Greedy Orthogonal Pivoting Algorithm for Non-negative Matrix Factorization

Kai Zhang [1] **Jun Liu** [2] **Jie Zhang** [3] **Jun Wang** [1]

Abstract

Non-negative matrix factorization is a powerful tool for learning useful representations in the data and has been widely applied in many problems such as data mining and signal processing. Orthogonal NMF, which can further improve the locality of decomposition, has drawn considerable interest in clustering problems. However, imposing simultaneous non-negative and orthogonal structure can be difficult, and so existing algorithms can only solve it approximately. To address this challenge, we propose an innovative procedure called Greedy Orthogonal Pivoting Algorithm (GOPA). The GOPA method fully exploits the sparsity of non-negative orthogonal solutions to break the global problem into a series of local optimizations, in which an adaptive subset of coordinates are updated in a greedy, closed-form manner. The biggest advantage of GOPA is that it promotes exact orthogonality and provides solid empirical evidence that stronger orthogonality does contribute favorably to better clustering performance. On the other hand, we have designed randomized and batch-mode version of GOPA, which can further reduce the computational cost and improve accuracy, making it suitable for large data.

1. Introduction

Non-negative matrix factorization (Lee & Seung, 2000; Ding et al., 2006; Wang & Zhang, 2013) is a powerful tool for learning useful representations. Given an $n \times d$ data matrix $\mathbf{X}$ where each column is one feature and each row is one data point, the main idea behind NMF is to approximate these input vectors by nonnegative linear combinations of nonnegative basis vectors (rows of of $\mathbf{H}$) with the coefficients stored in columns of $\mathbf{W}$, as follows

$$\begin{aligned} \min_{\mathbf{W},\mathbf{H}} \quad & \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 \\ s.t. \quad & \mathbf{W} \geq 0, \mathbf{H} \geq 0. \end{aligned}$$

By imposing simultaneous non-negative structures in data reconstruction and basis identification, NMF has shown great potential in unravelling important structures in the data from various applications, such as data clustering (Kuang et al., 2012; Li & Ding, 2006), image processing (Lee & Seung, 2000), text mining (Xu et al., 2003; Ding et al., 2008), and signal processing (Ozerov & Fevotte, 2010). See a recent review in (Wang & Zhang, 2013).

The NMF is not jointly convex with $\mathbf{W}$ and $\mathbf{H}$, and various approaches have been proposed to solve it. Early attempt mainly focused on multiplicative updates. For example, Lee & Seung (2000) proposed the following procedures

$$\mathbf{W}_{ij} = \mathbf{W}_{ij} \odot \frac{(\mathbf{X}\mathbf{H}')_{ij}}{(\mathbf{W}\mathbf{H}\mathbf{H}')_{ij}},$$
$$\mathbf{H}_{ij} = \mathbf{H}_{ij} \odot \frac{(\mathbf{W}'\mathbf{X})_{ij}}{(\mathbf{W}'\mathbf{W}\mathbf{H})_{ij}},$$

and proved its convergence using the technique of auxiliary functions. More variations of multiplicative updates can be found in (Yang & Oja, 2011). In order to further improve the convergence rate and quality, many other optimization strategies were proposed, such as the projected gradient descent (Lin, 2007), alternating least squares (Cichocki & Anh-Huy, 2009), active set method (Kim & Park, 2008a;b), block-coordinate descent (Kim et al., 2014), and greedy coordinate descent (Hsieh & Dhillon, 2011).

NMF has been shown to generate part based representation in many applications such as image and signal processing, where data points are reconstructed using a limited number of components that can be easy to interpret. However, this may not be guaranteed for several reasons. First, the solution of NMF is not unique (even consider re-scaling), and extra constraints are needed to obtain (more) well-posed NMF problems. Indeed, it is found that the sample points have to fill a proper subset of the positive orthant such that a

[1]Shanghai Key Laboratory for Trustworthy Computing, School of Computer Science and Software Engineering, East China Normal University, Shanghai, China [2]Infinia ML Inc., Durham, North Carolina, USA [3]Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai, China. Correspondence to: Kai Zhang <kzhang980@gmail.com>, Jun Wang <jwang@sei.ecnu.edu.cn>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

unique simplicial cone corresponding to the NMF solution exists (Donoho & Stodden, 2004). Another reason is that solutions of NMF may not always be sparse since there is no direct control over sparsity of solutions, and as a result the decomposition can have a global support (Gillis, 2012).

In order to solve these problems, much attention has been put on incorporating extra locality constraints to improve quality of decomposition. For example, the sparsity-inducing norms (such as ℓ_0 and ℓ_1 norm) have been incorporated in NMF to improves the found decompositions (Hoyer, 2004; Peharz & Pernkopf, 2012); on the other hand, Cai et al. (2011) incorporated the graph-based manifold regularization such that the decomposition is more aligned with the manifold structures.

In this paper we focus on improving the decomposition of NMF using orthogonality constraints, namely the factor matrix should have orthonormal columns. Given a data matrix $\mathbf{X}$, the (uni-)orthogonal NMF is defined as

$$\begin{aligned} \min_{\mathbf{W},\mathbf{H}} \quad & \|\mathbf{X}-\mathbf{W}\mathbf{H}\|^2 \\ s.t. \quad & \mathbf{W}\geq 0,\ \mathbf{H}\geq 0,\ \mathbf{W}'\mathbf{W}=\mathbf{I}. \end{aligned} \tag{1}$$

Orthogonal NMF can effectively control the model complexity and lead to unique solutions (Ding et al., 2006). Indeed, orthogonal NMF has a intrinsic connection to clustering considering that the cluster indicator matrix takes exactly the form of an orthogonal matrix (Kuang et al., 2012; Li & Ding, 2006). Furthermore, empirical evidence shows that it performs remarkably well in certain clustering tasks, such as document classification (Xu et al., 2003).

However, the coupled non-negative and orthogonal constraints can be challenging to optimization, and various algorithms have been proposed. Early efforts mainly focused on multiplicative updates. For instance, Ding et al. (2006) first proposed a general proof on the equivalence between k-means clustering and orthogonal NMF, and further generalized it to co-clustering. They pioneered the use of multiplicative update in approximately solving the NMF with hard orthogonality constraint

$$\begin{aligned} \mathbf{H}_{ij} &= \mathbf{H}_{ij}\odot\frac{(\mathbf{X}'\mathbf{W})_{ij}}{(\mathbf{H}\mathbf{W}'\mathbf{W})_{ij}}, \\ \mathbf{W}_{ij} &= \mathbf{W}_{ij}\odot\frac{(\mathbf{X}\mathbf{H})_{ij}}{(\mathbf{W}\mathbf{W}'\mathbf{X}\mathbf{H})_{ij}}. \end{aligned}$$

Convergence of such updates and correctness of solution is given in (Ding et al., 2006). Other examples of multiplicative updates includes (Yoo & Choi, 2008).

In Multiplicative updates once an entry becomes zero it will keep vanishing throughout subsequent iterations, which is called zero-locking and may lead to premature convergence. To solve this problem and also improve the rate of convergence, a number of optimization approaches have been proposed. For example, Kimura et al. (2014) adopted the hierarchical alternating least squares to update column-wise update of the orthogonal factor matrix. The Lagrangian of (1) with local orthogonal constraints (i.e., involving only the inner product between one column and the rest columns in $\mathbf{W}$) is solved. Since exact Lagrangian multiplier associated with the orthogonality constraint is very difficult to determine, an approximation is used which is similar to that in (Ding et al., 2006).

In order to avoid solving the Lagrangian multiplier with hard orthogonality constraint, a soft penalty term can be used

$$\begin{aligned} \min_{\mathbf{W},\mathbf{H}} \quad & \|\mathbf{X}-\mathbf{W}\mathbf{H}\|^2+\beta\cdot\|\mathbf{W}'\mathbf{W}-\mathbf{I}\|, \\ s.t. \quad & \mathbf{W}\geq 0,\ \mathbf{H}\geq 0. \end{aligned}$$

Here β is the regularization parameter that controls the orthogonality of solution. For example, Shiga et al. (2016) adopted the soft orthogonal constraint together with an automatic relevance determination (ARD) prior in Gaussian noise for NMF to identifying the potential constituent chemical components from spectral imaging. Shiga et al. (2014) generalized the projected gradient descent method (Lin, 2007) where a modified version of the additive update rules are designed to solve orthogonal NMF, with rigorous proof of convergence. These algorithms do not have zero locking problem and perform better in clustering tasks.

Besides multiplicative updates and soft orthogonal constraints, other methods have been proposed as well. Pompili et al. (2014) proved equivalence between (uni-)orthogonal NMF and a weighted version of spherical k-means clustering algorithm, and designed an EM-iteration to solve the clustering problem where the hard cluster assignment can be deemed as an orthogonal solution; Asteris et al. (2015) proposed an approximation to Nonnegative Principal Component Analysis to solve orthogonal NMF with provable guarantees; a large penalty term needs to be enforced to promote orthogonality.

Overall, computing an orthogonal non-negative factorization is still challenging. Almost all the existing orthogonal NMF methods can only approximately enforce the orthogonality constraint; on the other hand, the level of orthogonality may not be explicitly controllable. In this paper, we propose a new and systematic perturbation scheme for more effective orthogonal NMF. By fully exploiting the sparse and exclusive structures of orthogonal solutions, we decompose the original problem into a series of local optimization, in which an adaptive subset of coordinates are updated in a greedy, closed-form manner. Our method is called Greedy Orthogonal Pivoting Algorithm (GOPA), which has several advantages. First, each iteration operates on a small subset of variables with closed-form solution,

making the implementation easy; second, the iterations can be parallelized to different levels, making it very computationally efficient; finally, our approach can enforce exact orthogonal solutions and show significant performance gains in clustering problems against state-of-the-arts.

The rest paper is organized as follows. Section 2 introduces the proposed methods, including the sequential GOPA method and its randomized version in Section 2.1, the batch GOPA method in Section 2.2, complexity analysis in Section 2.3 and convergence analysis in Section 2.4. Section 3 reports empirical evaluations and Section 4 concludes the paper and points our several interesting future research directions.

2. Greedy Orthogonal Pivoting Algorithm

2.1. Sequential GOPA

Given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ with n samples and d features, consider non-negative factorizations of the form $\mathbf{X} \approx \mathbf{WH}$ to solve the clustering problem, where $\mathbf{W} \in \mathbb{R}^{n\times k}$ is an orthonormal cluster indicator matrix such that $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$, and $\mathbf{H} \in \mathbb{R}^{k\times d}$ contains cluster centers as its rows, with k being the number of clusters.

Given a feasible solution $\mathbf{W}$ where each row has exactly one non-negative entry and each column has unit norm. Consider updating a single row in $\mathbf{W}$, namely, switching the non-zero entry in this row from the original location to another location (column), probably re-scaled numerically, and then see if this update can potentially reduce the objective. Such perturbation naturally satisfies the orthogonality constraint, since each row will always be filled with only one non-negative entry after each switching operation. In practice, one needs to attempt k switches for each row (i.e., including the current location considering the possible numerical changes), such that the optimal location of the non-zero entry and its optimal value can be determined.

We illustrate the basic idea in Figure 1. Suppose we want to update the lth row in $\mathbf{W}$, where without loss of generality the qth column is assumed to be non-zero, and we want to switch it to the pth column with value $x \in (0, 1)$. Here $\mathbf{W}_{cp}$ is the sub-vector corresponding to the non-zeros in the pth column of $\mathbf{W}$, and $\mathbf{W}_{cq}$ represents the non-zeros excluding the lth entry in the qth column of $\mathbf{W}$. Note that both the pth and qth column of $\mathbf{W}$ will be re-scaled correspondingly after swapping the non-zero entry, in order to guarantee the unit norm condition as shown Figure 1. Our goal is to calculate an optimal x for this switch and the resultant change of the objective value.

Note that throughout the updating procedures the solution always reside in the feasible domain, i.e., $\mathbf{W}'\mathbf{W} = \mathbf{I}$, and as a result the objective function (1) can be written equivalently as

$$\begin{aligned} \max_{\mathbf{W}} \quad & \text{tr}(\mathbf{WHX}') \\ s.t. \quad & \mathbf{W} \geq 0. \end{aligned}$$

Let $\mathbf{R} = \mathbf{XH}'$ and $\mathbf{R}_p$ the pth column of $\mathbf{R}$. Then trace objective before the switch can be written equivalently as

$$J = \text{tr}(\mathbf{W}'_p\mathbf{R}_p) + \text{tr}(\mathbf{W}'_q\mathbf{R}_q) + \sum_{j\neq p,q} \text{tr}(\mathbf{W}'_j\mathbf{R}_j).$$

After swapping, the objective becomes

$$\widetilde{J} \quad = \quad \text{tr}(\widetilde{\mathbf{W}}'_p\mathbf{R}_p) + \text{tr}(\widetilde{\mathbf{W}}'_q\mathbf{R}_q) + \sum_{j\neq p,q} \text{tr}(\mathbf{W}'_j\mathbf{R}_j).$$

Let $\mathcal{I}_p$ be the index of non-zero entries in $\mathbf{W}_p$, $\mathbf{R}_{p[\mathcal{I}_p]}$ be the corresponding sub-vector in $\mathbf{R}_p$, and $\mathbf{R}_{p[l]}$ be the lth entry in $\mathbf{R}_p$. By removing the constant part of $\widetilde{J}$ (i.e., the third term), the objective can be written as follows,

$$\max_{0<x<1} \sqrt{1-x^2}(\mathbf{W}'_{cp}\mathbf{R}_{p[\mathcal{I}_p]}) + x\mathbf{R}_{p[l]}. \tag{2}$$

Through simple derivations we can obtain a closed form solution for x, as

$$x^* = \frac{\mathbf{R}_{p[l]}}{\sqrt{\mathbf{R}^2_{p[l]} + (\mathbf{W}'_{cp}\mathbf{R}_{p[\mathcal{I}]})^2}}. \tag{3}$$

which automatically satisfies $x \in [0, 1]$ since $\mathbf{R} = \mathbf{XH}'$ and both $\mathbf{X}$ and $\mathbf{H}$ and $\mathbf{W}$ is non-negative too.

The change of the objective after updating the lth row of $\mathbf{W}$ can be computed as

$$\widetilde{J} - J = \text{tr}\left((\widetilde{\mathbf{W}}_p - \mathbf{W}_p)'\mathbf{R}_p\right) + \text{tr}\left((\widetilde{\mathbf{W}}_q - \mathbf{W}_q)'p\mathbf{R}_q\right).$$

Another possibility is that the non-zero entry in the lth row of $\mathbf{W}$ only needs to be re-scaled but does not have to be moved to another column. In such case, we will have the following objective function

$$\max_{0<x<1} \quad \sqrt{1-x^2}\mathbf{W}'_{cq}\mathbf{R}_{q[\mathcal{I}_q]} + x \cdot \mathbf{R}_{q[l]}. \tag{4}$$

Here $\mathcal{I}_q$ is the index of the non-zero entries in $\mathbf{W}_q$ excluding the lth entry. This problem is very similar to that in (3) and the closed-form solution can be written as

$$x^* = \frac{\mathbf{R}_{q[l]}}{\sqrt{\mathbf{R}^2_{q[l]} + (\mathbf{W}'_{cq}\mathbf{R}_{q[\mathcal{I}_q]})^2}}.$$

By trying all the k possible switches, and for each swap we examine the decrement of the objective, we can then choose the optimal switch. If none of the k swaps leads to an improved objective, the lth row remains unchanged.

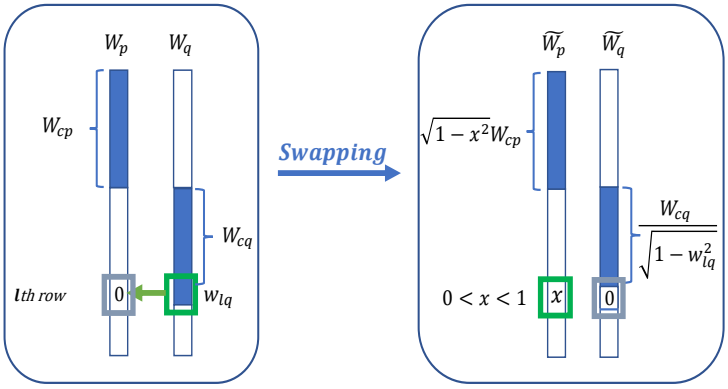


Figure 1. Illustration of the basic idea of greedy orthogonal pivoting algorithm (GOPA) in updating one row of an orthogonal $\mathbf{W}$. GOPA updates the lth row by shifting the non-zero entry from qth column to the pth column where p traverses through all the k columns of $\mathbf{W}$.

On the other hand, if the best switch is associated with an optimal solution of $x^* = 0$ or $x^* = 1$, the solution will be discarded because the orthogonality condition would be violated. The update of the $\mathbf{H}$ matrix is fairly simple, i.e., $\mathbf{H} = \mathbf{W}'\mathbf{X}$ since $\mathbf{W}$ has orthonormal columns.

The detailed algorithm is presented in Algorithm 1. We call it "Greedy Orthogonal Pivoting Algorithm" (GOPA) since it fully exploits the structure of non-negative orthogonal solutions and adopts a greedy scheme to determine the location and value of the non-zero entry in each row of $\mathbf{W}$.

Algorithm 1 Greedy Orthogonal Pivoting Algorithm.
Input. Data matrix $\mathbf{X}$, initial orthogonal $\mathbf{W} \in \mathbb{R}^{n\times k}$.

for $t = 1, t++, t \leq T$ **do**
  for each $i \in [1, n]$ **do**
    Swap the non-zero entry of $\mathbf{W}_{[i,:]}$ from the original location (qth column) to each of the $\{1, 2, ..., k\}$ locations;
    Calculate optimal x^* (3) for each swap;
    Place the non-zero entry in the pth column with minimum objective (4);
    Re-scale pth and qth column of $\mathbf{W}$ to norm 1;
  end for
  Update matrix $\mathbf{H}$ by $\mathbf{H} = \mathbf{W}'\mathbf{X}$.
end for

In figure 2, we illustrate the behaviour GOPA in comparison with two popular multiplicative update algorithms (Ding et al., 2006; Choi, 2008). We star from an initial $\mathbf{W}$ where each row only has a single non-zero entry that is randomly located. All three algorithms here can preserve the orthogonality of solutions. However, multiplicative updates suffer from zero-locking and quickly reach premature convergence; in comparison, GOPA procedures can significantly reduce the objective even starting from such a sparse initial $\mathbf{W}$ matrix.

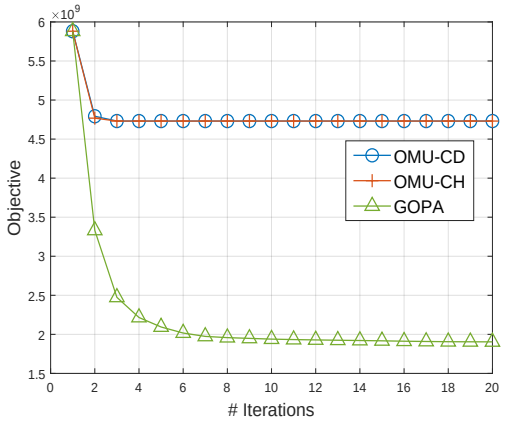


Figure 2. Convergence behaviour of multiplicative updates and GOPA on one benchmark data. When starting from a non-negative orthogonal solution of $\mathbf{W}$, multiplicative updates quickly reaches pre-mature convergence, while the GOPA can substantially decrease the objective.

Randomized GOPA. The sequential updating procedures of GOPA can be randomized by selecting only a subset of matrix rows (and resultant columns) to update in each round of iterations. By doing this, the iterations can have the potential to jump out of local optima in a similar way to the stochastic gradient descent, and in the meantime the computational cost can be reduced as well.

In Figure 3, we plot the objective function corresponding to different choices of the updating ratio. As can be expected, a smaller update ratio may require more iterations to converge; however, the final objective is similar or even (slightly) superior to that of using a full update. Empirically, using a properly selected update ratio (e.g., 0.5) can simultaneously save computations and improve convergence, which has been observed in the majority of the benchmark

data sets.

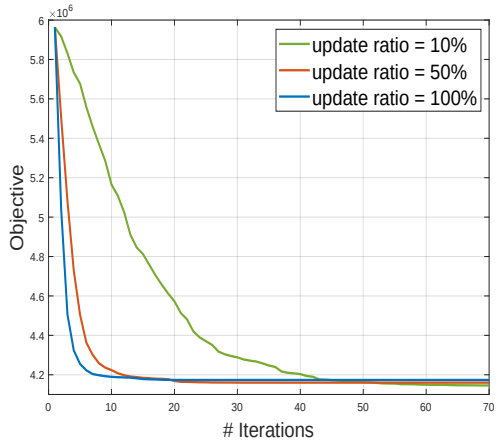


Figure 3. Randomized GOPA with different random updating ratio on a benchmark data. A smaller ratio (i.e. incomplete updating) can potentially save computation and improve convergence.

2.2. Batch-Mode GOPA

The greedy orthogonal pivoting updates one row of $\mathbf{W}$ at a time and proceeds sequentially. In practice, it can be desirable if a subset of rows can be updated simultaneously. We therefore consider generalizing the GOPA to a batch mode.

Exact generalization of GOPA to batch-mode can be challenging due to its sequential nature. This is because updating one row of $\mathbf{W}$ can involve perturbing two columns in $\mathbf{W}$, corresponding to the original and target locations of the non-zero entry in that row. As a result, subsequent pivoting steps will all be affected. We therefore drop such sequential constraints and allow several rows in $\mathbf{W}$ to be updated simultaneously. More specifically, we first apply the pivoting procedure on the rows of $\mathbf{W}$ and record the updated location of the non-zero entries for each row. This step can be done in parallel for all the rows. We then simultaneously calculate values of the non-zero entries that are located in the same column. We can further randomize the procedure by performing the updates on only a subset of the rows.

Figure 4 illustrates the batch-mode GOPA procedure. We first introduce randomization in each round of iterations by picking out $n \cdot (1 - r\%)$ rows from $\mathbf{W}$, which do not go through any pivoting; namely we only perform GOPA on the remaining $n \cdot r\%$ rows, whose non-zero entries will possibly be shifted to a new column. Then we reorder the rows such that their non-zero entries are aligned by column. For each of $p = 1, 2, ..., k$, we collect the rows whose non-zero entries are located in the pth column, and divide their indices into $\mathcal{I}_p^c$ and $\mathcal{I}_p^u$, corresponding to the rows without and with the pivoting operation, respectively; related sub-vectors are defined as $\mathbf{W}_p^c = \mathbf{W}_{p[\mathcal{I}_p^c]}$ and $\mathbf{W}_p^u = \mathbf{W}_{p[\mathcal{I}_p^u]}$.

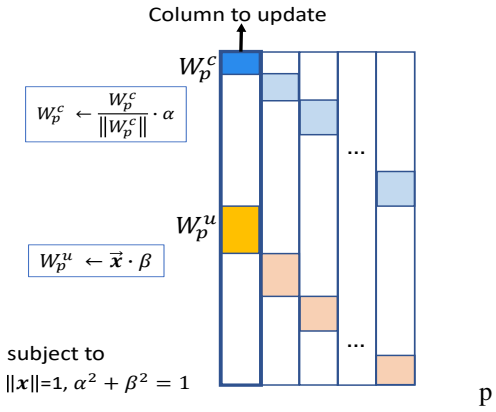


Figure 4. Batch mode update of matrix $\mathbf{W}$. In each column, the non-zero entries are divided into two sub-vectors: $\mathbf{W}_p^c$, which is only subject to a concurrent re-scaling, and $\mathbf{W}_p^u$, whose entries are determined by the optimization problem (5).

Based on the updated locations of the non-zero entries of all the rows, we can then formulate the optimization problem as follows. First, entries in $\mathbf{W}_p^c$ will remain unchanged and will only be subject to a re-scaling to guarantee the unit-norm for each column; in contrast, entries in $\mathbf{W}_p^u$ will be sufficiently optimized in order to reduce the objective function. Let the pth column of $\mathbf{R} = \mathbf{X}\mathbf{H}'$ be $\mathbf{R}_p$. Then our goal is to optimize the following objective

$$\begin{aligned} \max_{\alpha,\beta,\mathbf{x}} \quad & \alpha \cdot e + \beta \cdot \mathbf{x}'\mathbf{u} \\ s.t. \quad & \alpha^2 + \beta^2 = 1, \ \ \|\mathbf{x}\| = 1. \\ & \mathbf{u} = \mathbf{R}_p(\mathcal{I}_p^u), \ \ e = \frac{\mathbf{W}_p^{c\prime}\mathbf{R}_p(\mathcal{I}_p^c)}{\|\mathbf{W}_p^c\|}. \end{aligned} \tag{5}$$

To solve the above problem, write the Lagrangian of the objective as

$$J = \alpha e + \beta \mathbf{x}'\mathbf{u} - \lambda_1(\alpha^2 + \beta^2 - 1) - \lambda_2(\mathbf{x}'\mathbf{x} - 1).$$

By setting the derivative to zeros, we have the following conditions:

$$e = 2\alpha\lambda_1, \ \mathbf{x}'\mathbf{u} = 2\beta\lambda_1, \ \beta\mathbf{u} = 2\lambda_2\mathbf{x}.$$

By combining these conditions with $\alpha^2 + \beta^2 = 1$ and $\|\mathbf{x}\| = 1$, we have the following closed-form solution

$$\mathbf{x} = \frac{\mathbf{u}}{\|\mathbf{u}\|}, \ \ \alpha = \frac{e}{e^2 + \|\mathbf{u}\|^2}, \ \ \beta = \sqrt{1 - \alpha^2}.$$

In Figure 5, we plot the objective with different choices of the updating ratio in batch-mode GOPA. Again, a smaller update ratio requires more iterations but the final objective is similar or even superior to that of using a full update, and can be computationally cheaper in the meantime.

Double-Sided Orthogonality. In case co-clustering is considered (Kuang et al., 2012), we want to enforce orthogonality on both the $\mathbf{W}$ and the $\mathbf{H}$ matrix, to which the GOPA method can be extended trivially.

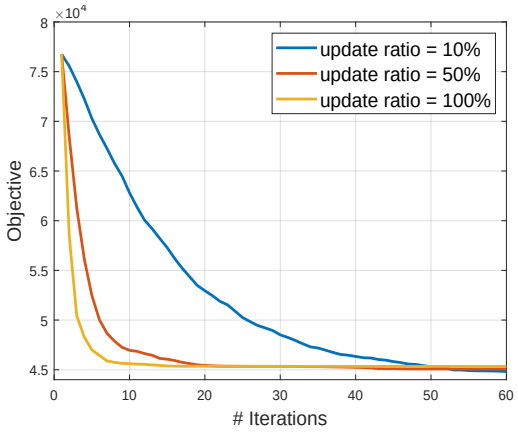


Figure 5. The convergence behaviour of batch-mode GOPA with different choices of the updating ratio on a benchmark data. A smaller update ratio gives equally good (or better) objective and can potentially save computational cost when properly selected.

Algorithm 2 Batch-mode Greedy Orthogonal Pivoting.
Input. Data $\mathbf{X}$, initial orthogonal $\mathbf{W} \in \mathbb{R}^{n\times k}$, ratio $r\%$.

1: Perform GOPA on $n \cdot r\%$ randomly selected rows in matrix $\mathbf{W}$;
2: **for** $t = 1, t++, t \leq T$ **do**
3: **for** $p \in [1, k]$ **do**
4: Collect rows whose non-zero entries are located in the pth column, indexed by π_p;
5: For the pth column of $\mathbf{W}$, update rows in π_p (i.e., $\mathbf{W}_p^c$ and $\mathbf{W}_p^u$) using (5);
6: **end for**
7: Update matrix $\mathbf{H}$ by $\mathbf{H} = \mathbf{W}'\mathbf{X}$.
8: **end for**

2.3. Complexity

For the sequential GOPA (Algorithm 1), given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and a pre-defined rank k, the complexity is $O\left(T(rnkb + ndk)\right)$, where T is the number of iterations, r is the updating ratio in each round of iterations, and b is the average cluster size (which is close to $\frac{n}{k}$ in case of balanced clusters). Note that the k switches of the non-zero entry associated with each row of $\mathbf{W}$ is totally independent, therefore they can be implemented in a parallel fashion and the time complexity will then be reduced to $O\left(T(rnb + ndk)\right)$ if there are k concurrent workers.

For the batch-mode GOPA (Algorithm 2), there can be two levels of parallelization. First, the pivoting of each single row requires calculating the optimal non-zero entry for all the k columns, which can be done in parallel. Second, The pivoting of multiple rows can also be performed independent of each other. Suppose there are $w_1 w_2$ workers, where each group of w_2 workers calculate the pivoting for a single row in $\mathbf{W}$. Then the complexity of the parallel GOPA method can be written as $O(T(\frac{rnkb}{w_1 w_2} + ndk))$.

2.4. Convergence

Each orthogonal pivot starts from switching the non-zero entry in one row of $\mathbf{W}$ from the original column to a destination column, which may end up with updating either one or two columns of $\mathbf{W}$, depending on the change of the objective of each switch and feasibility of solution. Therefore, the GOPA method may be considered as a block coordinate descent. However, there are two special properties that make it different from standard coordinate descent method. First, the subset of coordinates in each step are selected adaptively based on the current model ($\mathbf{W}^{(t)}$), instead of using a fixed scheme (e.g. cyclic or random choices); second, in determining which column(s) to update when working with one row of $\mathbf{W}$, the GOPA procedure will actually choose the best step among all the k possible switches. Therefore, although convergence of block coordinate descent has been well studied in the literature (Beck & Tetruashvill, 2013; Tseng, 2001), generalizing it to GOPA method is highly non-trivial and is still being pursued.

Indeed, the sequential GOPA algorithm is guaranteed to decrease the objective (or preserve the same objective) in each step. This is because in working with each row of $\mathbf{W}$, the final switch is chosen to be the best step among all the k possible switches; in case none of these switches leads simultaneously to a feasible solution and decreased objective, then no update is performed. Therefore, by combining continuous optimization (obtaining the non-zero entry value x^*) and greedy selection of block coordinates, the sequential GOPA method will efficiently converge in a finite number of iterations since the objective is always non-negative. For batch-mode GOPA, due to simultaneous updates of the non-zero entries in $\mathbf{W}$, a non-increasing objective may not be strictly assured during the iterations. However, empirically, non-increasing and converging objective is always observed for batch-mode GOPA when the update ratio is properly selected (e.g., 0.5 or larger). Indeed, the batch-mode outperforms the sequential version on a large portion of the benchmark data set, as reported in Section 3, demonstrating its ability to better combat local optima.

3. Experiments

In this section we perform empirical evaluations. We have compared both sequential and batch GOPA with altogether 6 state-of-the-art algorithms for orthogonal NMF. The first two methods are multiplicative updates; The EM-ONMF

Table 1. Averaged clustering accuracy and standard deviation for different orthogonal NMF algorithms on benchmark datasets over 30 repeated runs. All algorithms start from random initialization.

Dataset	OMU-CD	OMU-CH	OAUR	HALS-ONMF	SOARD	EM-ONMF	Seq. GOPA	Bat. GOPA
RCV1	65.40±9.45	66.63±8.83	69.14±5.16	70.63±8.16	70.13±8.20	66.06±10.57	69.37±7.18	**71.69±3.84**
Reuters	23.97±1.75	25.36±1.37	23.74±1.52	23.50±1.01	23.31±1.06	25.14±1.86	25.86±1.28	**30.12±1.23**
Newsgp4	28.81±4.16	29.08±4.42	29.78±2.38	29.98±4.10	30.73±3.62	38.58±6.93	36.36±6.61	**41.91±5.50**
Webkb4	71.61±9.32	76.92±6.38	76.36±2.95	**77.53±7.27**	**77.43±7.53**	74.51±8.66	**77.39±5.67**	76.90±7.32
WebACE	50.13±3.91	49.12±4.06	47.66±4.72	46.59±2.86	46.69±2.38	**51.49±5.53**	49.57±3.81	**51.89±5.20**
TDT2	49.02±2.70	45.01±2.17	41.66±1.24	46.51±1.65	46.24±1.73	54.06±3.74	48.92±2.07	**55.13±2.73**
UCI	50.18±3.78	47.32±3.61	46.98±3.88	76.48±3.82	75.01±3.35	73.94±6.21	**80.44±5.13**	**79.59±4.09**
USPS	57.93±4.82	61.50±4.39	48.18±3.77	62.28±2.39	65.44±2.31	65.41±3.20	65.65±3.78	**67.44±3.71**
DNA	68.04±7.05	65.73±9.47	62.25±6.16	72.18±4.29	71.05±6.07	73.77±3.62	**74.80±3.77**	**75.09±2.73**
Connect	36.72±1.73	35.54±0.95	34.90±0.75	36.42±0.89	36.49±1.09	**37.60±1.26**	**37.58±1.22**	**38.24±1.64**
Protein	**42.17±0.86**	40.88±1.90	41.25±2.37	40.91±1.59	40.22±1.56	41.43±1.10	**42.13±1.10**	41.63±1.49
Coil20	32.29±1.04	32.77±1.15	37.52±1.22	**61.04±2.05**	59.57±2.53	56.28±5.46	**61.22±1.65**	59.86±4.16
20News	16.27±2.14	15.74±1.94	15.04±1.50	**17.01±0.64**	**16.86±0.73**	**16.91±1.74**	15.38±0.98	16.37±1.23

can achieve exact orthogonal solutions because it adopts a clustering framework, and the other three methods use soft orthogonality constraints. Our matlab codes can be found at https://github.com/kzhang980/ORNMF.

1. Orthogonal multiplicative update (OMU-1) proposed by (Ding et al., 2006);
2. Orthogonal multiplicative update (OMU-2) proposed by (Choi, 2008);
3. Orthogonal NMF with additive update rules (OAUR) proposed by (Shiga et al., 2014);
4. Orthogonal EM-based NMF (EM-ONMF) proposed by (Pompili et al., 2014);
5. Fast Hierarchical Alternating Least Squares for Orthogonal NMF (HALS-ONMF) (Kimura et al., 2014);
6. Soft Orthogonal NMF with automatic relevance determination prior (SOARD) (Shiga et al., 2016).

Table 2. Benchmark datasets

Dataset	Number of Clusters	Number of Features	Sample Size
RCV1	4	9625	29992
Reuters	65	8293	18933
20Newsgp	20	18846	26214
Newsgp4	4	3946	1000
Webkb4	4	4190	1000
WebACE	20	2340	1000
TDT2	30	9394	36771
UCI	10	3823	64
USPS	10	7291	256
DNA	3	2000	180
Connect	3	6756	126
Protein	3	4965	357
Coil20	20	1440	1024

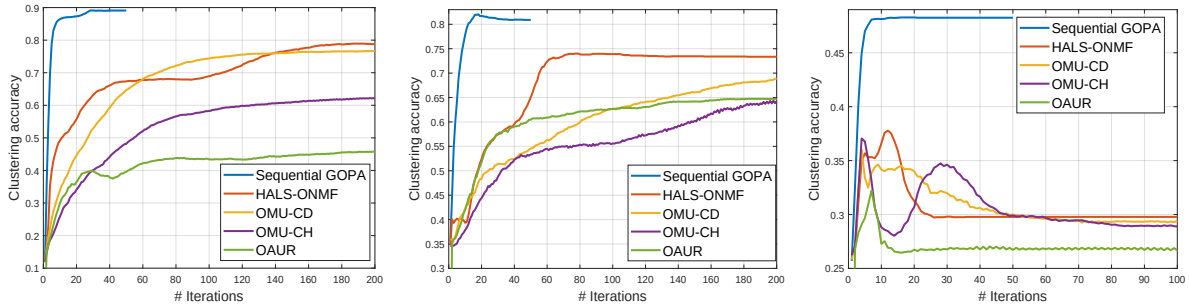
3.1. Clustering Tasks

We have selected altogether 13 benchmark data sets that have been widely used in evaluating the performance of clustering algorithms, including sparse text data, digital images, and medical data, with details listed in Table 2. We use random initialization for all the algorithms (for GOPA we use sparse random initialization namely each row of $\mathbf{W}$ only has one non-zero entry). To reduce statistical variations, we repeat each algorithm 30 times and report the mean and standard deviation of clustering accuracy. Results that are statistically significantly better than others are in bold letters. Both sequential and batch-mode GOPA methods use a updating ratio of 0.5.

As can be seen from Table 1, generally soft-constraints based orthogonal NMF produce better clustering results than those using multiplicative updates. We speculate this might be due to the zero-locking problem that is intrinsic to multiplicative updates. The proposed sequential and batch-mode GOPA methods demonstrate promising results on the majority of the data sets. Note that generally the batch-mode GOPA method is almost as good as the sequential counterpart, while on several text data sets the batch-mode GOPA shows notable improvement in accuracy (RCV1, Reuters, Newsgroup4). This should be attributed to the con-current updates of the non-zero entries of a number of rows of $\mathbf{W}$. Overall, our approaches produce the highest accuracy on 12 out of altogether 13 data sets. This provides empirical evidence that strong orthogonality benefits clustering.

3.2. Convergence Behaviour

In this section we further study the convergence behavior of different algorithms. First, it is worthwhile to note that a fair comparison of the objective function can be quite non-trivial for orthogonal NMF problems. This is because the



(a) UCI data (b) DNA data (c) Newsgroup4 Data

Figure 6. Convergence behaviour of different orthogonal NMF algorithms in terms of clustering accuracy. The GOPA method shows faster convergence and better stability during the iterations.

objective value $\|\mathbf{X} - \mathbf{WH}\|^2$ depends crucially on the sparsity level of solutions: a dense model ($\mathbf{W}$) with more non-zero entries can easily achieve a lower objective than a sparse model. While in practice, different orthogonal NMF algorithms can generate solutions whose level of orthogonality, and equivalently the sparsity, vary significantly. As a result, simply reporting the objective values does not faithfully mirror their performance and behaviours.

Given this consideration, we will adopt the clustering accuracy as a more informative metric for examination. In Figure 6, we report the clustering accuracy versus the number of iterations for different algorithms on a few representative benchmark data sets. We can observe that sequential GOPA converges very fast, typically in less than 30 iterations, while other methods may take one to two hundreds steps. Note that the objective value of GOPA is not the lowest, however, its clustering performance is significantly better than others. Therefore, the superior performance of GOPA can only be attributed to its strictly orthogonal solution. In comparison, all other competing algorithms of orthogonal NMF (except the clustering-based formulation (Pompili et al., 2014)) can only achieve partial orthogonality. This clearly demonstrates the importance of enforcing useful structural constraints in obtaining good clustering performance. Our results provide strong evidence that strict orthogonality can be more desirable than approximate one in clustering, which adds to our current understanding of NMF in solving clustering problems.

4. Conclusion and Future Work

In this paper, we proposed a greedy pivoting algorithm to obtain orthogonal solutions in non-negative matrix factorization. Our approach promotes exact orthogonality and can also benefit from flexible randomization and parallelization schemes. We provide solid empirical evidence that strict orthogonality in NMF improves the clustering performance on a wide collection of benchmark data sets.

There are a number of interesting directions currently being pursued. For example, we are interested in a rigorous theoretic analysis of the convergence rate of the sequential GOPA method, and probabilistic error bounds that can be used to quantify the behaviour of batch-mode GOPA. We are also studying new ways of decomposing the orthogonal NMF problem to improve optimality of solutions. Finally, we will study how to adaptively relax the orthogonality constraints (and so the sparsity of solutions) to solve pattern mining problems more robustly.

Acknowledgements

This work is supported in part by the National Science Foundation of China NO.61672236.

References

Asteris, M., Papailiopoulos, D., and Dimakis, A. G. Orthogonal nmf through subspace exploration. In *Advances in Neural Information Processing Systems 28*, pp. 343–351. 2015.

Beck, A. and Tetruashvill, L. On the convergence of block coordinate descent type methods. *SIAM Journal of Optimization*, 23(4):2037–2060, 2013.

Cai, D., He, X., Han, J., and Huang, T. S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

Choi, S. Algorithms for orthogonal nonnegative matrix factorization. In *IEEE International Joint Conference on Neural Networks*, 2008.

Cichocki, A. and Anh-Huy, P. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 92(3):708–721, 2009.

Ding, C., Li, T., Peng, W., and Park, H. Orthogonal nonnegative tri-matrix factorizations for clustering. In *SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 126–135, 2006.

Ding, C., Li, T., and Peng, W. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.

Donoho, D. and Stodden, V. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems 16*, pp. 1141–1148. 2004.

Gillis, N. Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13:3349–3386, 2012.

Hoyer, P. O. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research archive*, 5:1457–1469, 2004.

Hsieh, C.-J. and Dhillon, I. S. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *pInternational Conference on Knowledge Discovery and Data Engineering*, 2011.

Kim, J. and Park, H. Non-negative matrix factorization based on alternating non-negativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730, 2008a.

Kim, J. and Park, H. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 353–p362, 2008b.

Kim, J., He, Y., and Park, H. Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58:285–319, 2014.

Kimura, K., Tanaka, Y., and Kudo, M. A fast hierarchical alternating least squares algorithm for orthogonal nonnegative matrix factorization. In *Asian Conference on Machine Learning*, 2014.

Kuang, D., Ding, C., and Park, H. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international Conference on Data Mining*, 2012.

Lee, D. D. and Seung, S. H. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pp. 556–562, 2000.

Li, T. and Ding, C. The relationships among various nonnegative matrix factorization methods for clustering. In *Sixth International Conference on Data Mining*, 2006.

Lin, C.-J. Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19:2756–2779, 2007.

Ozerov, A. and Fevotte, C. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Neural Networksp*, 18(3):550–563, 2010.

Peharz, R. and Pernkopf, F. Sparse nonnegative matrix factorization with l0-constraints. *Neurocomputing*, 80(15): 38–46, 2012.

Pompili, F., Gillis, N., Absil, P.-A., and Glineurp, F. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, 141:15–25, 2014.

Shiga, M., Tatsumi, K., Muto, S., Tsuda, K., Yamamoto, Y., Mori, T., and Tanji, T. A convergent algorithm for orthogonal nonnegative matrix factorization. *Journal of Computational and Applied Mathematics*, 260:149–166, 2014.

Shiga, M., Tatsumi, K., Muto, S., Tsuda, K., Yamamoto, Y., Mori, T., and Tanji, T. Sparse modeling of eels and edx spectral imaging data by nonnegative matrix factorization. *Ultramicroscopy*, 170:43–59, 2016.

Tseng, P. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):457–494, 2001.

Wang, Y.-X. and Zhang, Y.-J. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 6:1336–1353, 2013.

Xu, W., Liu, L., and Gong, Y. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267–273, 2003.

Yang, Z. and Oja, E. Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 22(12):1878–1891, 2011.

Yoo, J. and Choi, S. Orthogonal nonnegative matrix factorization: Multiplicative updates on stiefel manifolds. *Intelligent Data Engineering and Automated Learning*, pp. 140–147, 2008.