A MODIFIED FRANK-WOLFE ALGORITHM FOR TENSOR FACTORIZATION WITH UNIMODAL SIGNALS

Junting Chen[1,2] *and Urbashi Mitra*[1]

[1]Ming Hsieh Department of Electrical Engineering, University of Southern California, USA
[2]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China

ABSTRACT

Unimodality-constrained matrix or tensor factorization has applications in various domains, such as non-parametric source localization and data clustering, where the signals of interest are unimodal. Such factorizations are challenged by the non-convex nature of unimodality constraints. This paper develops a modified Frank-Wolfe algorithm with a successive programming technique, which produces a sequence of linear subproblems with modified and adaptive constraints. The algorithm is proven to converge and the subproblems are shown to be solved easily. In an application example of solving unimodality-constrained tensor factorization problems, the proposed algorithm demonstrates substantial complexity reduction while achieving the same convergence performance as compared to a brute-force projected gradient algorithm.

***Index Terms*—** Unimodal, Frank-Wolfe algorithm, tensor decomposition, source localization, data fusion

1. INTRODUCTION

In source localization, if a sensor moves in a straight line to measure the source signal, the received signal strength (RSS) first increases to a peak and decreases thereafter. Such a unimodality is essential for finding a source in harsh environment where the signal propagation characteristic is not known. For example, for an acoustic source in the ocean, it is difficult to determine either the propagation speed or the power decay rate of the acoustic signal [1, 2]. In other localization scenarios, such as searching for a primary user in cognitive radio or finding a jammer in sensor networks [3, 4], the target does not cooperate with the sensors. As a result, conventional range-based trilateration is not applicable and it is essential to exploit unimodality for localization [5, 6]. Furthermore, unimodality is also found in many estimation problems in various domains including biostatistics, chemometrics, and data mining for medical treatments [7–12].

The various estimation problems mentioned above can be generally formulated into the following unimodality-constrained problem

$$\mathscr{P}: \quad \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in \mathscr{U} \cap \mathcal{M}$$

where the vector variable $\boldsymbol{x}$ is the target signal that is believed to be unimodal, $\mathcal{M} = \{\boldsymbol{x} : a \leq \|\boldsymbol{x}\|_1 \leq b\}$ for some $a, b \in [0, \infty]$, and $\mathscr{U}$ represents the set of all unimodal vectors. Specifically, a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathscr{U}$ is unimodal if

$$\begin{aligned} 0 \leq & x_1 \leq x_2 \leq \cdots \leq x_s \\ & x_s \geq x_{s+1} \geq \cdots \geq x_n \geq 0 \end{aligned} \tag{1}$$

are satisfied for some $1 \leq s \leq n$.

Relation to Prior Work: Recent techniques to solve $\mathscr{P}$ are based on unimodal projections. For example, [7] develops a projected alternating minimization algorithm, and [5] provides a projected gradient algorithm for a matrix factorization objective f. Specifically, projecting $\boldsymbol{x}$ onto $\mathscr{U}$ is equivalent to minimizing $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|$ subject to $\hat{\boldsymbol{x}} \in \mathscr{U}$. There exists many efficient isotonic projection algorithms [13–15]. It is argued in [14] that an exact unimodal projection requires $\mathcal{O}(n)$ time under the L_2-norm metric, whereas, it requires $\mathcal{O}(n \log n)$ time under the L_1 metric. However, these projection techniques may not be easily extended to general constraint sets $\mathscr{U} \cap \mathcal{M}$ with the same efficiency. Note that the overall complexity to solve $\mathscr{P}$ is very sensitive to the efficiency of the projection.

In this paper, we focus on projection-free algorithms to solve $\mathscr{P}$. The fundamental idea is to construct a series of easy-to-solve subproblems, and iteratively approximate the solution by solving the subproblems. A widely researched algorithm of this kind is the Frank-Wolfe algorithm [16–20]. However, Frank-Wolfe algorithms require the constraint set to be convex. For non-convex constraints, the Frank-Wolfe update is not guaranteed to stay inside the constraint set. Other related solutions include sequential programming [21, 22], which successively constructs simplified (convex) constraints from the original problem. However, none of them were designed to handle the unimodality constraints.

Our Contributions: The specific challenge for $\mathscr{P}$ is how to dynamically design simplified constraint subsets for inner

This research has been funded in part by one or more of the following grants: ONR N00014-15-1-2550, NSF CNS-1213128, NSF CCF-1718560, NSF CCF-1410009, NSF CPS-1446901, and AFOSR FA9550-12-1-0215.

subproblems such that (i) *the subproblems are easy to solve,* and (ii) *the algorithm can converge to a reasonably good solution.* In tackling these challenge, this paper makes the following contributions: First, we develop a unimodal Frank-Wolfe (U-FW) algorithm based on successive programming, where each iteration solves a linear program (LP) constrained by a dynamic set; the LP can be solved by at most $2n$ steps. Second, we prove the convergence of the algorithm and show that $\mathcal{O}(1/\sqrt{t})$ convergence rate can be achieved using an adaptive step size. Third, we demonstrate the application of the U-FW algorithm for solving unimodality-constrained tensor factorization problems. In an application of multimodal data fusion for source localization, our numerical results demonstrate good convergence and the computational efficiency of the U-FW algorithm.

2. ALGORITHM DESIGN

2.1. Review of Frank-Wolfe Algorithm

If the unimodality constraint $\boldsymbol{x} \in \mathscr{U}$ is not present, the basic procedure of Frank-Wolfe algorithm is to sequentially compute an approximate solution to $\mathscr{P}$ by linearizing the objective f based on the current iterate $\boldsymbol{x}^{(t)}$. Specifically, $\boldsymbol{x}^{(t+1)}$ is updated as

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \lambda_t(\hat{\boldsymbol{y}} - \boldsymbol{x}^{(t)}) \tag{2}$$

where $\hat{\boldsymbol{y}} = \arg\min_{\boldsymbol{y}\in\mathcal{M}} \nabla f(\boldsymbol{x}^{(t)})^{\mathrm{T}}\boldsymbol{y}$ with λ_t being the step size. Since the constraint set $\mathcal{M}$ of interest is convex and $\boldsymbol{x}^{(t+1)}$ is constructed as a convex combination of $\hat{\boldsymbol{y}}$ and $\boldsymbol{x}^{(t)}$, $\boldsymbol{x}^{(t+1)}$ will stay in $\mathcal{M}$. Therefore, the prerequisites of the Frank-Wolfe algorithms are, first, the constraints are convex, and second, the subproblem $\min_{\boldsymbol{y}\in\mathcal{M}} \nabla f(\boldsymbol{x}^{(t)})^{\mathrm{T}}\boldsymbol{y}$ can be solved efficiently.

The basic step of successive programming is similar except that it also approximates the constraint set using a convex subset:

$$\begin{aligned} \mathscr{P}_{\text{SP}}: \quad & \underset{\boldsymbol{y}\in\mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{x}^{(t)}) + \nabla f(\boldsymbol{x}^{(t)})^{\mathrm{T}}\boldsymbol{y} \\ & \text{subject to} \quad \boldsymbol{y} \in U(\boldsymbol{x}^{(t)}) \end{aligned}$$

where $U(\boldsymbol{x}) \subseteq \mathscr{U} \cap \mathcal{M}$ is a convex set that contains $\boldsymbol{x}$.

However, it is a challenge to design the dynamic subset $U(\boldsymbol{x})$ for $\mathscr{P}_{\text{SP}}$, because the design of $U(\boldsymbol{x})$ affects both the convergence and the computational complexity to solve $\mathscr{P}_{\text{SP}}$. The rest of the paper will focus on constructing $U(\boldsymbol{x})$ and prove the convergence of the algorithm.

2.2. Properties of the Unimodal Set

Let $\mathcal{U}_s$ be the set of vectors $\boldsymbol{x}$ that satisfy the conditions (1) with the sth entry being the largest one. As a result, the unimodal set $\mathscr{U}$ can be written as $\mathscr{U} = \bigcup_{s=1}^{n} \mathcal{U}_s$. Intuitively, one should construct $U(\boldsymbol{x})$ to cover as many component sets $\mathcal{U}_s$'s as possible, so that the update (2), as driven by the solution of $\mathscr{P}_{\text{SP}}$, can move from one component set $\mathcal{U}_s$ to another. We find that a good candidate for $U(\boldsymbol{x})$ is $\widetilde{\mathcal{U}}_s = \text{conv}\big(\mathcal{U}_{s-1} \cup \mathcal{U}_s \cup \mathcal{U}_{s+1}\big)$, where we define $\mathcal{U}_0 = \mathcal{U}_{n+1} = \varnothing$. A nice property of $\tilde{\mathcal{U}}_s$ is that the set can be characterized by simple linear constraints.

Proposition 1. *A vector $\boldsymbol{x} \in \widetilde{\mathcal{U}}_s = \text{conv}\big(\mathcal{U}_{s-1} \cup \mathcal{U}_s \cup \mathcal{U}_{s+1}\big)$ if and only if the following conditions are satisfied:*

$$0 \le x_1 \le x_2 \le \cdots \le x_{s-1}, \qquad \text{if } s \ge 2 \tag{3}$$

$$x_s \ge 0, \tag{4}$$

$$x_{s+1} \ge x_{s+2} \ge \cdots \ge x_n \ge 0, \quad \text{if } s \le n-1. \tag{5}$$

Let $s = S(\boldsymbol{x}) \triangleq \arg\max_{1\le i\le n} x_i$ be the index where x_s takes the largest value. Since $\widetilde{\mathcal{U}}_s$ is unbounded, we form the dynamic constraint set $U(\boldsymbol{x})$ by intersecting $\widetilde{\mathcal{U}}_s$ with two hyperplanes $\|\boldsymbol{x}\|_1 \ge a$ and $\|\boldsymbol{x}\|_1 \le b$. More specifically,

$$\begin{aligned} U(\boldsymbol{x}) := \widetilde{\mathcal{U}}_{S(\boldsymbol{x})} \cap & \\ \{\boldsymbol{y} \in \mathbb{R}^n : a(\boldsymbol{x}) \le \|\boldsymbol{y}\|_1 \le b(\boldsymbol{x})\} & \end{aligned} \tag{6}$$

where $a(\boldsymbol{x}) = \min\{1, \hat{\alpha}(\boldsymbol{x})\}\|\boldsymbol{x}\|_1, b(\boldsymbol{x}) = \max\{1, \hat{\alpha}(\boldsymbol{x})\}\|\boldsymbol{x}\|_1$, and $\hat{\alpha}(\boldsymbol{x})$ is the solution that minimizes $f(\alpha\boldsymbol{x})$ subject to $\alpha\boldsymbol{x} \in \mathcal{M}$. Note that minimizing $f(\alpha\boldsymbol{x})$ can be solved efficiently under many popular cost functions. For example, when f is a quadratic function of $\boldsymbol{x}$, the solution $\hat{\alpha}(\boldsymbol{x})$ can be found with closed forms. In general, $\hat{\alpha}(\boldsymbol{x})$ can be computed using bisection search for the root of $\partial f(\alpha\boldsymbol{x})/\partial\alpha$, although global optimality is only guaranteed when f is convex.

We find that the polytope $U(\boldsymbol{x})$ has at most $2n$ extreme points.

Proposition 2. *The polygon $U(\boldsymbol{x})$ defined in* (6) *has n extreme points when $a(\boldsymbol{x}) = b(\boldsymbol{x})$, and $2n$ extreme points when $a(\boldsymbol{x}) \ne b(\boldsymbol{x})$.*

Proof. (Sketch) The proof can be established by finding the extreme points according to (3)–(5) and (6). □

Proposition 2 implies that if one uses a simplex method to solve $\mathscr{P}_{\text{SP}}$, the solution can be found by at most $2n$ steps, because the simplex method moves only at the extreme points of the constraint set [23,24]. This further suggests that, handling the unimodality constraint may just add in marginal complexity using the proposed strategy.

Finally, it turns out that although $U(\boldsymbol{x}) \nsubseteq \mathscr{U}$, the update $\boldsymbol{x} + \lambda(\hat{\boldsymbol{y}} - \boldsymbol{x})$ for any $\lambda \in [0,1]$ and $\hat{\boldsymbol{y}}$ as the solution to $\mathscr{P}_{\text{SP}}$ still belongs to the unimodal set $\mathscr{U}$.

Proposition 3. *Suppose that $\hat{\alpha}(\boldsymbol{x})$ is bounded. Let $\hat{\boldsymbol{y}}$ be the solution to $\mathscr{P}_{LP}$ with $U(\boldsymbol{x})$ defined from* (6). *If $\boldsymbol{x} \in \mathscr{U}$, then $\boldsymbol{x}' = \boldsymbol{x} + \lambda(\hat{\boldsymbol{y}} - \boldsymbol{x}) \in \mathscr{U}$ for any $\lambda \in [0,1]$.*

Proof. (Sketch) The solution of an LP is always at the extreme points of the constraint set, and the extreme points always belong to $\mathscr{U}$. □

Algorithm 1 Unimodal Frank-Wolfe (U-FW) algorithm

1. Initialization: Choose $\boldsymbol{x}^{(0)} \in \mathscr{U} \cap \mathcal{M}$ and a (small) stopping threshold $\epsilon > 0$.
2. At each step $t = 0, 1, \ldots$, compute the constraint set $U(\boldsymbol{x}^{(t)})$ according to (6).
3. Find $\hat{\boldsymbol{y}}$ as the solution to $\mathscr{P}_{\text{SP}}$.
4. Update $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \lambda_t(\hat{\boldsymbol{y}} - \boldsymbol{x}^{(t)})$, where $\lambda_t \in [0, 1]$. A possible choice of λ_t is $\lambda_t = 2/(t+2)$.
5. Repeat from Step 2 until $\nabla f(\boldsymbol{x}^{(t)})^{\mathrm{T}}(\boldsymbol{x}^{(t)} - \hat{\boldsymbol{y}}) < \epsilon$

2.3. Algorithm Design

Using the properties in Propositions 2 – 3, one can construct an algorithm whose trajectory never leaves the constraint set $\mathscr{U} \cap \mathcal{M}$, and, at the same time, can be computed efficiently. This is summarized in Algorithm 1.

3. CONVERGENCE ANALYSIS

While $\mathscr{P}$ is non-convex, we evaluate the convergence by analyzing the *dual gap* defined below

$$g(\boldsymbol{x}) = \max_{\boldsymbol{w} \in U(\boldsymbol{x})} -\nabla f(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{w} - \boldsymbol{x}) \tag{7}$$

which can be shown to be a lower bound of the duality gap of problem $\mathscr{P}$ in the Frank-Wolfe algorithm literature [16–20].[1]

The dual gap $g(\boldsymbol{x})$ in (7) explains the stopping criterion in Step 5 of Algorithm 1, which computes $\nabla f(\boldsymbol{x}^{(t)})^{\mathrm{T}}(\boldsymbol{x}^{(t)} - \hat{\boldsymbol{y}}) = g(\boldsymbol{x}^{(t)})$. Note that $g(\boldsymbol{x}) \geq 0$ because if one takes $\boldsymbol{w} = \boldsymbol{x} \in U(\boldsymbol{x})$, the right hand side (R.H.S.) of (7) becomes 0. Thus, $g(\boldsymbol{x}) = 0$ defines a stationary point.

To study the convergence, the following mild condition is assumed.

Assumption 1 (Smoothness)**.** *The objective function f is differentiable and its gradient ∇f is Lipschitz continuous, i.e., there exists $L < \infty$, such that $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}$ under some norm $\|\cdot\|$.*

From Assumption 1, the Lipschitz smoothness condition implies that [24, 25]

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2. \tag{8}$$

Moreover, the fact that $\mathcal{M}$ is compact implies that $M_s = \sup_{\boldsymbol{x}, \boldsymbol{y} \in \tilde{\mathcal{U}}_s \cap \mathcal{M}} \|\boldsymbol{x} - \boldsymbol{y}\|^2$ are finite, for $s = 1, 2, \ldots, n$.

[1] However, the existing analysis for the conventional Frank-Wolfe method requires the constraint to be convex, which is not the case here.

Applying the update rule $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \lambda_t(\hat{\boldsymbol{y}} - \boldsymbol{x}^{(t)})$ to the Lipschitz smoothness inequality (8), one arrives at

$$\begin{aligned} &f(\boldsymbol{x}^{(t+1)}) \\ &\quad \leq f(\boldsymbol{x}^{(t)}) + \gamma_t \nabla f(\boldsymbol{x}^{(t)})^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{x}^{(t)}) + \gamma_t^2 \frac{L}{2}\|\hat{\boldsymbol{y}} - \boldsymbol{x}^{(t)}\|^2 \\ &\quad \leq f(\boldsymbol{x}^{(t)}) - \gamma_t g(\boldsymbol{x}^{(t)}) + \frac{\gamma_t^2 L M_s}{2}. \end{aligned} \tag{9}$$

Manipulating the above inequality (9), the following convergence result can be easily obtained.

Theorem 1 (Convergence)**.** *Algorithm 1 terminates after finite steps $T < \infty$. Specifically, $\min_{0 \leq k \leq t} g(\boldsymbol{x}^{(k)}) \to 0$ as $t \to \infty$.*

Proof. (Sketch) The result follows from inequality (9) and the fact that $f(\boldsymbol{x}^{(0)}) - f(\boldsymbol{x}^*)$ is bounded. □

In addition, if one knows the parameters L and M_s, an adaptive step size λ_t can be derived by minimizing the upper bound (9). The solution is given by $\lambda_t^* = \frac{g(\boldsymbol{x}^{(t)})}{LM_s}$. Therefore, the step size rule can be designed as $\lambda_t = \min\{\frac{g(\boldsymbol{x}^{(t)})}{LM_s}, 1\}$. With this design, the algorithm can be shown to converge at a $\mathcal{O}(1/\sqrt{t})$ rate.

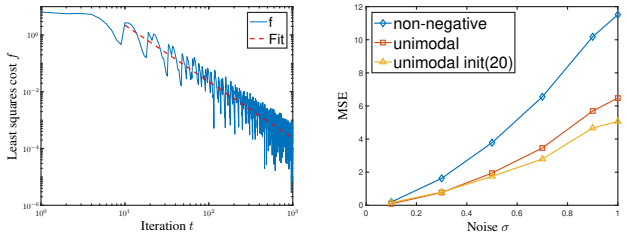
Theorem 2 (Rate of Convergence)**.** *Suppose that step size rule $\lambda_t = \min\{\frac{g(\boldsymbol{x}^{(t)})}{LM_s}, 1\}$ is used in Algorithm 1. Then Algorithm 1 terminates after $\mathcal{O}(1/\epsilon^2)$ steps. Specifically,*

$$\min_{0 \leq k \leq t} g(\boldsymbol{x}^{(k)}) \leq \frac{\max\{\sqrt{2h_0 C}, 2h_0\}}{\sqrt{t+1}}$$

where $h_0 = f(\boldsymbol{x}^{(0)}) - f^*$ is the initial gap, in which f^* is the minimum objective value.

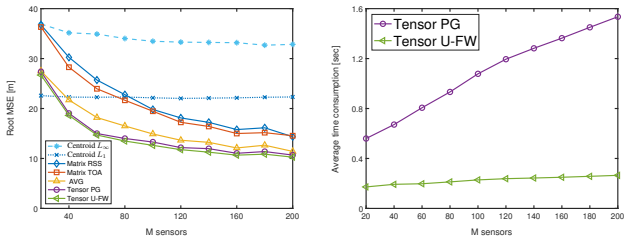
Proof. (Sketch) With the dual gap $g(\boldsymbol{x})$ defined in (7), the convergence proof follows an approach similar to [17]. □

As an example, consider estimating a unimodal signal $\boldsymbol{c} \in \mathbb{R}_+^n$, with $n = 20$, generated by projecting a vector $\boldsymbol{c}_0$ onto the unimodal set $\mathscr{U}$, where each element of $\boldsymbol{c}_0$ follows an independent, uniform distribution over $[0, 1]$. Assume that the observation is given by $\boldsymbol{z} = \boldsymbol{c} + \boldsymbol{n}$, where $\boldsymbol{n} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Consider recovering $\boldsymbol{c}$ by minimizing the least squares cost $f(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{z}\|_2^2$. As a result, the corresponding unimodality-constrained least squares estimation problem can be formulated as $\mathscr{P}$ with $\mathcal{M} = \mathbb{R}^n$. Fig. 1 (a) shows a realization of the algorithm trajectory for Algorithm 1. The observed convergence rate of the objective f is roughly $t^{-1.99}$. Fig. 1 (b) shows the recovery performance in terms of the mean squared error (MSE) $\mathbb{E}\{\|\hat{\boldsymbol{x}} - \boldsymbol{c}\|_2^2\}$ versus the standard deviation σ of the noise. The non-negative scheme simply projects the observation $\boldsymbol{z}$ onto $\mathbb{R}_+^n$. The "unimodal init(20)" scheme performs Algorithm 1 with 20 random initializations and picks the solution that yields the minimum cost. The result shows that imposing the unimodality constraints indeed better recovers the desired signal.



(a) Convergence under $\sigma = 0$ (b) MSE Performance

Fig. 1. Example under a quadratic cost



(a) Localization MSE (b) Time consumption

Fig. 2. Localization accuracy and computational complexity

4. APPLICATION IN TENSOR FACTORIZATION FOR SOURCE LOCALIZATION

Consider localizing an active signal source using a set of passive sensors deployed randomly in a 200×200 m^2 area. The sensors measure the signals emitted from the source. There could be multiple types of signals to measure, such as the RSS signals of electromagnetic waveforms and the time-of-arrival (TOA) signals of the waveforms. The signals are *significantly* noise disturbed to model a harsh environment. The noise-normalized RSS signal is modeled as $P_{\text{dB}}(d) = 70 - 36 \times \log_{10}(\max\{10, d\}) + \xi$, where $\xi \sim \mathcal{N}(0, \sigma_{\text{s}}^2)$ is to model log-normal shadowing and $\sigma_{\text{s}} = 10$ dB. The TOA signal is modeled as $t(d) = d/c + b$, where $c = 340$ m/s and $b \sim \mathcal{N}(0, \sigma_{\text{t}}^2)$ is to model synchronization errors and $\sigma_{\text{t}} = 100$ ms. To normalize the data, we use $h_1(d) = \exp(-\beta_1 10^{-P_{\text{dB}}(d)/10})$ and $h_2(d) = \exp(-\beta_2 t(d)^2)$. However, these models are not known by the system.

In [26], it was observed that a good way to fuse the multimodal data for non-parametric source localization is to arrange the data according to the sampling positions into a tensor, *i.e.*, a 3D data array, and apply tensor processing techniques to interpret the hidden unimodal structure of the data. Specifically, for each type of data, arrange the measurements according to their sampling positions into a sparse matrix $\mathbf{H}_k$, fill in the missing values using matrix completion methods to obtain a completed matrix $\boldsymbol{X}_k$, and stack $\boldsymbol{X}_k$'s to form a tensor $\boldsymbol{\mathcal{X}}$. Then, solve the following tensor factorization problem

$$\begin{aligned} \mathscr{P}_{\text{UTF}}: \underset{\alpha, \boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3}{\text{minimize}} \quad & \|\boldsymbol{\mathcal{X}} - \alpha \times_1 \boldsymbol{w}_1 \times_2 \boldsymbol{w}_2 \times_3 \boldsymbol{w}_3\|_{\text{F}}^2 \\ \text{subject to} \quad & \alpha > 0, \|\boldsymbol{w}_1\|_1 = \|\boldsymbol{w}_2\|_1 = \|\boldsymbol{w}_3\|_1 = 1 \\ & \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathscr{U}. \end{aligned}$$

This strategy yields the *signature vectors* $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ with their peaks locations indicating the source location. In this example, the peak locations of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are estimated using the reflected-correlation estimator developed in [5].

While the prior work [26] ignored the unimodality constraints $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathscr{U}$, $\mathscr{P}_{\text{UTF}}$ can now be efficiently solved using Algorithm 1. As a baseline, the "Tensor PG" scheme solves $\mathscr{P}_{\text{UTF}}$ using the projected gradient algorithm [5]. For a benchmark, we also compare the localization performance with weighted "Centroid" schemes [3] with L_∞-weight (just consider the sensor location that records the maximum measurement) and L_1-weight (a linear combinations of the sensor location by the measurement data), as well as "Matrix-RSS" and "Matrix-TOA" schemes [5, 26].

Fig. 2 (a) shows the localization accuracy versus the number of sensors. Essentially, the proposed scheme and "Tensor PG" schemes that are based on tensor models outperform all the baselines. The proposed scheme performs slightly better than the "Tensor PG" scheme. At the same time, the proposed algorithm is much more computationally efficient than projected gradient as seen from Fig. 2 (b).

We can also observe the benefit of exploiting the unimodal structure for localization. First, the "Centroid" schemes perform the worst, because they do not exploit the unimodal structure of the data. Second, the Matrix-RSS and Matrix-TOA schemes work slightly better than the naive scheme, as the matrix model also explicitly exploits the unimodal structure. However, both schemes can only exploit a portion of the measurement data. Third, the AVG scheme fuses the results by simply averaging the estimates $\hat{\mathbf{s}}'_{\text{RSS}}$ and $\hat{\mathbf{s}}'_{\text{TOA}}$ from the Matrix-RSS and Matrix-TOA schemes. Thus, such a result is not robust. From our experiment, the average scheme requires roughly 50% more measurement data to achieve the same localization accuracy from the tensor-based methods.

5. CONCLUSIONS

This paper developed a unimodal Frank-Wolfe algorithm to solve unimodality-constrained optimization problems. The algorithm approximates the original problem into a series of linear subproblems, where each subproblem is constrained by a different subset such that it can be solved with linear complexity. Convergence has been proven under various step size rules. In an application of multimodal data fusion for non-parametric source localization, the proposed algorithm was applied to solve tensor factorization problems. The proposed algorithm demonstrated substantial complexity reduction with no performance lost compared to a brute-force projected gradient algorithm.

6. REFERENCES

[1] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 84–89, 2009.

[2] L. M. Brekhovskikh, *Fundamentals of ocean acoustics*. Springer Science & Business Media, 2003.

[3] J. Wang, P. Urriza, Y. Han, and D. Cabric, "Weighted centroid localization algorithm: theoretical analysis and distributed implementation," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3403–3413, 2011.

[4] Z. Liu, H. Liu, W. Xu, and Y. Chen, "Exploiting jamming-caused neighbor changes for jammer localization," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 547–555, 2012.

[5] J. Chen and U. Mitra, "Unimodality-constrained matrix factorization for non-parametric source localization," *IEEE Trans. Signal Process.*, 2019, to appear, preprint arXiv:1711.07457.

[6] S. Choudhary, N. Kumar, S. Narayanan, and U. Mitra, "Active target localization using low-rank matrix completion and unimodal regression," *arXiv preprint arXiv:1601.07254*, 2016.

[7] R. Bro and N. D. Sidiropoulos, "Least squares algorithms under unimodality and non-negativity constraints," *Journal of Chemometrics*, vol. 12, no. 4, pp. 223–247, 1998.

[8] S. Chatterjee and J. Lafferty, "Adaptive risk bounds in unimodal regression," *arXiv preprint arXiv:1512.02956*, 2015.

[9] L. H. Gunn and D. B. Dunson, "A transformation approach for incorporating monotone or unimodal constraints," *Biostatistics*, vol. 6, no. 3, pp. 434–449, 2005.

[10] M. C. Meyer, "Nonparametric estimation of a smooth density with shape restrictions," *Statistica Sinica*, pp. 681–701, 2012.

[11] D. Z. Chen, C. Wang, and H. Wang, "Representing a functional curve by curves with fewer peaks," *Discrete & Computational Geometry*, vol. 46, no. 2, pp. 334–360, 2011.

[12] C. Köllmann, B. Bornkamp, and K. Ickstadt, "Unimodal regression using bernstein–schoenberg splines and penalties," *Biometrics*, vol. 70, no. 4, pp. 783–793, 2014.

[13] A. Németh and S. Németh, "How to project onto an isotone projection cone," *Linear Algebra and its Applications*, vol. 433, no. 1, pp. 41–51, 2010.

[14] Q. F. Stout, "Unimodal regression via prefix isotonic regression," *Computational Statistics & Data Analysis*, vol. 53, no. 2, pp. 289–297, 2008.

[15] J. Chun, K. Sadakane, and T. Tokuyama, "Linear time algorithm for approximating a curve by a single-peaked curve," *Algorithmica*, vol. 44, no. 2, pp. 103–115, 2006.

[16] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization." in *ICML (1)*, 2013, pp. 427–435.

[17] S. Lacoste-Julien, "Convergence rate of Frank-Wolfe for non-convex objectives," *arXiv preprint arXiv:1607.00345*, 2016.

[18] H.-T. Wai, J. Lafond, A. Scaglione, and E. Moulines, "Decentralized Frank–Wolfe algorithm for convex and nonconvex problems," *IEEE Trans. on Automatic Control*, vol. 62, no. 11, pp. 5522–5537, 2017.

[19] D. Garber and E. Hazan, "Faster rates for the Frank-Wolfe method over strongly-convex sets," *arXiv preprint arXiv:1406.1305*, 2014.

[20] H. Zhao and G. Gordon, "Frank-Wolfe optimization for symmetric-NMF under simplicial constraint," *arXiv preprint arXiv:1706.06348*, 2017.

[21] J. Bolte and E. Pauwels, "Majorization-minimization procedures and convergence of SQP methods for semi-algebraic and tame programs," *Mathematics of Operations Research*, vol. 41, no. 2, pp. 442–465, 2016.

[22] B. Jiang, T. Lin, S. Ma, and S. Zhang, "Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis," *arXiv preprint arXiv:1605.02408*, 2016.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[24] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[25] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[26] J. Chen and U. Mitra, "A tensor decomposition technique for source localization from multimodal data," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Calgary, Alberta, Canada, Apr. 2018.