Frank-Wolfe Algorithm for Simplicial and Nonnegative Component Analysis

Jingzhou Hu and Kejun Huang
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, Florida 32611
Email: (jingzhou, kejun.huang)@ufl.edu

2024 IEEE 13th Sensor Array and Multichannel Signal Processing Workshop (SAM) | 979-8-3503-4481-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/SAM60225.2024.10636440

***Abstract*—In this paper, we propose to solve simplicial and nonnegative component analysis problems using the Frank-Wolfe algorithm. Simplicial component analysis (SCA) is a blind source separation technique that is widely used in hyperspectral unmixing, which has an interesting geometric interpretation of finding the minimum volume enclosing simplex of a set of data points. A highly related but different problem is called nonnegative component analysis (NCA), which is a variant of the celebrated nonnegative matrix factorization. Both problems enjoy amiable identifiability guarantees under a mild assumption that the factor matrix with simplicial and nonnegative structures satisfies the sufficiently scattered condition in the probability simplex or the nonnegative orthant, respectively. Algorithm design for either formulations remains to be challenging. In this paper we propose to use the Frank-Wolfe algorithm to solve it. After a brief review of the Frank-Wolfe algorithm, we first provide an improved convergence analysis of it by employing the backtracking line search strategy for choosing step sizes, and show that for convex problems it converges to a global optimum at a linear rate if the step sizes can be lowerbounded. Notice that no strong convexity is necessary to achieve such linear rate, which is somewhat surprising. Despite the improved convergence rate analysis, the proof is surprisingly succinct and easy to understand. Then we show how it can be applied to SCA and NCA with similar implementations. We conclude the paper by showing its performance in numerical experiments comparing with some baseline algorithms.**

***Index Terms*—Frank-Wolfe, minimum volume, nonnegative matrix factorization, simplicial component analysis**

I. Introduction

Nonnegative matrix factorization (NMF) has been a powerful tool for signal and data analytics [1], particularly thanks to its identifiability property under the mild sufficiently scattered condition [2]. If identifiability is the main concern, people have discovered more principled formulation to help facilitate it, using a so-called minimum volume criterion. The idea stems from hyperspectral unmixing, which takes a similar matrix factorization model $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$, but assumes that $\boldsymbol{S}$ is not only elementwise nonnegative but also column sum to one, meaning that each column of $\boldsymbol{X}$, representing pixels of the hyperspectral image, is a *convex* combination of the columns of $\boldsymbol{A}$. To promote identifiability, the following formulation is proposed [3], [4],

Supported in part by NSF ECCS-2237640 and NIH R01LM014027.

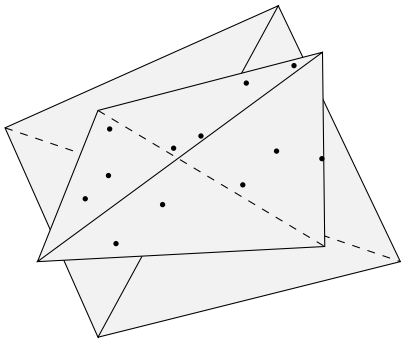

Fig. 1. A geometric interpretation of (1) of finding the minimum volume enclosing simplex for a set of points.

which we call *simplicial component analysis* (SCA) in this paper:

$$\begin{aligned} \underset{\boldsymbol{A},\boldsymbol{S}}{\text{minimize}} \quad & \det(\boldsymbol{A}^\top\boldsymbol{A} + \boldsymbol{1}\boldsymbol{1}^\top) \\ \text{subject to} \quad & \boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}, \quad \boldsymbol{S} \geq 0, \quad \boldsymbol{1}^\top\boldsymbol{S} = \boldsymbol{1}^\top. \end{aligned} \tag{1}$$

Formulation (1) has a very interesting geometric interpretation. Since each column of $\boldsymbol{S}$ is a nonnegative vector that sums to one, it means that every column of $\boldsymbol{X}$ is a convex combination of the columns of $\boldsymbol{A}$; in other words, the convex hull of the columns of $\boldsymbol{A}$ is a polytope that encloses the set of points defined by the columns of $\boldsymbol{X}$. Furthermore, if columns of $\boldsymbol{A}$ are affinely independent, or equivalently if the columns of the following matrix

$$\widetilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{1}^\top \\ \boldsymbol{A} \end{bmatrix}$$

are linearly independent, then the polytope is called a simplex. When $\widetilde{\boldsymbol{A}}$ is a square matrix, then the volume of the simplex is proportional to $|\det\widetilde{\boldsymbol{A}}|$; if $\widetilde{\boldsymbol{A}}$ is tall then we can calculate the (degenerate) volume $\det\widetilde{\boldsymbol{A}}^\top\widetilde{\boldsymbol{A}}$, which equals to the objective function of (1). As a result, formulation (1) has a nice geometric interpretation of finding the minimum volume enclosing simplex (MVES) of a set of points. Figure 1 shows an illustration of MVES.

In the context of matrix factorization, however, the column sum to one constraint in (1) is somewhat unnatural. Due to scaling ambiguity, it is more natural to constraint, without loss of generality, that the rows of $\boldsymbol{S}$ are bounded, leading to the following formulation that is much more related to the

celebrated NMF. Since it is not the same as the most common formulation for NMF, and that the factor $\boldsymbol{A}$ is not required to be strictly nonnegative, we call it *nonnegative component analysis* (NCA) in this paper:

$$\begin{aligned} \underset{\boldsymbol{A},\boldsymbol{S}}{\text{minimize}} \quad & \det(\boldsymbol{A}^\top \boldsymbol{A}) \\ \text{subject to} \quad & \boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}, \quad \boldsymbol{S} \geq 0, \quad \boldsymbol{S}\boldsymbol{1} = \boldsymbol{1}. \end{aligned} \tag{2}$$

As we can see, this formulation not only imposes a more natural row-sum-to-one constraint on $\boldsymbol{S}$, but also involves a simpler objective function to optimize. Nevertheless, it is evident that formulations (1) and (2) show great similarities, and one would like to design a unified algorithmic framework to solve both of them.

One of the biggest benefit of using either formulation (1) or (2) in practice is their potential of identifying the correct latent factors under mild conditions. In a nutshell, if the groundtruth factor $\boldsymbol{S}$ satisfies a so-called *sufficiently scattered* condition in either the probability simplex for (1) or in the nonnegative orthant for (2) and $\boldsymbol{A}$ has independent columns, then the solutions of (1) or (2) must be the groundtruth $\boldsymbol{S}$ up to row permutation and/or positive scaling [5]–[8]. Pertinent formulations have found numerous applications in machine learning and signal processing, including hyperspectral unmixing [9], topic modeling [10], hidden Markov models identification [11], community detection [12], and crowdsourcing [13], to name just a few.

The main goal of this paper is to design a unified algorithmic framework for solving both (1) and (2). Before this work, people have developed algorithms based on block coordinate descent (BCD) [10], [14], and more specifically the augmented Lagrangian method for SCA [15]. One of the main drawbacks of BCD is that it only works when $\boldsymbol{A}$ is square, so one needs to employ an additional step of dimensionality reduction for the over-determined case. Furthermore, BCD works the best if the constraints are separable over the blocks, which is true for NCA but not SCA, despite their similar formulations. In this paper, we propose a more unified algorithmic framework based on the Frank-Wolfe algorithm. We introduce the Frank-Wolfe (FW) algorithm in Section II, with a side contribution of showing an improved convergence analysis with backtracking line search for convex problems. Then we describe how to apply FW for SCA and NCA in Section III. We will see how FW provides a more unified treatment to the two similar-looking but different formulations (1) and (2). Numerical results show improved performance on either of the formulations.

II. The Frank-Wolfe algorithm

The Frank-Wolfe (FW) algorithm [16], also known as the conditional gradient method for constrained optimization [17], iteratively minimizes a linear objective, defined by the gradient at the current iterate, under the same constraint set to determine the search direction and obtain the next iterate via some line search approach along the search direction. More specifically, consider the following generic optimization problem

$$\underset{\boldsymbol{w}}{\text{minimize}} \;\; f(\boldsymbol{w}) \quad \text{subject to} \;\; \boldsymbol{w} \in C, \tag{3}$$

where C is a convex constraint set. The Frank-Wolfe algorithm takes the following iterative form:

$$\begin{cases} \boldsymbol{g}_t & \leftarrow \arg\min_{\boldsymbol{w}_t + \boldsymbol{g} \in C} \nabla f(\boldsymbol{w}_t)^\top \boldsymbol{g} \\ \boldsymbol{w}_{t+1} & \leftarrow \boldsymbol{w}_t + \gamma_t \boldsymbol{g}_t \;\; \text{via line search} \end{cases}. \tag{4}$$

Notice that the line search step is open for the user to choose. Some common choices include constant, diminishing, exact, and Armijo, also known as back-tracking [18]. In this paper we focus on the back-tracking line search strategy, which seeks for a step size γ_t such that the following inequality holds:

$$f(\boldsymbol{w}_t + \gamma_t \boldsymbol{g}_t) \leq f(\boldsymbol{w}_t) + \beta \gamma_t \nabla f(\boldsymbol{w}_t)^\top \boldsymbol{g}_t, \tag{5}$$

where $0 < \beta < 1$ is some user-specified parameter.

FW has been a popular algorithm for many machine learning applications [19], but our understanding of its convergence behavior has been surprisingly limited. The seminal work [19] showed that for convex problems it converges to a global minimum at a sublinear rate when using a diminishing step size strategy. In this paper, we provide a surprisingly simple proof to show that if the backtracking line search is adopted, then the convergence can be easily improved to a linear rate under mild conditions.

Proposition 1. *Suppose f is a convex function and C is a convex set. Let $\boldsymbol{w}_\star$ denote an optimal solution of* (3)*, then the optimality gap at iteration t of the Frank-Wolfe algorithm* (4) *with back-tracking line search satisfies:*

$$f(\boldsymbol{w}_t) - f(\boldsymbol{w}_\star) \leq (f(\boldsymbol{w}_0) - f(\boldsymbol{w}_\star)) \prod_{s=0}^{t-1} (1 - \beta\gamma_s). \tag{6}$$

Proof. Since the back-tracking line search is employed, we have

$$f(\boldsymbol{w}_t) \leq f(\boldsymbol{w}_{t-1}) + \beta \nabla f(\boldsymbol{w}_{t-1})(\boldsymbol{w}_t - \boldsymbol{w}_{t-1}), \tag{7}$$

according to (5). By definition of the Frank-Wolfe search direction, we have

$$\nabla f(\boldsymbol{w}_{t-1})^\top \boldsymbol{g}_{t-1} \leq \nabla f(\boldsymbol{w}_{t-1})^\top (\boldsymbol{w} - \boldsymbol{w}_{t-1})$$

for all $\boldsymbol{w} \in C$. Instantiating $\boldsymbol{w} = \boldsymbol{w}_\star$ on the right-hand-side, and recognizing that $\boldsymbol{g}_{t-1} = (\boldsymbol{w}_t - \boldsymbol{w}_{t-1})/\gamma_t$ on the left-hand-side, this leads to

$$\frac{1}{\gamma_{t-1}} \nabla f(\boldsymbol{w}_{t-1})^\top (\boldsymbol{w}_t - \boldsymbol{w}_{t-1}) \leq \nabla f(\boldsymbol{w}_{t-1})^\top (\boldsymbol{w}_\star - \boldsymbol{w}_{t-1}). \tag{8}$$

Combining (8) with (7) gives

$$f(\boldsymbol{w}_t) \leq f(\boldsymbol{w}_{t-1}) + \beta\gamma_{t-1} \nabla f(\boldsymbol{w}_{t-1})(\boldsymbol{w}_\star - \boldsymbol{w}_{t-1}). \tag{9}$$

Now if f is convex and differentiable, then we also have

$$f(\boldsymbol{w}_\star) \geq f(\boldsymbol{w}_{t-1}) + \nabla f(\boldsymbol{w}_{t-1})^\top (\boldsymbol{w}_\star - \boldsymbol{w}_{t-1}). \tag{10}$$

Combining (10) with (9) gives us

$$f(\boldsymbol{w}_t) \leq f(\boldsymbol{w}_{t-1}) + \beta\gamma_{t-1} \left(f(\boldsymbol{w}_\star) - f(\boldsymbol{w}_{t-1})\right).$$

Subtracting $f(\boldsymbol{w}_\star)$ from both sides and rearrange, we get

$$f(\boldsymbol{w}_t) - f(\boldsymbol{w}_\star) \leq (1 - \beta\gamma_{t-1}) \left(f(\boldsymbol{w}_{t-1} - f(\boldsymbol{w}_\star))\right). \tag{11}$$

Repeating this inequality from iteration 0 to t then shows (6). □

The bound in (6) of Proposition 1 shows that the optimality gap at iteration t is a fraction of the initial optimality at initialization, and the factor involves all the past step sizes $\gamma_0, \ldots, \gamma_{t-1}$. Since $0 < \beta < 1$ and $0 < \gamma_s \leq 1$ for all s, that factor is in $(0, 1)$ for sure. If we further have that the step sizes have a common lowerbound $\gamma < \gamma_t$ for all t, then we easily obtain a linear convergence rate

$$f(\boldsymbol{w}_t) - f(\boldsymbol{w}_\star) \leq (1 - \beta\gamma)^t \left(f(\boldsymbol{w}_0) - f(\boldsymbol{w}_\star)\right).$$

This can be easily achieve by, for example, further assume that gradients of f are Lipschitz continuous. Proposition 1 also shows that the best rate is potentially achieved with $\gamma_t = 1$ for all t, which is still a linear rate $(1 - \beta)^t$.

Theorem 1. *Suppose f is a convex function and C is a convex set. Also assume there exists a constant L such that $\|\nabla f(\boldsymbol{w}) - \nabla f(\tilde{\boldsymbol{w}})\| \leq L\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|$ for all $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$. Let $\boldsymbol{w}_\star$ denote an optimal solution of* (3)*, then the optimality gap at iteration t of the Frank-Wolfe algorithm* (4) *with back-tracking line search satisfies:*

$$f(\boldsymbol{w}_t) - f(\boldsymbol{w}_\star) \leq (1 - \beta\gamma)^t \left(f(\boldsymbol{w}_0) - f(\boldsymbol{w}_\star)\right), \qquad (12)$$

where $\gamma \leq \gamma_t$ for all t.

We skip the proof due to space limitation.

III. Frank-Wolfe for SCA and NCA

Before we apply FW for SCA or NCA, we first reformulate the problems by eliminating the $\boldsymbol{S}$ variable. Let us start with the relatively simpler formulation (2). Let $\boldsymbol{A}$ have independent columns, then it has a left inverse $\boldsymbol{W}$, and $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$ is equivalent to $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X}$. We can therefore eliminate the variable $\boldsymbol{S}$ and apply a change of variable from $\boldsymbol{A}$ to $\boldsymbol{W}$ as

$$\begin{aligned} \underset{\boldsymbol{W}}{\text{minimize}} \quad & -\log\det \boldsymbol{W}\boldsymbol{W}^\top \\ \text{subject to} \quad & \boldsymbol{W}\boldsymbol{X} \geq 0, \quad \boldsymbol{W}\boldsymbol{X}\boldsymbol{1} = \boldsymbol{1}. \end{aligned} \qquad (13)$$

This is a nonconvex optimization problem but subject to convex (linear) constraints.

As for formulation 1, recall we defined $\widetilde{\boldsymbol{A}}$ by concatenating an all-one row on top of $\boldsymbol{A}$, and we can similarly define $\widetilde{\boldsymbol{X}}$ by concatenating an all-one row on top of $\boldsymbol{X}$ to combine the two equality constraints of (1) into one

$$\widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{A}}\boldsymbol{S}.$$

If we instead treat $\widetilde{\boldsymbol{A}}$ as variable, we need to constrain the first row of $\widetilde{\boldsymbol{A}}$ to be all ones, i.e., $\boldsymbol{e}_1^\top\widetilde{\boldsymbol{A}} = \boldsymbol{1}^\top$. Now assume columns of $\widetilde{\boldsymbol{A}}$ are linearly independent, it has a left inverse $\boldsymbol{W}$, then apply a change-of-variable while eliminating $\boldsymbol{S}$ leads to

$$\begin{aligned} \underset{\boldsymbol{W}}{\text{minimize}} \quad & -\log\det \boldsymbol{W}\boldsymbol{W}^\top \\ \text{subject to} \quad & \boldsymbol{W}\widetilde{\boldsymbol{X}} \geq 0, \quad \boldsymbol{1}^\top\boldsymbol{W} = \boldsymbol{e}_1^\top. \end{aligned} \qquad (14)$$

This is again a nonconvex optimization problem but subject to convex (linear) constraints. From an optimization perspective, the main difference between (14) and (13) is that the constraints of (13) separates over the rows of $\boldsymbol{W}$, while those of (14) are not. This is the reason why applying block coordinate descent works much better for (13) than (14).

Both (13) and (14) are easily amendable for applying the Frank-Wolfe algorithm. For the log-determinant objective, we have that the gradient is $-(\boldsymbol{W}^\dagger)^\top$. As a result, the Frank-Wolfe algorithm for (13) or (14) is given in Algorithm 1.

Algorithm 1 Solving (13) or (14) with Frank-Wolfe

initialize $\boldsymbol{W}_{(0)}$
for $t = 0, 1, 2, \ldots$ until convergence **do**
  $\boldsymbol{W}_d = \arg\min_{\boldsymbol{W}} -\operatorname{Tr}(\boldsymbol{W}_{(t)}^\dagger\boldsymbol{W})$
    subject to $\boldsymbol{W}\boldsymbol{X} \geq 0, \quad \boldsymbol{W}\boldsymbol{X}\boldsymbol{1} = \boldsymbol{1}$ if solving (13)
      or $\boldsymbol{W}\widetilde{\boldsymbol{X}} \geq 0, \quad \boldsymbol{1}^\top\boldsymbol{W} = \boldsymbol{e}_1^\top$ if solving (14)
  $\gamma \leftarrow 1$
  while $-\log|\det(\boldsymbol{W}_{(t)} + \gamma_t(\boldsymbol{W}_d - \boldsymbol{W}_{(t)}))| >$
      $-\log|\det \boldsymbol{W}_{(t)}| + (\gamma/2)\operatorname{Tr}(\boldsymbol{W}_{(t)}^\dagger(\boldsymbol{W}_d - \boldsymbol{W}_{(t)}))$
  do
    $\gamma \leftarrow \gamma/2$
  end while
  $\boldsymbol{W}_{(t+1)} = \boldsymbol{W}_{(t)} + \gamma(\boldsymbol{W}_d - \boldsymbol{W}_{(t)})$
end for

Regarding the line search step, we propose to use the backtracking line search (Armijo rule) [17] to guarantee sufficient decrease of the objective function. Since the constraint set of (13) or (14) is convex, as long as $\boldsymbol{W}_{(t)}$ is feasible, then $\boldsymbol{W}_{(t+1)}$ is also feasible since it is a convex combination of $\boldsymbol{W}_{(t)}$ and $\boldsymbol{W}_d$, which are by definition feasible.

Since we are trying to solve nonconvex problems, it is expected that the performance depends on the initialization. In our experience, the behavior of the two problems (13) or (14) are quite different. It is relatively easy to achieve good performance with NCA (13), as we can simply optimize an arbitrary linear objective subject to the same constraint as (13) and use the result as initialization $\boldsymbol{W}_{(0)}$. For SCA (14), such an initialization often fails to achieve good result. We propose to use the successive projection algorithm (SPA) to initialize Algorithm 1 when trying to solve SCA.

In terms of complexity, each iteration is dominated by the linear programming with mk variables if $\boldsymbol{A} \in \mathbb{R}^{m\times k}$. For the SCA problem (14), the per-iteration complexity could be as high as $O(m^3k^3)$. However, the linear programming to be solved in the NCA problem (13) is blessed with structures to be exploited to greatly reduce the complexity. Denote $\boldsymbol{w}_i$ as the ith row of $\boldsymbol{W}$, then the linear programming in each iteration of Algorithm 1 is in fact k independent problems, each involving only one row of $\boldsymbol{W}$; let $\boldsymbol{f}_i$ denote the ith column of $\boldsymbol{W}_{(t)}^\dagger$, then we should solve the following problem with $i = 1, \ldots, k$

$$\begin{aligned} \underset{\boldsymbol{w}_i}{\text{minimize}} \quad & -\boldsymbol{f}_i^\top\boldsymbol{w}_i \\ \text{subject to} \quad & \boldsymbol{w}_i^\top\boldsymbol{X} \geq 0, \quad \boldsymbol{w}_i^\top\boldsymbol{X}\boldsymbol{1} = 1 \end{aligned} \qquad (15)$$

Each of these problems involves m variables, which can be solved with $O(m^3)$ flops. This important observation brings

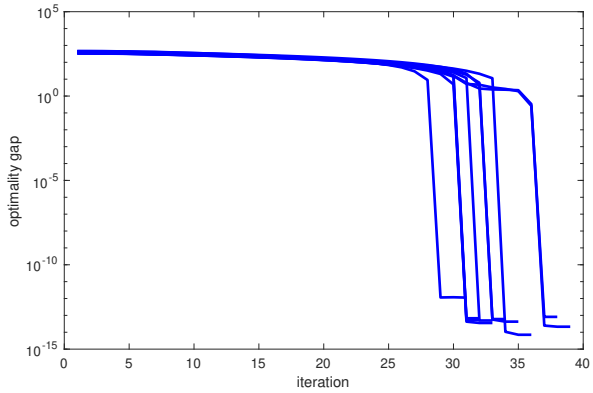

Fig. 2. Convergence of Algorithm 1 for the NCA problem (13) on 10 random trials.

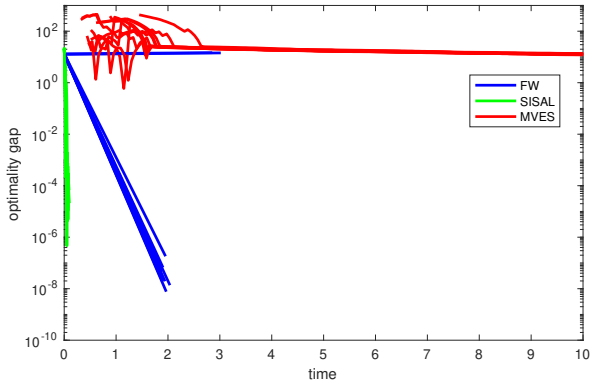

Fig. 3. Convergence of Algorithm 1 for the SCA problem (14) on 10 random trials.

the per-iteration complexity of Algorithm 1 down to $O(km^3)$ when solving the NCA problem (13).

IV. Numerical Validation

We conclude the paper by providing some numerical validation to the proposed theoretical analysis. Since SCA and NCA both guarantee identifiability under the sufficiently scattered condition, we are going to generate the data matrix $\boldsymbol{X}$ with a groundtruth latent factorization $\boldsymbol{A}_\natural\boldsymbol{S}_\natural$. This way, even though we are trying to solve a nonconvex problem either (13) or (14), we know the optimal value is $\log\det \boldsymbol{A}^\top\boldsymbol{A}$, and we can use it to measure the optimality gap, which ideally would go to zero if the proposed FW algorithm manages to solve (13) or (14) to global optimality.

We implement the algorithm in MATLAB and use the built-in `linprog` function in MATLAB to solve each of the linear programming sub-problems.

A. Performance on NCA

We fix $n = 1000$ and $m = k = 20$. Matrix $\boldsymbol{A}_\natural$ is simply generated from i.i.d. standard normal distribution. The $\boldsymbol{S}_\natural$ factor is first generated from i.i.d. exponential distribution, then approximately 50% of entries are randomly selected to be zeros. It has been empirically observed that such a randomly generated matrix $\boldsymbol{S}_\natural$ has a very high probability of satisfying the sufficiently scattered condition in the nonnegative orthant [1]. The data matrix $\boldsymbol{X} = \boldsymbol{A}_\natural\boldsymbol{S}_\natural$, and is feed into Algorithm 1 to generate the results. Figure 2 shows the convergence of Algorithm 1 for the NCA problem (13) on 10 random trials. As we can see, in all 10 instances a global optimum is attained within approximately 40 iterations. The surprising effectiveness is well-worth further investigation.

B. Performance on SCA

We fix $n = 1000$ and $m = k = 20$. Matrix $\boldsymbol{A}_\natural$ is simply generated from i.i.d. standard normal distribution. The $\boldsymbol{S}_\natural$ factor is first generated from i.i.d. exponential distribution, then approximately 50% of entries are randomly selected to be zeros, and each column of $\boldsymbol{S}$ is rescaled to sum to one. It has been empirically observed that such a randomly generated matrix $\boldsymbol{S}_\natural$ has a very high probability of satisfying the sufficiently scattered condition in the probability simplex [1]. The data matrix $\boldsymbol{X} = \boldsymbol{A}_\natural\boldsymbol{S}_\natural$, and is feed into Algorithm 1 to generate the results. Figure 3 shows the convergence of Algorithm 1 for the SCA problem (14) on 10 random trials. As we can see, 9 out of 10 instances a global optimum is attained in exactly one iteration. As we explained before, FW for SCA is initialized by SPA, which explains why it converges in just one iteration if it works. On the other hand, it is more likely to converge to a saddle point in the SCA case.

The proposed FW algorithm is compared with two classical algorithms for SCA: MVES based on the block coordinate descent (BCD) algorithm proposed in [14] and the simplex identification via split augmented Lagrangian (SISAL) method in [15]. With the problem dimension of this size, MVES is not performing very well, but SISAL works very well and converges much faster than FW, since it does not rely on existing convex optimization solvers.

V. Conclusion

In this paper we consider the simplicial and nonnegative component analysis problems, which are tightly related to the celebrated nonnegative matrix factorization and many of its applications such as hyperspectral unmixing, topic modeling, and community detection. We propose to solve both problems using a unified algorithmic framework based on the Frank-Wolfe algorithm. As a side contribution, we also provide an improved convergence analysis of Frank-Wolfe with backtracking line search on convex problems, and show that linear convergence can be easily obtained if there is a lowerbound on the step sizes obtained from the backtracking line search. Numerical experiments show impressive performance of the proposed algorithm.

References

[1] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and

applications," *IEEE Signal Processing Magazine*, vol. 36, no. 2, pp. 59–80, 2019.
[2] K. Huang, N. D. Sidiropoulos, and A. Swami, "Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 1, pp. 211–224, 2013.
[3] M. D. Craig, "Minimum-volume transforms for remotely sensed data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 3, pp. 542–552, 1994.
[4] J. Li and J. M. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, vol. 3. IEEE, 2008, pp. III–250.
[5] X. Fu, W.-K. Ma, K. Huang, and N. D. Sidiropoulos, "Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2306–2320, 2015.
[6] C.-H. Lin, W.-K. Ma, W.-C. Li, C.-Y. Chi, and A. Ambikapathi, "Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 10, pp. 5530–5546, 2015.
[7] K. Huang, N. D. Sidiropoulos, E. E. Papalexakis, C. Faloutsos, P. P. Talukdar, and T. M. Mitchell, "Principled neuro-functional connectivity discovery," in *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015, pp. 631–639.
[8] X. Fu, K. Huang, and N. D. Sidiropoulos, "On identifiability of nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 328–332, 2018.
[9] W.-K. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C.-Y. Chi, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 67–81, 2013.
[10] K. Huang, X. Fu, and N. D. Sidiropoulos, "Anchor-free correlated topic modeling: Identifiability and algorithm," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
[11] ——, "Learning hidden Markov models from pairwise co-occurrences with application to topic modeling," in *International Conference on Machine Learning (ICML)*, 2018, pp. 2073–2082.
[12] K. Huang and X. Fu, "Detecting overlapping and correlated communities without pure nodes: Identifiability and algorithm," in *International Conference on Machine Learning*, 2019, pp. 2859–2868.
[13] S. Ibrahim, X. Fu, N. Kargas, and K. Huang, "Crowdsourcing via pairwise co-occurrences: Identifiability and algorithms," *Advances in neural information processing systems*, vol. 32, 2019.
[14] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, "A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4418–4432, 2009.
[15] J. M. Bioucas-Dias, "A variable splitting augmented lagrangian approach to linear spectral unmixing," in *2009 First workshop on hyperspectral image and signal processing: Evolution in remote sensing*. IEEE, 2009, pp. 1–4.
[16] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.
[17] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.
[18] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[19] M. Jaggi, "Revisiting frank-wolfe: Projection-free sparse convex optimization," in *International conference on machine learning*. PMLR, 2013, pp. 427–435.