad (13)$$
$$\mathbf{C}_t:=\mathbf{C}_{t-1}+\mathbf{r}_t\mathbf{h}_t^\top. \quad (14)$$

4) Update dictionary matrix $\mathbf{W}_t$ iteratively using (18)

$$\mathbf{W}_t=\arg\min_{\mathbf{W}\geq 0}\frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|\mathbf{v}_i-\mathbf{W}\mathbf{h}_i-\mathbf{r}_i\right\|_F^2+\lambda\left\|\mathbf{r}_i\right\|_1 \quad (15)$$

End for

Output: Dictionary matrix $\mathbf{W}_N$

where $\boldsymbol{\lambda}_{F\times N}$ is an $F\times N$ matrix with each entry equal to λ. Here $\tilde{\mathbf{W}}$ denotes the previous value of $\mathbf{W}$ in a sequence of iterations. $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{R}}$ have similar meanings.

Next, we show the cost function in (5) is non-increasing under the updates in (6) to (8), based on the *majorization-minimization* framework (see [18] for details). Under such framework, it suffices to find auxiliary (upperbound) functions for $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{R}$ and verify the updates minimize the auxiliary functions. We do so in the next two lemmas, whose proofs are deferred to the extended version.

Lemma 1 *Let* $F(\mathbf{h})=\frac{1}{2}\left\|\mathbf{v}-\mathbf{W}\mathbf{h}-\mathbf{r}\right\|_2^2+\lambda\left\|\mathbf{r}\right\|_1$, *then*

$$\tilde{F}(\mathbf{h}|\tilde{\mathbf{h}})=\frac{1}{2}\mathbf{h}^\top\mathbf{M}\mathbf{h}-\mathbf{v}^\top\mathbf{W}\mathbf{h}+\frac{1}{2}\left(\mathbf{r}^\top\mathbf{W}\tilde{\mathbf{h}}+\left\|\mathbf{v}'\right\|_2^2+2\lambda\left\|\mathbf{r}\right\|_1\right) \quad (9)$$

is an auxiliary function for $F(\mathbf{h})$, *where* $\mathbf{M}=\mathrm{diag}\left(\frac{\mathbf{W}^\top\mathbf{W}\tilde{\mathbf{h}}+\mathbf{W}^\top\mathbf{r}}{\tilde{\mathbf{h}}}\right)$ *and* $\mathbf{v}'=\mathbf{v}-\mathbf{r}$. *Here* $\mathbf{X}=\mathrm{diag}(\mathbf{x})$ *denotes the diagonal matrix formed from the entries of the vector* $\mathbf{x}$.

Lemma 2 *Let* $G(\mathbf{r})=\frac{1}{2}\left\|\mathbf{v}-\mathbf{W}\mathbf{h}-\mathbf{r}\right\|_2^2+\lambda\left\|\mathbf{r}\right\|_1$, *then*

$$\tilde{G}(\mathbf{r}|\tilde{\mathbf{r}})=\frac{1}{2}\mathbf{r}^\top(\mathbf{K}+\mathbf{I})\mathbf{r}-\mathbf{v}^\top\mathbf{r}+\frac{1}{2}\left(\left\|\mathbf{v}-\mathbf{W}\mathbf{h}\right\|_2^2+(\mathbf{W}\mathbf{h})^\top\tilde{\mathbf{r}}+\lambda\left\|\tilde{\mathbf{r}}\right\|_1\right) \quad (10)$$

is an auxiliary function of $G(\mathbf{r})$, *where* $\mathbf{K}=\mathrm{diag}\left(\frac{\mathbf{W}\mathbf{h}+\boldsymbol{\lambda}_{F\times 1}}{\tilde{\mathbf{r}}}\right)$.

Minimizations on (9) and (10) result in updates (7) and (8). Due to symmetry between $\mathbf{W}$ and $\mathbf{H}$, (6) can be easily obtained.

3.2. Online Optimization Algorithms

Since we only have $\mathbf{v}_t$ at time t, we cannot solve (1) exactly. Therefore in this section we derive an online optimization algorithm that approximately solves (1) based on the *stochastic approximation*

framework (e.g., Mairal *et al.* [19]). This technique approximately optimizes (1) in two steps. First, at time t, after observing $\mathbf{v}_t$, we first iteratively learn $\mathbf{h}_t$ and $\mathbf{r}_t$ based on $\mathbf{W}_{t-1}$ by solving (3). Based on (7) and (8), we have the following update rules *for one iteration*

$$\mathbf{h} = \tilde{\mathbf{h}} \cdot \frac{\mathbf{W}_{t-1}^\top \mathbf{v}}{\mathbf{W}_{t-1}^\top \mathbf{W}_{t-1} \tilde{\mathbf{h}} + \mathbf{W}_{t-1}^\top \mathbf{r}}, \tag{16}$$

$$\mathbf{r} = \tilde{\mathbf{r}} \cdot \frac{\mathbf{v}}{\mathbf{W}_{t-1}\mathbf{h} + \tilde{\mathbf{r}} + \boldsymbol{\lambda}_{F\times 1}}. \tag{17}$$

Next, after obtaining $\mathbf{h}_t$ and $\mathbf{r}_t$, we update $\mathbf{W}_t$ from $\mathbf{W}_{t-1}$ by minimizing the cost with respect to all past samples indexed by $i \leq t$. Based on (6), we have the following update rule for one iteration

$$\mathbf{W} = \tilde{\mathbf{W}} \cdot \frac{\mathbf{A}_t}{\tilde{\mathbf{W}}\mathbf{B}_t + \mathbf{C}_t}, \tag{18}$$

where $\mathbf{A}_t, \mathbf{B}_t$ and $\mathbf{C}_t$ are *sufficient statistics* and defined as

$$\mathbf{A}_t = \sum_{i=1}^{t} \mathbf{v}_i \mathbf{h}_i^\top, \qquad \mathbf{B}_t = \sum_{i=1}^{t} \mathbf{h}_i \mathbf{h}_i^\top, \qquad \mathbf{C}_t = \sum_{i=1}^{t} \mathbf{r}_i \mathbf{h}_i^\top.$$

This means after learning $\mathbf{h}_t$ and $\mathbf{r}_t$, we update these sufficient statistics and may discard $\mathbf{h}_t$ and $\mathbf{r}_t$ henceforth. In this way, we update $\mathbf{W}_t$ in an *online* manner without accessing any of the past data $\{\mathbf{v}_i, \mathbf{h}_i, \mathbf{r}_i\}_{i=1}^t$. This reduces the storage complexity tremendously, since only the sufficient statistics and $\mathbf{W}_t$ need to be stored. A complete algorithm is shown in Algorithm 1.

3.3. Convergence Analysis

Theorem 3 *Assume*

1. *The distribution $\mathcal{P}$ in* (4) *has a closed and bounded support.*
2. *Both the optimal solution of* (4)*, namely $\mathbf{W}^*$, and the initial matrix $\mathbf{W}_0$ in Algorithm 1 have full column rank K.*
3. *The function*

$$\tilde{f}_t(\mathbf{W}) = \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2} \|\mathbf{v}_i - \mathbf{W}\mathbf{h}_i - \mathbf{r}_i\|_F^2 + \lambda \|\mathbf{r}_i\|_1, \tag{19}$$

an upper bound for $f_t(\mathbf{W})$, is strongly convex in $\mathbf{W}$.

Then the sequence $\{\mathbf{W}_t\}$ converges to $\mathbf{W}^*$ entrywise almost surely.

Intuitively, this theorem suggests for large-scale streaming data, our online algorithm can achieve almost the same performance as its batch counterpart. Note that some assumptions are natural: the magnitudes of real data are typically bounded so $\mathcal{P}$ has a compact support; also, the full rank assumption of $\mathbf{W}^*$ is corroborated by the parts-based basis representation property of $\mathbf{W}^*$. We verified assumption 3 holds empirically in our numerical experiments.

Proof Sketch (1) We show $\tilde{f}_t(\mathbf{W}_t)$ converges almost surely (a.s.) by showing it is a quasi-martingale. (2) We show $\|\mathbf{W}_t - \mathbf{W}_{t-1}\|_F = O(1/t)$ a.s. (3) We show $f(\mathbf{W}_t)$ converges a.s. by showing $f(\mathbf{W}_t) - \tilde{f}_t(\mathbf{W}_t)$ converges a.s. to zero. (4) We show the sequence $\{\mathbf{W}_t\}$ converges to $\mathbf{W}^*$ entrywise a.s., by showing $\mathbf{W}_t$ satisfies the optimality condition of (4) as $t \to \infty$.

Note that previous works [9, 19, 20] provide similar theoretical guarantees, but due to different objective functions, nonnegativity constraints and different optimization algorithms, our analysis is different from that in all previous works. In particular, we enforce sparsity on outlier vector $\mathbf{r}$ instead of coefficient vector $\mathbf{h}$, thus we need not assume the solution for the LASSO problem is unique as in [19]. Also, compared to [20], we have a different set of optimality conditions of (3) due to nonnegativity on $\mathbf{h}$ and $\mathbf{r}$. Similarly, these conditions are also different from that in [9] due to outliers. Moreover, the differences between our method to update dictionary $\mathbf{W}_t$ (cf. (6)) with that in [9, 19, 20] introduces many other differences in the proof. Details are defered to the extended version.

4. EXPERIMENTS

4.1. Experiment Setup

In our experiments, we used three datasets, namely the ORL face dataset [21], the UMIST face dataset [22] and the PIE face dataset [23, 24]. ORL, UMIST and PIE have 400, 300 and 3000 images respectively. Images are resampled to 32×32 pixels for ORL and UMIST and 20×20 pixels for PIE.

For implementation, the maximum gray level of all images in these three datasets was set to 50 and all pixel values were normalized accordingly. For each dataset, we randomly added entry-wise nonnegative outliers with density $\rho \in \{10\%, 20\%, 30\%, 40\%\}$ to each image, to simulate both sparse and dense outliers. All outliers were drawn i.i.d. from a uniform distribution $\mathcal{U}[30, 50]$. If the resultant pixel value exceeded the maximum gray level of 50, we thresholded it to 50.

4.2. Comparison to Other Algorithms

In this section we compare the performance of our algorithm ONMFO against other three classes of algorithms, namely, RNMF ONMF and ORPCA. We select one representative algorithm for each class, namely, our batch algorithm (in Section 3.1) for RNMF, [9] for ONMF and [20] for ORPCA. We use our own algorithm for RNMF to better demonstrate the differences between online and batch algorithms, since both of them use multiplicative updates.

To make a fair comparison, we ran all online algorithms (ONMFO, ONMF and ORPCA) for two passes on each face dataset. We fixed the latent dimensionality $K = 49$ for all algorithms; in fact, K can be learned automatically using the algorithm proposed in [25]. Furthermore, we fixed $\lambda = 1$ in our algorithm throughout all tasks. Each data point is an average result of 10 random initializations.

The running time of all algorithms on each dataset are shown in Figure 1. From Figure 1, we can observe the running time of ONMFO exhibit minimal variations across all outlier densities and they are the second shortest overall. Although ORPCA has a shorter running time than ONMFO, as we show later, it significantly underperforms ONMFO in basis representations and image reconstruction. This is because images are nonnegative data and NMF-type algorithms are more capable to deal with them.

Next we compare the basis learned by the four algorithms on the ORL dataset. We show results for $\rho = 10\%$ and $\rho = 40\%$ (in Figure 2 and Figure 3 respectively), in order to compare settings with sparse and dense outliers. When the outliers are sparse ($\rho = 10\%$), the basis learned by ONMFO is comparable to that by RNMF. Both bases are local and appear to be free of outliers. In contrast, the basis learned by ONMF appear to be noisy and not very local. When the outliers are dense ($\rho = 40\%$), somewhat surprisingly, we still manage to learn a local representation with most of the outliers removed. Note that RNMF has a slightly cleaner and more local representation, since it solves (1) exactly. In contrast, the basis learned by ONMF are largely contaminated by outliers, so that meaningful facial parts are hardly observed. We also notice in both cases, ORPCA

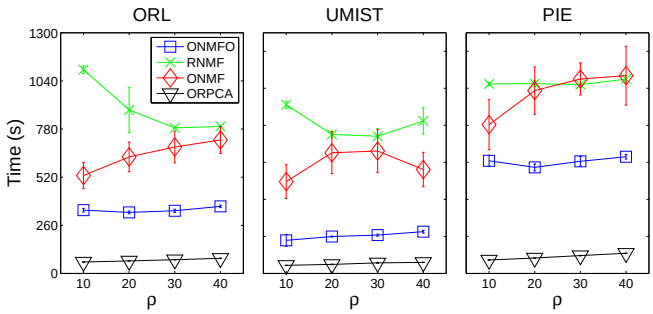
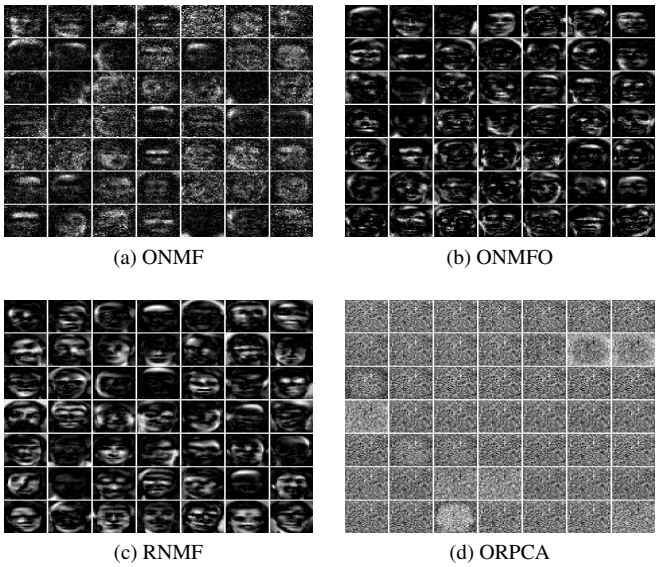


Fig. 1. Runnning time of all four algorithms in seconds.



(a) ONMF (b) ONMFO

(c) RNMF (d) ORPCA

Fig. 2. Basis representations of all four algorithms on the ORL dataset with outlier density 10% in one initialization.

fails to learn parts-based basis vectors.

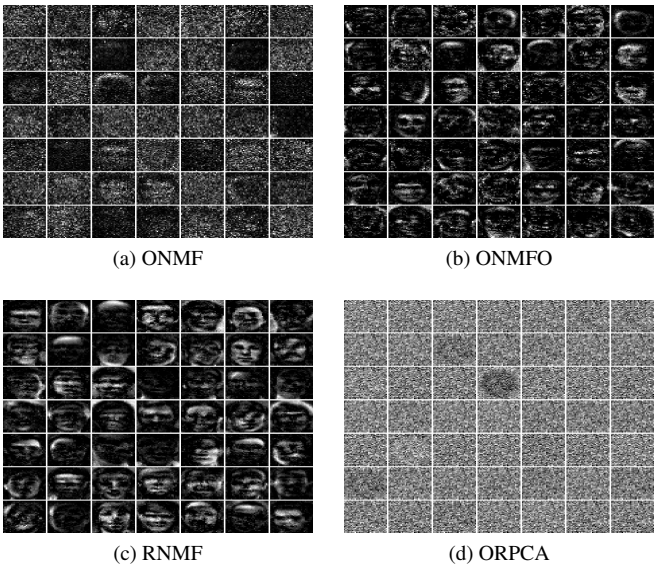
Finally we compare the quality of image reconstruction of all algorithms. We define *average reconstruction error* E_{avg} as

$$E_{\text{avg}} = \frac{1}{FN} \left\| \tilde{\mathbf{V}} - \mathbf{WH} \right\|_F^2, \tag{20}$$

where $\tilde{\mathbf{V}}$ denotes the original set of images without outliers. E_{avg} directly measures the extent of outlier removal, hence the quality of image reconstruction. (Note that the definition (20) is also valid for ORPCA.) Results are shown in Figure 4. From it, we observe that our algorithm slightly underperforms RNMF but outperforms the other two significantly. Similar to basis representations, as ρ increases, E_{avg} produced by ONMFO only degrades gently.

4.3. Discussions

We discuss two issues pertaining to the experiments. The first concerns the selection of the regularization weight λ. This is a long-standing problem in RNMF and so far no work has provided a principled way to select λ. As observed empirically, the performance of ONMFO was not sensitive to λ. Thus, for simplicity, we set $\lambda = 1$.



(a) ONMF (b) ONMFO

(c) RNMF (d) ORPCA

Fig. 3. Similar to Figure 2 with outlier density 40%.

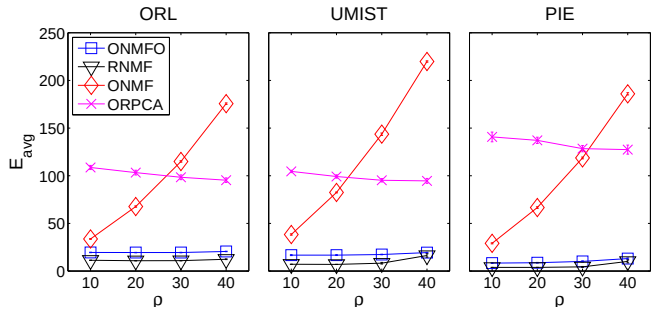


Fig. 4. Average reconstruction errors of all four algorithms.

Next, we explain why our algorithm (ONMFO) is able to maintain a reasonable performance when outliers are dense. This relates to the subspace recovery problem in the presence of outliers. In [26], the authors showed if the original data matrix $\tilde{\mathbf{V}}$ is low-rank and the support of outliers are uniformly distributed on $\tilde{\mathbf{V}}$, the subspace spanned by columns of $\tilde{\mathbf{V}}$ can be recovered up to *constant* outlier density $\tilde{\rho}$ with high probability, by solving

$$\begin{aligned} \min \quad & \|\tilde{\mathbf{V}}\|_* + \tilde{\lambda} \|\mathbf{R}\|_{1,1} \\ \text{subject to} \quad & \tilde{\mathbf{V}} + \mathbf{R} = \mathbf{V} \end{aligned} \tag{21}$$

In [27], the authors further added the Frobenius loss term in (5) to (21) and provided the same recovery guarantee. In addition, [28] showed $\tilde{\rho}$ can be any value in $(0, 1)$ under some weak assumptions, i.e., the corruptions can be *dense*. We observe the only difference between (1) and (21) is the nuclear norm term $\|\tilde{\mathbf{V}}\|_*$, which enforces $\tilde{\mathbf{V}}$ to be low–rank. By fixing K to a small value in our experiments, which is a common practice in other NMF works, we indeed enforce $\tilde{\mathbf{V}}$ to be low rank. This suggests our batch (RNMF) and online (ONMFO) algorithms are capable of dealing with a constant fraction of dense outliers. However, we remark that for real data, the technical conditions in [26] are not exactly satisfied, hence subspace recovery and outlier removal may be imperfect (shown in Figure 4).

5. REFERENCES

[1] S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita, "Dimensionality reduction using non-negative matrix factorization for information retrieval," in *IEEE SMC*, 2001, vol. 2, pp. 960–965.

[2] Chris Ding, Xiaofeng He, and Horst D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *SDM*, 2005.

[3] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2000, pp. 556–562.

[4] Jingu Kim and Haesun Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *IEEE ICDM*, Dec 2008, pp. 353–362.

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

[6] Hyunsoo Kim and Haesun Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. A.*, vol. 30, no. 2, pp. 713–730, July 2008.

[7] Bin Cao, Dou Shen, Jian-Tao Sun, Xuanhui Wang, Qiang Yang, and Zheng Chen, "Detect and track latent factors with online nonnegative matrix factorization," in *IJCAI*, 2007, pp. 2689–2694.

[8] Serhat S. Bucak and Bilge Gunsel, "Incremental subspace learning via non-negative matrix factorization," *Pattern Recogn.*, vol. 42, no. 5, pp. 788–797, May 2009.

[9] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1087–1099, July 2012.

[10] A Lefèvre, F. Bach, and C. Févotte, "Online algorithms for nonnegative matrix factorization with the itakura-saito divergence," in *IEEE WASPAA*, Oct 2011, pp. 313–316.

[11] Fei Wang, Chenhao Tan, Arnd Christian Konig, and Ping Li, "Efficient document clustering via online nonnegative matrix factorizations," in *SDM*, April 2011.

[12] Yi Wu, Bin Shen, and Haibin Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 374–383, March 2014.

[13] Bin Shen, Baodi Liu, Qifan Wang, and Rongrong Ji, "Robust nonnegative matrix factorization via l_1 norm regularization by multiplicative updating rules," in *IEEE ICIP*, Oct 2014.

[14] Liang Du, Xuan Li, and Yi-Dong Shen, "Robust nonnegative matrix factorization via half-quadratic minimization," in *IEEE ICDM*, Dec 2012, pp. 201–210.

[15] Deguang Kong, Chris Ding, and Heng Huang, "Robust nonnegative matrix factorization using ℓ_{21}-norm," in *ACM CIKM*, 2011.

[16] Cèdric Févotte and Nicolas Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *ArXiv preprint*, 2014.

[17] L. Zhang, Z. Chen, M. Zheng, and X. He, "Robust nonnegative matrix factorization," *Frontiers of Elect. and Electron. Eng. in China*, vol. 6, no. 2, pp. 192–200, 2011.

[18] David R. Hunter and Kenneth Lange, "Quantile regression via an mm algorithm," *J. Comput. Graphical Stat.*, pp. 60–77, 2000.

[19] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, March 2010.

[20] Jiashi Feng, Huan Xu, and Shuicheng Yan, "Online robust pca via stochastic optimization," in *NIPS*, pp. 404–412. 2013.

[21] Ferdinando Samaria and Andy Harter, "Parameterisation of a stochastic model for human face identification," in *IEEE WACV*, Dec 1994.

[22] Daniel B Graham and Nigel M Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," *Face Recogn.: From Theory to Applicat.*, vol. 163, pp. 446–456, 1998.

[23] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S Baker, "Multi-pie," in *IEEE AFGR*, 2008.

[24] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S Baker, "Multi-pie," in *Image and Vision Comput.*, 2009.

[25] V. Y. F. Tan and C. Févotte, "Automatic relevance determination in nonnegative matrix factorization with the β-divergence," *IEEE Trans. Pattern Anal.*, vol. 35, no. 7, pp. 1592–1605, July 2013.

[26] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, May 2011.

[27] Daniel Hsu, Sham M. Kakade, and Tong Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Trans. Info. Theory*, 2011.

[28] Arvind Ganesh, John Wright, Xiaodong Li, Emmanuel Candes, and Yi Ma, "Dense error correction for low-rank matrices via principal component pursuit," in *ISIT*, 2010.