NONNEGATIVE MATRIX FACTORIZATION WITH GRADIENT VERTEX PURSUIT

Dung N. Tran[†], *Tao Xiong*[†], *Sang Peter Chin*[†♯], *Trac D. Tran*[†]

[†]The Johns Hopkins University, Department of ECE, 3400 N. Charles St., Baltimore, MD 21218
[♯]Boston University, Dept. of CS, 111 Cummington Mall, Boston, MA 02215

ABSTRACT

Nonnegative Matrix Factorization (NMF), defined as factorizing a nonnegative matrix into two nonnegative factor matrices, is a particularly important problem in machine learning. Unfortunately, it is also ill-posed and NP-hard. We propose a fast, robust, and provably correct algorithm, namely *Gradient Vertex Pursuit* (GVP), for solving a well-defined instance of the problem which results in a unique solution: there exists a polytope, whose vertices consist of a few columns of the original matrix, covering the entire set of remaining columns. Our algorithm is greedy: it detects, at each iteration, a correct vertex until the entire polytope is identified. We evaluate the proposed algorithm on both synthetic and real hyperspectral data, and show its superior performance compared with other state-of-the-art greedy pursuit algorithms.

***Index Terms*—** Machine learning, nonnegative matrix factorization, greedy pursuit, Gradient Vertex Pursuit

1. INTRODUCTION

In the *Nonnegative Matrix Factorization* (NMF) problem [1], given an $n \times m$ nonnegative matrix Y and a positive integer $s < \min\{n, m\}$, one is asked to express Y as AX, where A and X are nonnegative matrices of size $n \times s$ and $s \times m$ respectively. This problem, though finds itself in enormous number of applications in various fields, is ill–posed [2] and NP–hard [3]. Most traditional methods rely on solving a non-convex optimization problem which lack of optimality guarantee [4]. Therefore, provable algorithms for computing NMF under appropriate assumptions are of particular interest.

Recently, it has been shown that under the separable assumption, the NMF problem admits a unique solution [2].

Definition 1 (Separable NMF)**.** *A data matrix Y is s–separable if there exists a cone generated by a few columns of Y that contains the entire dataset.*

In this paper, we tackle the problem of factorizing a nonnegative matrix under the convex hull assumption, a variant of the separibility, which, too, results in a unique factorization: given a data matrix, all of its columns reside in a convex hull generated by a small subset of a few columns of the matrix itself. This assumption was justified in several applications such as text mining, hyperspectral unmixing, and blind source separation [5] [6] [7].

1.1. Notation

We work in the usual Euclidean space $\mathbb{R}^n$ with the canonical basis $\{e_j\}_{j=1}^n$. For a set Γ, let $|\Gamma|$ denote its cardinality. Given a matrix A, the notations A_j and A_Γ represent its column j and a matrix extracted from $|\Gamma|$ columns of A with indices in Γ, respectively. The probability simplex in the subspace spanned by $\{e_j\}_{j\in\Gamma}$ in $\mathbb{R}^n$ is defined as $\Omega_\Gamma^n = \{x \in \mathbb{R}^n : x \geq 0, \|x\|_1 = 1, \text{and } x_k = 0 \text{ for all } k \notin \Gamma\}$. The norm $\|\cdot\|_{0,\text{row}}$ counts the number of the nonzero rows of a matrix. We write $\mathbf{1}$ for the all one vector and $\boldsymbol{I}$ for the identity matrix.

1.2. Problem statement

We now formally state our main assumption.

Assumption 2 (Convex hull assumption)**.** *Given a matrix $Y \in \mathbb{R}^{n\times m}$ and some positive integer $s < \min\{n, m\}$, there exists an index set $\mathcal{S}$ of cardinality s such that $Y = Y_\mathcal{S} X$ for some $X \in \mathbb{R}_+^{s\times m}$ satisfying $\mathbf{1}^T X = \mathbf{1}^T$.*

Solving the NMF problem under Assumption 2, called as separable NMF with sum–to–one constraint as well, can be reduced to identifying the vertices of the polytope, denoted by $\mathcal{C} = \text{conv}(\{Y_j\}_{j\in\mathcal{S}})$, covering the dataset. This is the approach we will take. Importantly, an algorithm correctly factorizes separable NMF with this assumption can exactly solve, by first normalizing columns of the input matrix to sum to one, the separable NMF problem without the sum–to–one constraint. Throughout of the paper, the notation $(Y; \mathcal{S}, \mathcal{C})$ refers to a data matrix Y satisfying assumption 2: the columns of Y are contained in polytope $\mathcal{C}$ with vertex index set $\mathcal{S}$.

The equation $Y = Y_\mathcal{S} X$ in Assumption 2 can then be rewritten as $Y = YX$ for some $X \in \mathbb{R}_+^{s\times m}$ such that $\mathbf{1}^T X = \mathbf{1}^T$ and X has at most s nonzero rows whose indices are contained in $\mathcal{S}$. We can thus formulate the NMF problem under

This work is partially supported by NSF-CCF-1117545, NSF-CCF-1422995, ARO-60219-MA, ONR-N00014-12-1-0765, NSF-DMS-1222567, and FA9550-12-1-0136.

978-1-4673-6997-8/15/$31.00 ©2015 IEEE

this assumption as a general self-dictionary learning problem:

$$\min \mathcal{L}(Y, YX) \text{ s.t. } \|X\|_{0,\text{row}} \leq s,\ X \geq 0,\ \mathbf{1}^T X = \mathbf{1}^T, \tag{1}$$

where $\mathcal{L}$ is a loss function which measures the consistency between the reconstructed matrix and the original one. We consider a class of loss functions satisfying the separability.

Assumption 3. *The loss function $\mathcal{L}$ is separable into the sum of functions of the individual columns of its argument; i.e., $\mathcal{L}(A) = \sum_j f(A_j)$ where $f : \mathbb{R}^n \to \mathbb{R}_+$ is a strongly convex function with parameter $\mu > 0$ and its gradient is Lipschitz continuous with constant L. That is, $L\boldsymbol{I} \prec \nabla^2 f \prec \mu\boldsymbol{I}$ where $\nabla^2 f$ is the Hessian of f. We further assume that $f(0) = 0$ if and only if $x = 0$.*

Therefore, (1) is equivalent to

$$\min \sum_j f(Y_j, YX_j) \text{ s.t. } \|X\|_{0,\text{row}} \leq s,\ X \geq 0,\ \mathbf{1}^T X = \mathbf{1}^T. \tag{2}$$

The reason for imposing these conditions on f will become clear in the journal version of the paper, where we analyze the robustness of the proposed algorithm. Furthermore, the strongly convexity of f allows fast implementation of the algorithm. To solve (1), we introduce a fast, robust, and provably correct greedy pursuit algorithm, *Gradient Vertex Pursuit* (GVP), based on these assumptions.

Several provable algorithms based on the separability assumption have been proposed in the last few years such as linear programming [8] [9] [10], convex optimization [11] [12], and geometric insights [13] [14] [15] [16] based methods. As we will show in the experiment section, GVP, while possessing the flexibility and low complexity of a typical greedy algorithm (and thereby faster than linear and convex optimization methods), is more robust than other greedy pursuit algorithms for solving the NMF problem under assumption 2.

2. THEORETICAL FOUNDATION

Our solution relies on a greedy approach: given the input data $(Y; \mathcal{S}, \mathcal{C})$, we maintain an estimate of $\mathcal{S}$ and incrementally augment this set one vertex at an iteration. This estimate at some iteration t is denoted by $\mathcal{S}^t$ and the convex hull generated by $\{Y_j\}_{j \in \mathcal{S}^t}$ is represented by $\mathcal{C}^t$. Furthermore, we call $\mathcal{C}^t$ the *sub-polytope* of $\mathcal{C}$ at iteration t. The basic intuition of our method stems from the following observation.

Claim 4. *If there is a point lying outside a sub–polytope $\mathcal{C}^t$ of the original vertex hull, there exists at least one vertex which does not belong to $\mathcal{C}^t$.*

The proof for this claim is trivially, thus can be omitted. Finding such a point is easy and can be done efficiently by solving the convex optimization problem:

$$\min \mathcal{L}(Y, Y_{\mathcal{S}^t} X) \quad \text{s.t.} \quad X \geq 0,\ \mathbf{1}^T X = \mathbf{1}^T. \tag{3}$$

It can be easily seen that solving (3) is equivalent to solving

$$\min \mathcal{L}(Y, YX) \quad \text{s.t.} \quad X_j \in \Omega^m_{\mathcal{S}^t}, \forall j \in \{1, ..., m\}. \tag{4}$$

Let X^t be the optimal solution to (4); each column of the matrix YX^t is the projection of the corresponding data point onto $\mathcal{C}^t$. Consequently, the zero columns of the residual matrix $R = Y - YX^t$ correspond to the interior points of the sub–polytope, whereas the residuals of the data points lying outside $\mathcal{C}^t$ are nonzero.

The main concern now is on a strategy for vertex identification given a sub–polytope and some exterior point. The following lemma suggests a way to proceed.

Lemma 5. *Let Y_l be a column lying outside a sub–polytope $\mathcal{C}^t$, and let X^t be the optimal solution to (4), then*

$$\frac{\partial f(Y_l, YX_l^t)}{\partial x_j} \geq \min_{k \in \mathcal{S}} \frac{\partial f(Y_l, YX_l^t)}{\partial x_k} \tag{5}$$

for any $j \in \{1, ..., m\}$.

This lemma can be proved by applying the chain rule to the left hand side of (5) and then utilizing Assumption 2. Lemma 5 is a generalization of a basic result in polyhedra theory: there exists at least one vertex that is an optimal solution to the problem of maximizing a linear function over a polytope [17]. Indeed, if f is chosen to be the l_2 loss, then (5) becomes

$$R_l^T Y_j \leq \max_{k \in \mathcal{S}} R_l^T Y_k \tag{6}$$

for all $j \in \{1, ..., m\}$. As a result of the lemma, a vertex can be identified by minimizing the left hand side of (5) over the whole data set; in fact, this is the *greedy selection criteria* that we will use in the algorithm. Importantly, it can be proved that none of the vertices of the sub–polytope is an optimal solution to this minimization problem.

Lemma 6. *Assuming Y_l is a column lying outside a sub–polytope $\mathcal{C}^t$ with X^t as the optimal solution to (4), the following holds:*

$$\frac{\partial f(Y_l, YX_l^t)}{\partial x_l} < \frac{\partial f(Y_l, YX_l^t)}{\partial x_k} \tag{7}$$

for any $k \in \mathcal{S}^t$.

The proof for this lemma starts from the optimality condition of (4). It can easily be obtained by combining a few simple facts and assumptions: $e_k \in \Omega^n_{\mathcal{S}^t}$ for any $k \in \mathcal{S}^t$, f is convex, $f(Y_l, Ye_l) = 0$, and $f(Y_l, YX_l^t) > 0$. A column whose index uniquely minimizes the left hand side of (5) is in fact a vertex that does not belong to $\mathcal{C}^t$. We conclude this section with our main theorem.

Theorem 7. *Given a sub-polytope $\mathcal{C}^t$ and an exterior point Y_l, and let X^t be the optimal solution to (4); if the optimization problem*

$$\min_{j \in \{1,...,m\}} \frac{\partial f(Y_l, YX_l^t)}{\partial x_j} \tag{8}$$

has a unique solution k^{t+1}, then $k^{t+1} \in \mathcal{S} \setminus \mathcal{S}^t$.

3. GRADIENT VERTEX PURSUIT ALGORITHM

The following algorithm naturally follows from the analysis in the previous section.

Algorithm 8 (**GVP**).

Input: *matrix* $Y \in \mathbb{R}^{n\times m}$ *satisfying Assumption 2, the number of vertices* s.
Output: *A set* $\tilde{\mathcal{S}}$ *of cardinality* s, *and a matrix* $\tilde{X} \in \mathbb{R}^{m\times m}$.
Procedure:

1. *Initialize the vertex set estimate* $\mathcal{S}^0 = \emptyset$, *the coefficient* $X^0 = 0$, *the residual* $R^0 = 0$, *and the iteration counter* $t = 0$.
2. *Arbitrarily choose a nonzero residual* R^t_l, *and find*
$$k^{t+1} = \underset{j\in\{1,\dots,m\}}{\operatorname{argmin}} \frac{\partial f(Y_l, YX^t_l)}{\partial x_j}$$
3. *Augment* $\mathcal{S}^{t+1} = \mathcal{S}^t \cup \{k^{t+1}\}$.
4. *Project* Y *onto* $\mathcal{C}^{t+1}$ *by finding* X^{t+1} *that solves:*
$$\min \mathcal{L}(Y, YX) \quad \textit{s.t.} \quad X_j \in \Omega^m_{\mathcal{S}^{t+1}}, \forall j \in \{1, ..., m\},$$
where $\Omega^m_{\mathcal{S}^{t+1}} = \{x \in \mathbb{R}^m : x \geq 0, \|x\|_1 = 1,$ *and* $x_k = 0$ *for all* $k \notin \mathcal{S}^{t+1}\}$.
5. *Compute the residual* $R^{t+1} = Y - YX^{t+1}$.
6. *Set* $t = t + 1$; *return to step 2 if* $t < s$, *otherwise, terminate the algorithm.*
7. *Set* $\tilde{\mathcal{S}} = \mathcal{S}^t$ *and* $\tilde{X} = X^t$.

Step 2 of the algorithm formalizes the greedy selection criteria mentioned in the previous section. The intuition behind it can be seen by letting f be the l_2 loss function. At initialization, this step finds the column with the largest l_2 norm; moreover, at next iterations, it identifies the column that most correlates to the chosen nonzero residual. As a result of Theorem 7, one of the vertices is selected at each iteration and none can be ever chosen twice. This result is stated in Theorem 9 whose proof can be obtained easily by applying Theorem 7.

Theorem 9 (Correctness of GVP)**.** *If the input data* $(Y; \mathcal{S}, \mathcal{C})$ *satisfies Assumption 2 and the optimization problem at step 2 at each iteration has a unique solution, the GVP algorithm correctly identifies all vertices after exactly* $|\mathcal{S}|$ *iterations.*

The solution of the optimization problem at step 4 in iteration t represents the coefficient of the data matrix when all data is projected onto the corresponding sub–polytope. This minimization problem is convex, thus can be solved efficiently by many off-the-shelf optimization solvers.

Remark. Similar to [13], the following action can be performed to deal with the situation when (8) has multiple different optimal solutions at certain iteration t. If the solutions are vertices, all of them are added to $\mathcal{S}^t$; otherwise, GVP can be called recursively to identify the vertices of this set of solutions and add them to $\mathcal{S}^t$. Strikingly, the unique solution assumption in Theorem 9 is not strict: it satisfies with a high probability when data is randomly distributed. The proof for this claim is nontrivial, thus beyond the scope of the paper.

Moreover, the performance of the algorithm is not affected by the presence of duplicate columns. If step 2 results in identical vertices, only one of them is added to the vertex set estimate. As shown in Theorem 7, none of them is selected during subsequent iterations.

It is important to note that our method is different from the XRAY algorithm [13] [14] as the latter fails to solve the NMF problem under Assumption 2. This can be seen by considering a counter example shown in Figure 1a. Furthermore, in many cases, although XRAY successfully identifies the polytope vertices, it fails when there is small perturbation in the data as shown in Figure 1b.

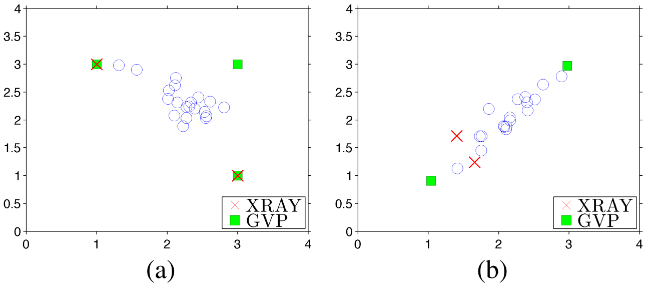


Fig. 1. Counter examples. (a) Data contained in a triangle with vertices $(3, 1)$, $(1, 3)$, and $(3, 3)$. (b) Noisy version of a dataset distributed on the line connecting vertices $(1.1, 1)$ and $(3, 2.9)$. XRAY fails in both cases, whereas GVP correctly identifies all vertices.

Computational complexity. The GVP algorithm requires $\mathcal{O}(nms)$ operations in total. A comparison of its complexity to several state-of-the-art algorithms is shown in Table 1. Here, c is the number of iterations performed in the ADMM algorithm for solving the ℓ_{12}-minimization problem [12].

Table 1. Computational complexity comparison.

VCA [18]	SPA [15]	XRAY [13]	GVP	$\ell_{1,2}$ [12]
$\mathcal{O}(nms)$	$\mathcal{O}(nms)$	$\mathcal{O}(nms)$	$\mathcal{O}(nms)$	$\mathcal{O}(cm^3)$

4. NUMERICAL EXPERIMENTS

This section evaluates the GVP algorithm on both synthetic and real data[1], and compare its performance to those of various greedy algorithms in Table 1. We also compare our algorithm to the $\ell_{1,2}$–minimization method [12] to show that GVP, while being greedy, has almost the same superior performance as this convex relaxation method. We use a similar experiment setup to [19] and let f to be the ℓ_2 loss in all experiments.

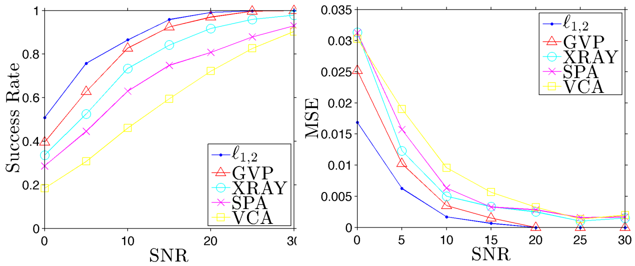
[1] We thank Qing Qu for his valuable suggestions.

Table 2. Running time comparison on synthetic data.

Algorithm	VCA	SPA	XRAY	GVP	$\ell_{1,2}$
Time	0.35	0.14	1.99	5.01	30.55

4.1. Synthetic data

We test the robustness of our proposed algorithm against noise on a USGS library [2]. For each simulation, the data is generated as follows. Each column of the vertex matrix $Y_{\mathcal{S}} \in \mathbb{R}^{n\times s}$ is randomly selected from the library; the coefficient matrix $X \in \mathbb{R}^{s\times m}$ has the form of $\Gamma[\boldsymbol{I_s}, X']$, where $\boldsymbol{I_s} \in \mathbb{R}^{s\times s}$ is the identity matrix, each column of $X' \in \mathbb{R}_+^{m\times(m-s)}$ follows from a Dirichlet distribution whose parameters are chosen from a uniform distribution on $[0,1]$ and Γ is a permutation matrix. The data matrix is generated by $Y = Y_{\mathcal{S}}X + N$ where each element of N is drawn from a Normal distribution. The signal–to–noise ratio (SNR), defined as $\text{SNR} = 10\log_{10}\left(\frac{1}{ns}\frac{\|Y\|_F}{\|N\|_F}\right)$, is varied from 0 to 30 dB. For each SNR level, the simulation is repeated 100 times. The success rate and mean square error are shown in Figure 2. We can see that the GVP algorithm outperforms other greedy algorithms. Moreover, it is approximately 6 times faster than the $\ell_{1,2}$–minimization as shown in Table 2.



(a) Success Rate

(b) Mean Square Error

Fig. 2. Robustness comparison on synthetic data.

4.2. Hyperspectral unmixing

This subsection presents numerical results of the GVP algorithm when applied to the hyperspectral unmixing problem. We use the Urban data [3] in our experiments. This data is mainly constituted of six types of materials including road, roof, metal, dirt, grass, and tree. Additionally, the dimension of the preprocessed data cube is 307 × 307 × 162 [19]. The parameters of the original data matrix $\tilde{Y} \in \mathbb{R}^{n\times m}$ are thus given by: the signal dimension $n = 162$, the total number of data points $m' = 307 \times 307 = 94249$, and the number of endmembers $s = 6$. We reduced the size of the dataset by merging similar columns, resulting in a reduced data matrix $Y \in \mathbb{R}^{n\times m}$ of size 162×1147. Algorithms are then applied to this reduced dataset to extract $s = 6$ endmembers. Figure 3 illustrates the vertices identified by various methods. Our GVP algorithm extracts distinct endmembers which are mostly similar to ones manually labeled.

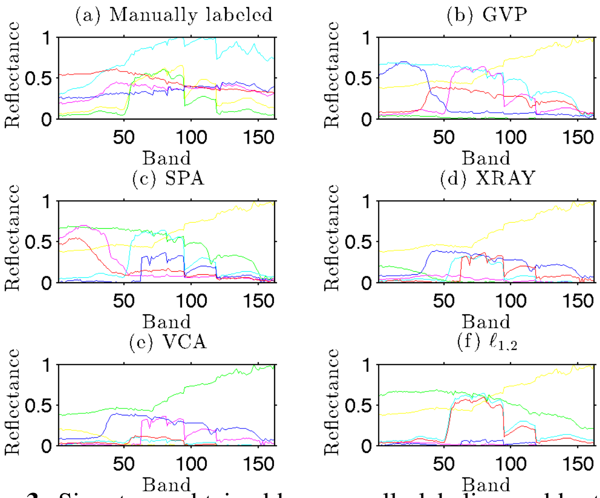


Fig. 3. Signatures obtained by manually labeling and by the algorithms.

To further compare the performance of the algorithms, we use the root mean square error $\text{RMSE} = \frac{1}{\sqrt{ns}}\|\tilde{Y} - Y_{\mathcal{S}}X\|_F$. Here, $\mathcal{S}$ is the endmember index set extracted from the reduced data matrix Y by the algorithms, and X is the coefficient of the original data $\tilde{Y}$ when projected onto the estimated polytope, i.e., $X = \text{argmin}\ \|\tilde{Y} - Y_{\mathcal{S}}X\|_F$ s.t. $X \geq 0$ and $\mathbf{1}^T X = \mathbf{1}^T$. This quantity measures the quality of the approximation: a small value of RMSE indicates that the detected polytope covers the entire data set well. Results for the Urban data set is shown in Table 3. It can be seen that the computationally-efficient GVP algorithm outperforms other greedy algorithms by a significant margin. In fact, its performance approaches that of $\ell_{1,2}$–minimization, while GVP is approximately 100 times faster than this convex method.

Table 3. RMSE and running time comparison on Urban data.

Algorithm	VCA	SPA	XRAY	GVP	$\ell_{1,2}$
RMSE	93.20	37.88	54.02	26.27	25.28
Time	0.11	0.098	0.24	2.23	199.71

5. CONCLUSIONS

This paper presents the GVP algorithm for solving the separable NMF problem with the sum–to–one constraint. GVP is fast, robust, and provably correct. Its robustness to noise, as shown empirically in this paper, will be addressed and rigorously proved in the follow-up journal version of this paper. Finally, its quality of the factorization and approximation will be further investigated by applying the method to various other applications.

[2] http://www.lx.it.pt/ bioucas/code/sunsal demo.zip

[3] http://www.agc.army.mil/

6. REFERENCES

[1] D. Lee and S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.

[2] David Donoho and Victoria Stodden, “When does non-negative matrix factorization give correct decomposition into parts?,” in *NIPS*. 2003, MIT Press.

[3] S. Vavasis, “On the complexity of nonnegative matrix factorization,” *SIAM J. on Optimization*, vol. 20, pp. 1364–1377, 2009.

[4] Daniel D. Lee and H. Sebastian Seung, “Algorithms for non-negative matrix factorization,” in *NIPS*. 2000, pp. 556–562, MIT Press.

[5] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu, “A practical algorithm for topic modeling with provable guarantees,” in *ICML*, 2009, vol. 28, pp. 280–288.

[6] T.H. Chan, W.K. Ma, C.Y. Chi, and Y. Wang, “A convex analysis framework for blind separation of non-negative sources,” *IEEE Trans. on Signal Processing*, vol. 56, pp. 5120–5134, 2008.

[7] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, pp. 354–379, 2012.

[8] S. Arora, R. Ge, R. Kannan, and A. Moitra, “Computing a nonnegative matrix factorization provably,” in *Proceedings of the 44th symposium on Theory of Computing*, 2012, pp. 145–162.

[9] V. Bittorf, B. Recht, C. Re, and J.A. Tropp, “Factoring nonnegative matrices with linear programs,” in *NIPS*, 2012, pp. 1223–1231.

[10] N. Gillis and R. Luce, “Robust near-separable nonnegative matrix factorization using linear optimization,” *Journal of Machine Learning Research*, vol. 15, pp. 1249–1280, Apr 2014.

[11] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, “A convex model for nonnegative matrix factorization and dimensionality reduction on physical space,” *IEEE Transactions on Image Processing*, vol. 21, pp. 3239–3252, 2012.

[12] E. Elhamifar, G. Sapiro, and R. Vidal, “See all by looking at a few: Sparse modeling for finding representative objects,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[13] A. Kumar, V. Sindhwani, and P. Kambadur, “Fast conical hull algorithms for near-separable non-negative matrix factorization,” in *ICML*, 2012, vol. 28, pp. 231–239.

[14] A. Kumar and V. Sindhwani, “Near-separable non-negative matrix factorization with ℓ_1 and bregman loss functions,” in *arXiv:1312.7167*, 2013.

[15] N. Gillis and S.A. Vavasis, “Fast and robust recursive algorithms for separable nonnegative matrix factorization,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 698–714, 2014.

[16] N. Gillis, “Successive nonnegative projection algorithm for robust nonnegative blind source separation,” *SIAM J. on Imaging Sciences*, vol. 7, pp. 1420–1450, 2014.

[17] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver, “Combinatorial optimization,” in *Proceedings Title*. John Wiley & Sons, Inc., 1998.

[18] Jose M. P. Nascimento and Jose M. B. Dias, “Vertex component analysis: A fast algorithm to unmix hyperspectral data,” *IEEE TRANS. GEOSCI. REM. SENS*, vol. 43, pp. 898–910, 2004.

[19] Q. Qu, X. Sun, N.M. Nasrabadi, and T.D. Tran, “Subspace vertex pursuit for separable non-negative matrix factorization in hyperspectral unmixing,” in *IEEE International Conference onAcoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 8115–8119.