Hello everyone, My name is Tri Nguyen. Thank you for serving as my committee. Today I'm glad to present my work for my qualifying exam.

Memory-Efficient Separable Simplex-Structured Matrix Factorization via the Frank-Wolfe Method

Tri Nguyen

Qualifying Exam
Oregon State University

May 2, 2022

- A brief outline of the talk. We'll go through 4 parts. In the first section, I'll introduce the problem, elaborate the setting, and show some examples why we are interested in it.
- Then a quick look at how this problem has been solved. Particularly, there are 2 related different approaches, and we'll see what are their drawback and what could we offer to improve.
- In the third part, we advocate using an optimization method named Frank-Wolfe. We'll try to show our intuition and rationale behind the proposal.
- Lastly, we showcase performance of the proposal via **both** synthetic experiment and real data experiments.

Outline

- We are interesting a branch of matrix factorization, namely simplex structure matrix factorization.
- In particular, the model assumes that the data matrix $\boldsymbol{X}$ is generated as a production of 2 low-rank matrix $\boldsymbol{W}, \boldsymbol{H}$, as we will refer as latent factor. The inner dimension K is assumed to be relatively small compared to M, N.
- In addition, it is assumed that columns of $\boldsymbol{H}$ reside in a probability simplex. Note that we do not require nonnegativity on $\boldsymbol{W}$ as in NMF.
- This model are closely related to NMF in a sense that we can always convert a NMF model into SSMF model by performing a normalization on columns of $\boldsymbol{X}$.
- This model has witnessed a large interest from various domain, including machine learning, signal processing.
- What? So the problem is: Given $\boldsymbol{X}$, how do we find the ground truth $\boldsymbol{W}, \boldsymbol{H}$.
- Why? Finding $\boldsymbol{W}, \boldsymbol{H}$ is meaningful as they carries physical meaning depending on particular applications.
- T.-H. Chan et al. 2008 is using pure pixel model in addition to SSMF

Simplex Structured Matrix Factorization

Simplex Structured Matrix Factorization (SSMF)

Data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times M}$ is assumed to be generated by $\boldsymbol{W} \in \mathbb{R}^{N \times K}, \boldsymbol{H} \in \mathbb{R}^{K \times M}, K \ll \min(M, N)$ such that

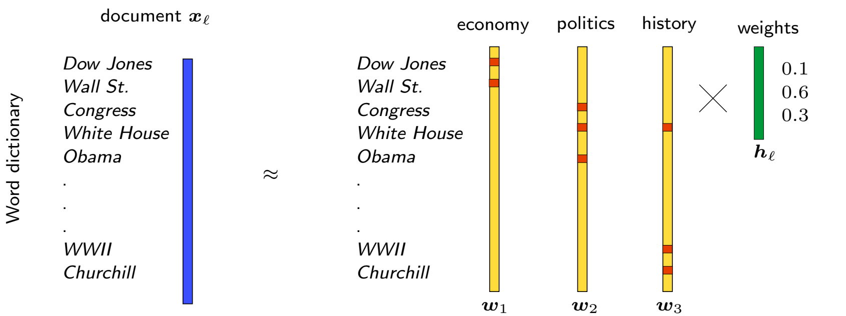
$$\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H} + \boldsymbol{V} \quad \text{subject to } \boldsymbol{H} \geq 0, \mathbf{1}^\top \boldsymbol{H} = \mathbf{1}^\top$$

Given $\boldsymbol{X}$,how do we find the latent factors $\boldsymbol{W}, \boldsymbol{H}$?

- Closely related to nonnegative matrix factorization.
- Has received significant attention across many domains [S. Arora et al. 2012; Sanjeev Arora et al. 2013; T.-H. Chan et al. 2008; X. Fu et al. 2016; Huang et al. 2019; Keshava et al. 2002; Mao et al. 2017b; Panov et al. 2017; Recht et al. 2012]

- For example, in topic modeling, if a document is presented using bag-of-word, then a document is assumed to be a convex combination of some small set of topics, in this case presented as 3 vectors:...
- The task of topic discovery is to find the latent representation of topic, in this case the $\boldsymbol{W}$ matrix. Note that coefficients are sum-up-to 1 in this case.

Application: Topic Modeling



$\boldsymbol{x}_\ell \approx 0.1 \times \boldsymbol{w}_1 + 0.6 \times \boldsymbol{w}_2 + 0.3 \times \boldsymbol{w}_3$

doc. $\approx 0.1 \times$ **economy** $+ 0.6 \times$ **politics** $+ 0.3 \times$ **history**

high prob. word

A demonstration of $\boldsymbol{x}_\ell \approx \boldsymbol{W}\boldsymbol{h}_\ell$

- $\boldsymbol{X}$ is a vocab-document matrix, then $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H}$ where
  - $\boldsymbol{H} \geq 0, \mathbf{1}^\top \boldsymbol{H} = \mathbf{1}^\top$
  - K is number of topics
- This model has been used in [S. Arora et al. 2012; Sanjeev Arora et al. 2013, 2016; Huang et al. 2016; Recht et al. 2012]

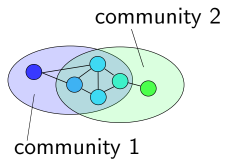
- Another application is community detection, where given a graph, we wish to discovery a small number of community constituted by set of nodes that are considered to be close to each other.
- As a well-known model, the mixed membership stochastic blockmodels models . Under this model, the task amounts to finding the membership matrix $\boldsymbol{H}$ from an observed adjacency matrix.
- As a membership vector, it is natural that $\boldsymbol{h}$ is in a probability simplex.
- This is nothing but the SSMF as we saw earlier.

Application: Community Detection

- The mixed membership stochastic blockmodels [Airoldi et al. 2008]

$$P_{i,j} = \boldsymbol{h}_i^\top \boldsymbol{B} \boldsymbol{h}_j$$
$$\boldsymbol{A}(i,j) = \boldsymbol{A}(j,i) \sim \text{Bernoulli}(\boldsymbol{P}(i,j))$$

where $\boldsymbol{h}_i = [h_{1,i}, \ldots, h_{K,i}]^\top$ represents membership of node i, $\boldsymbol{B}$ represents community-community connection.



Demonstration of a graph with $K = 2$ communities

  - By physical interpretation, $\boldsymbol{H} \geq 0, \mathbf{1}^\top \boldsymbol{H} = \mathbf{1}^\top$.
  - Range space of $\boldsymbol{H}$ can be estimated from K leading eigenvectors of $\boldsymbol{A}$. [Lei et al. 2015; Mao et al. 2017a,b; Panov et al. 2017]

$$\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H} + \boldsymbol{N}$$

- Note that we do not just want to explain $\boldsymbol{X}$ with some matrices $\boldsymbol{W}, \boldsymbol{H}$. What we really want is finding the ground truth $\boldsymbol{W}^\star, \boldsymbol{H}^\star$. However, NMF in general is an NP-hard problem.
- A natural attempt is to consider finding ..., and hope that the found solution could reveal $\boldsymbol{W}^\star, \boldsymbol{H}^\star$
- Unfortunately, the solution of Problem 1 is not unique. We can see why. It is trivial to construct $\boldsymbol{Q}$ ...
- $\boldsymbol{W}', \boldsymbol{H}'$ in this case would not bring much useful information. For example, in topic modeling , $\boldsymbol{W}^\star$ represent the topics, while the sought $\boldsymbol{W}'$ represent a mixed of topics. Hence we haven't really got a good representation of topics by using $\boldsymbol{W}'$.
- By that reason, we focus our interest to those models whose can be identified. In particular, by saying SSMF model is identifiable we mean that if criterion (1) has solution $\boldsymbol{W}, \boldsymbol{H}$, then they are just some permutation of the ground truth.
- This kind of ambiguity is unharmed.
- The definition is borrowed from this work and modified to our specific SSMF problem.
- the point is: problem is hard

Identifiability

- Given a SSMF model with $\boldsymbol{X} = \boldsymbol{W}^\star \boldsymbol{H}^\star$, finding $\boldsymbol{W}^\star, \boldsymbol{H}^\star$ is a difficult problem.

$$\begin{aligned} &\text{find} && \boldsymbol{W}, \boldsymbol{H} && \text{(1a)} \\ &\text{subject to} && \boldsymbol{X} = \boldsymbol{W}\boldsymbol{H} && \text{(1b)} \\ & && \boldsymbol{H} \geq 0, \mathbf{1}^\top \boldsymbol{H} = \mathbf{1}^\top && \text{(1c)} \end{aligned}$$

- The solution is not unique. There exists non-singular $\boldsymbol{Q}$ such that

$$\boldsymbol{X} = \boldsymbol{W}^\star \boldsymbol{H}^\star = \underbrace{(\boldsymbol{W}^\star \boldsymbol{Q}^{-1})}_{\boldsymbol{W}'}\underbrace{(\boldsymbol{Q}\boldsymbol{H}^\star)}_{\boldsymbol{H}'}, \text{ and } \boldsymbol{H}' \geq 0, \mathbf{1}^\top \boldsymbol{H}' = \mathbf{1}^\top$$

Definition (Identifiability [Xiao Fu et al. 2019])

A SSMF model where $\boldsymbol{X} = \boldsymbol{W}^\star \boldsymbol{H}^\star$ is called identifiable respect to criterion (1) if for all $\boldsymbol{W}, \boldsymbol{H}$ satisfying criterion (1), it holds that $\boldsymbol{W} = \boldsymbol{W}^\star \boldsymbol{\Pi}, \boldsymbol{H} = \boldsymbol{\Pi}^\top \boldsymbol{H}^\star$, where $\boldsymbol{\Pi}$ is a permutation matrix.

- There have been several works that investigate the conditions that can guarantee identifiability. One of the well-known conditions is separability condition.
- It states that .... The term was first coined in this work, and have been adapted in many works after that.
- The condition has been exploited to when it comes to algorithm design. Particularly, the problem of finding $\boldsymbol{W}, \boldsymbol{H}$ boils down to finding the set $\mathcal{K}$. The rationale is like this.

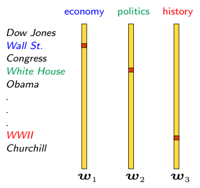
  This notation means a sub-matrix constructed by from columns of $\boldsymbol{X}$ with indices from $\mathcal{K}$. Therefore, knowing $\mathcal{K}$ already reveals $\boldsymbol{W}$. Then finding $\boldsymbol{H}$ becomes a trivial task.
- In terms of application, the condition imposes interesting and reasonable physical interpretation.
  - For example, in topic modeling, it asserts that for each topic, there exists a word that only belong to that topic.
  - In communities, there exists a node that only belongs to a single community.
  - Other similar interpretations are presented in other applications.

Separability Condition

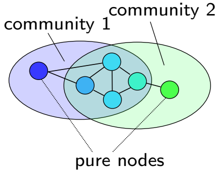
Separability condition [Donoho et al. 2003]

There exists set $\mathcal{K}$ so that $\boldsymbol{H}^\star(:, \mathcal{K}) = \boldsymbol{I}$.

- Have been adapted in many works [Sanjeev Arora et al. 2016; Tsung-Han Chan et al. 2011; Gillis et al. 2014a; Nascimento et al. 2005]
- Finding $\mathcal{K}$ is the key to estimate ground truth $\boldsymbol{W}^\star, \boldsymbol{H}^\star$.
  - In noiseless case, $\boldsymbol{X}(:, \mathcal{K}) = \boldsymbol{W}^\star \boldsymbol{H}^\star(:, \mathcal{K}) = \boldsymbol{W}^\star$.
- Physical interpretation
  - Anchor word [S. Arora et al. 2012] in topic modeling
  - Pure node [Mao et al. 2017b] in community detection



Demonstration of anchor word



Demonstration of pure node

  - Expert annotator in crowd-sourcing [Ibrahim et al. 2019]
  - Pure pixels in hyperspectral unmixing [Ma et al. 2014]

- There have been many formulations developed for problem of finding $\mathcal{K}$ under separability condition. One interesting perspective is from self-diction and sparse regression, as shown in this formulation. The objective function is row-0 norm, which counts number of nonzero rows in $\boldsymbol{C}$.
- The optimal solution of this problem has a particular structure, which is:
  - A subset of rows of $\boldsymbol{C}$ with indices from $\mathcal{K}$ is $\boldsymbol{H}$
  - The other rows are $\mathbf{0}$ rows.

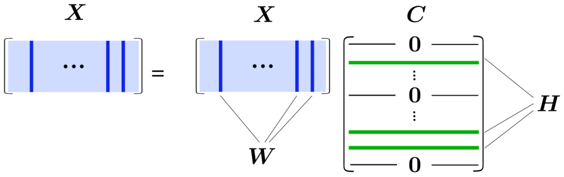
  First of all, [look at picture], we can see that this $\boldsymbol{C}$ satisfies all constraints. Objective value at $\boldsymbol{C}$ is K.
- And for a full rank $\boldsymbol{W}$, one will need at least K non-zero rows from $\boldsymbol{C}$.
- With this structure of $\boldsymbol{C}_{\text{opt}}$, we can easily identify $\mathcal{K}$.

A Self-Dictionary Perspective

- Consider the self-dictionary and sparse regression formulation, [Elhamifar et al. 2012; Esser et al. 2012; Iordache et al. 2014; Recht et al. 2012]

$$\begin{aligned} \underset{\boldsymbol{C}}{\text{minimize}} \quad & \|\boldsymbol{C}\|_{\text{row-0}} \\ \text{subject to} \quad & \boldsymbol{X} = \boldsymbol{X}\boldsymbol{C} \\ & \boldsymbol{C} \geq 0, \mathbf{1}^\top \boldsymbol{C} = \mathbf{1}^\top \end{aligned}$$

- $\boldsymbol{C}_{\text{opt}}(\mathcal{K}, :) = \boldsymbol{H}, \boldsymbol{C}_{\text{opt}}(\mathcal{K}^c, :) = \mathbf{0}$ is an optimal solution point.
  - $\|\boldsymbol{C}_{\text{opt}}\|_{\text{row-0}} = K$.
  - For a full rank $\boldsymbol{W}$, one needs at least K data points to represent $\boldsymbol{X}$.



Row-sparsity matrix $\boldsymbol{C}$

- Solving this optimization problem is not a trivial task. It is firstly not a convex problem due to the row-0, and secondly, and it is a combinatorial problem in essence.
- One of the approaches has been largely studied is greedy approach. As the name suggested, methods in this approach construct set $\mathcal{K}$ by adding 1 index at a time.
- A famous successive projection algorithm (SPA) is a representative of this approach.
- And it has been shown that estimating exact $\mathcal{K}$ is guaranteed, even under noisy condition.
- However, all methods following the greedy approach have a Gram-Schmidt structure in their algorithms. When noise is present, an error made in one iteration will be propagated to future iterations.

Greedy Approach

$$\begin{aligned} \underset{\boldsymbol{C}}{\text{minimize}} \quad & \|\boldsymbol{C}\|_{\text{row-0}} \\ \text{subject to} \quad & \boldsymbol{X} = \boldsymbol{X}\boldsymbol{C} \\ & \boldsymbol{C} \geq 0, \mathbf{1}^\top \boldsymbol{C} = \mathbf{1}^\top \end{aligned}$$

- The greedy approach identifies the set $\mathcal{K}$ by adding one index at a time [Xiao Fu et al. 2015b].
- Successive projection algorithm [Araújo et al. 2001] is a representative.
- Extracting $\mathcal{K}$ is guaranteed even in noisy case [Gillis et al. 2014a].
- All greedy-based methods have a Gram-Schmidt structure which is prone to error propagation under noisy conditions.

- The second approach is called convex relaxation.
- As mentioned before, the original problem is non-convex, which makes it hard in terms of optimisation. So a natural thing to do is to use a convex opt problem as a surrogate. There has been many convex relaxation formulations proposed in the literature.

  One example is this formulation where the objective function comprises of 2 terms: the fitting error and a regularization to promote row-sparsity of $\boldsymbol{C}$.
- Under this formulation, identifiability is guaranteed.
- Since the algorithm does not suffer from error propagation, it is often more robust then the previous approach. We'll see some evidence on this in our experiments.
- However, memory is an obstacle for this approach. Size of variable $\boldsymbol{C}$ is N by N. If it is a dense matrix, memory requirement will grow quadratically.

  As an example, FastGradient is a method following this approach and is considered as state-of-the-art. We run FastGradient on synthetic data and measure its memory consumption. We can see that memory cost grows quadratically to N. This memory cost prohibits this approach's applicability to large scale problem when N could reach to 100000.
- The question is can we do any better?

Convex Relaxation Approach

- Relax the problem to a convex optimization problem [Ammanouil et al. 2014; Elhamifar et al. 2012; Gillis 2013; Gillis et al. 2018, 2014b; Recht et al. 2012]
- An example of this approach is [Esser et al. 2012; Xiao Fu et al. 2015a; Gillis et al. 2018]

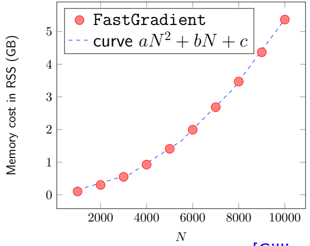
$$\begin{aligned} \underset{\boldsymbol{C}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_{\mathrm{F}}^2 + \lambda R(\boldsymbol{C}) \\ \text{subject to} \quad & \boldsymbol{C} \geq 0, \mathbf{1}^\top \boldsymbol{C} = \mathbf{1}^\top \end{aligned}$$

  where $R(\boldsymbol{C})$ is some regularization term to promote row-sparsity.
- $\mathcal{K}$ is identified based on optimal $\boldsymbol{C}$
- Often more robust than greedy approach

Potential Memory Issue

The variable $\boldsymbol{C}$ has size $N \times N$.

A dense matrix $\boldsymbol{C}$ with $N = 100000$ requires 74.5GB.



Memory consumption of FastGradient [Gillis et al. 2018]

Yes, in the followings, I'll present our proposal on the use of FW. Particularly,

- We follow the convex relaxation approach because of its noise robustness
- We propose using Frank-Wolfe method as the optimization method that can guarantee a memory cost of $O(KN)$.

Proposal: Frank-Wolfe

In order to gain noise robustness and memory efficiency while obtaining identifiability,

- We follow the convex relaxation approach.
- We propose to use Frank-Wolfe as the optimization method to guarantee $O(KN)$ memory consumption.

- To see how FW can realize our goal, let's start with a noiseless case. Consider the following simple optimization problem. This problem is convex, but could have multiple solutions.
- As example a solution $\boldsymbol{C}^{\star}$. It is a desired solution because by inspecting $\boldsymbol{C}^{\star}$, such as examining l1-norm of of the rows, we can expect that the l1-norm is 1 if the corresponding index is from $\mathcal{K}$, 0 otherwise .
- There are other solutions as well, for example, an identity matrix, but it provides no information about $\mathcal{K}$.
- To demonstrate FW's magic, we run Accelerated Proximal Gradient, a typical first order method for constrained optimization problem, and compare it with Frank-Wolfe.
- Firstly, both methods converge and gives a good solutions in terms of objective value. However, if we look at l1-norm in the optimal solution, only FW reveals perfect $\mathcal{K}$.
- Secondly, and more interestingly, if we take a look at the density of $\boldsymbol{C}$ during the optimization produce, we can see that FW consistently keeps $\boldsymbol{C}$ being very sparse, compared to APG. This is the key for memory efficiency when using FW.

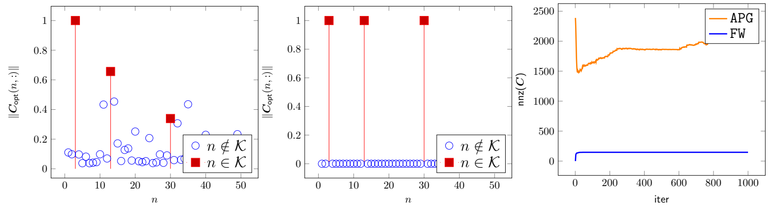
Warm-up with the Noiseless Case

$$\underset{\boldsymbol{C}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_{\mathrm{F}}^2 \tag{2a}$$

$$\text{subject to} \quad \boldsymbol{C} \geq 0, \mathbf{1}^{\top}\boldsymbol{C} = \mathbf{1}^{\top} \tag{2b}$$

Problem (2) can have several solutions

- A desired solution $\boldsymbol{C}^{\star}(\mathcal{K},:) = \boldsymbol{H}, \boldsymbol{C}^{\star}(\mathcal{K}^c,:) = \mathbf{0}$
- A trivial solution $\boldsymbol{I}_N$



APG - Objective value: $3.99e{-5}$ FW - Objective value: $2.86e{-5}$ Number of nonzeros (nnz) of $\boldsymbol{C}$

Accelerated proximal gradient (APG) vs Frank-Wolfe (FW). Unlike APG, FW outputs exact $\boldsymbol{C}^{\star}$ and keeps $\boldsymbol{C}$ sparse during its procedure. $M = 10, N = 50, K = 3$

- Before showing how FW produces such result, let's us briefly review about this method.
- The other name is conditional gradient descent, and has been invented in the 1950s.
- It solves a constraint optimization where the objective f is convex and the constraint $\mathcal{D}$ is a compact convex set.
- A standard update procedure involves 2 steps. The first step involves a sub-problem which can be solved very efficiently for many constraints.
- In our case, the constraint is a probability simplex, and solving it only cost $O(n)$ in terms of computation.

  In detail, the solution $\boldsymbol{s}$ for this step is a canonical unit vector, where the index $n^\star$ is index of the smallest element of the gradient. This observation plays an important role that leads to memory efficiency.

Frank-Wolfe (FW) method [Frank et al. 1956]

- Assume $f(\boldsymbol{x})$ is convex and $\mathcal{D}$ is a compact convex constraint

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{D} \end{aligned}$$

- FW's standard procedure: at iteration t,

$$\begin{aligned} \boldsymbol{s}^t &\leftarrow \underset{\boldsymbol{s}\in\mathcal{D}}{\arg\min}\ \nabla f(\boldsymbol{x}^t)^\top \boldsymbol{s} \\ \boldsymbol{x}^{t+1} &\leftarrow \boldsymbol{x}^t + \alpha^t(\boldsymbol{s}^t - \boldsymbol{x}^t), \quad \alpha^t = 2/(2+t) \end{aligned} \tag{3}$$

- For our problem,

When $\mathcal{D} = \left\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} \geq 0, \mathbf{1}^\top \boldsymbol{x} = 1\right\}$, solving (3) only cost $O(n)$, i.e.,

$$\boldsymbol{s} = \boldsymbol{e}_{n^\star}, \ n^\star = \underset{n}{\arg\min} [\nabla f(\boldsymbol{x}^t)]_n$$

- Get back to our problem, note that the original problem is decomposible and hence we can solve it for each column of $\boldsymbol{C}$ independently. We omit the subscript denoting index of columns for simplification.
- The updating procedure is given by these 2 steps.
- The key observation is that when initialising $\boldsymbol{c}$ at $\mathbf{0}$, if $\boldsymbol{n}^\star \in \mathcal{K}$, then supp$(\boldsymbol{c}^t) \subseteq \mathcal{K}$ for all t. This implies that most part of $\boldsymbol{C}$ are actually just 0, only KN elements in C aren't.
- So the magic is the fact that $n^\star \in \mathcal{K}$ really holds.

FW in the Noiseless Case

- The original problem can be solved for each column $\boldsymbol{c}$ independently.

$$\begin{aligned} \underset{\boldsymbol{c}\in\mathbb{R}^N}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{X}\boldsymbol{c}\|_{\mathrm{F}}^2 := f(\boldsymbol{c}) \\ \text{subject to} \quad & \boldsymbol{c} \geq 0, \mathbf{1}^\top \boldsymbol{c} = 1 \end{aligned}$$

- Updating procedure:

$$\begin{aligned} \boldsymbol{s}^t &\leftarrow \boldsymbol{e}_{n^\star}, \quad n^\star = \arg\min_n \left[\nabla f(\boldsymbol{x}^t)\right] \\ \boldsymbol{c}^{t+1} &\leftarrow \boldsymbol{c}^t + \alpha^t(\boldsymbol{s}^t - \boldsymbol{c}^t), \quad \alpha^t = 2/(2+t) \end{aligned}$$

- If FW picks $n^\star \in \mathcal{K}$ in all iterations, then with $\boldsymbol{c}^0 = \mathbf{0}$,

$$\text{supp}(\boldsymbol{c}^t) \subseteq \mathcal{K}$$

holds in all iterations t until FW terminates.

- Firstly, gradient has this form
- Note that we are looking for index of the smallest element. That index is $n^\star$ if either
  - $\boldsymbol{h}_{n^\star}$ is some canonical unit vector.
  - Or $\boldsymbol{q} = \mathbf{0}$. In this case, we can terminate FW since the desired solution is found.
- To sum up, what we have shown is that with initialization at $\mathbf{0}$, supp($\boldsymbol{c}$) is always a subset of $\mathcal{K}$, and when FW terminates, we get a desired solution $\boldsymbol{c}^\star$. Therefore, we can conclude that FW outputs $\boldsymbol{C}^\star$ using only $O(KN)$ memory.

FW in the Noiseless Case

FW always picks $n^\star \in \mathcal{K}$.

- Gradient

$$\nabla f(\mathbf{c}) = [\boldsymbol{h}_1^\top \boldsymbol{q}, \ldots, \boldsymbol{h}_N^\top \boldsymbol{q}]^\top, \quad \boldsymbol{q} = \boldsymbol{W}^\top \boldsymbol{W}(\boldsymbol{H}\boldsymbol{c} - \boldsymbol{h})$$

- For $n^\star = \arg\min_n \boldsymbol{h}_n^\top \boldsymbol{q}$, either
  - $\boldsymbol{h}_{n^\star} = \boldsymbol{e}_{k^\star}$, where $k^\star = \arg\min_{k \in [K]} q_k$. By definition, $n^\star \in \mathcal{K}$.
  - $\boldsymbol{q} = \mathbf{0} \Rightarrow$ desired solution $\boldsymbol{c}^\star$ is found because,

$$\boldsymbol{q} = \mathbf{0} \Leftrightarrow \boldsymbol{H}\boldsymbol{c} = \boldsymbol{h} \xLeftrightarrow{\text{assume } \mathcal{K} = [K]} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{H}' \end{bmatrix} \boldsymbol{c} = \boldsymbol{h} \Leftrightarrow \boldsymbol{c} = \begin{bmatrix} \boldsymbol{h} \\ \mathbf{0} \end{bmatrix} = \boldsymbol{c}^\star$$

To sum up, in the noiseless case, with $\boldsymbol{c}^0 = \mathbf{0}$,

- supp($\boldsymbol{c}^t$) $\subseteq \mathcal{K}$ for all t.
- FW terminates when $\boldsymbol{c}^t = \boldsymbol{c}^\star = \begin{bmatrix} \boldsymbol{h} \\ \mathbf{0} \end{bmatrix}$.
- Therefore, FW outputs $\boldsymbol{C}_{\text{opt}} = \boldsymbol{C}^\star$ using only $O(KN)$ memory.

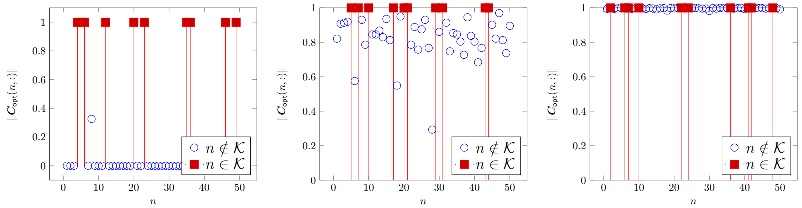
- However, when noise is introduced, the picked index $n^\star$ could be outside of $\mathcal{K}$.
- Thus, FW is no longer guaranteed to output $\boldsymbol{C}^\star$.
- As a simple demonstration, we run FW on 3 cases with increasing noise level. It is evident that when noise is large, the solution obtained by FW could not gives us the desired $\boldsymbol{C}^\star$.

FW in the Noisy Case

- In the noisy case, i.e., $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H} + \boldsymbol{V}, \boldsymbol{V} \neq \boldsymbol{0}$, the gradient is

$$\nabla f(\boldsymbol{c}) = [\boldsymbol{h}_1^\top \boldsymbol{q}, \ldots, \boldsymbol{h}_n^\top \boldsymbol{q}] + \boldsymbol{n}, \quad (\boldsymbol{n} \text{ depends on the noise } \boldsymbol{V})$$

  then the picked index $n^\star$ could be outside of $\mathcal{K}$.
- FW is no longer guaranteed to output $\boldsymbol{C}^\star$.



SNR= 35dB   SNR= 25dB   SNR= 15dB

$\boldsymbol{C}_{\rm opt}$ obtained by FW; $M = 40, N = 50, K = 10$.

- In order to deal with noise, it is common to introduce regularizations. In our problem, the prior is row-sparsity of $\boldsymbol{C}$. There have been many different regularizations used in the literature to promote row-sparsity of $\boldsymbol{C}$
- The mixed-norm l1, l-inf is an example.
- In terms of optimization, FW works best with a smooth function. Therefore, we propose a smooth function to approximate the mix-norm l1, l-inf.
- The hyperparameter μ controls the accuracy of approximating l-inf.
- To this end, we propose to solve the following problem. The objective includes 2 terms: the fitting error and our smooth regularization.
- Given the convex relaxation on the self-dictionary problem with smooth regularization, we propose `MERIT`, a FW-based algorithm for solving:

Enhancement in the Noisy Case

- Different regularizations have been used to promote row-sparsity [Elhamifar et al. 2012; Esser et al. 2012; Xiao Fu et al. 2015a; Gillis et al. 2018, 2014b; Recht et al. 2012]. For example, [Esser et al. 2012; Xiao Fu et al. 2015a] use

$$\|\boldsymbol{C}\|_{\infty,1} := \sum_{i=1}^{N} \|\boldsymbol{C}(i,:)\|_{\infty}$$

- FW works best with smooth functions

$$\Phi_\mu(\boldsymbol{C}) = \sum_{i=1}^{N} \varphi_\mu(\boldsymbol{C}(i,:)), \quad \varphi_\mu(\boldsymbol{C}(i,:)) = \mu \log\left(\frac{1}{N}\sum_{j=1}^{N} \exp\left(\frac{c_{i,j}}{\mu}\right)\right)$$

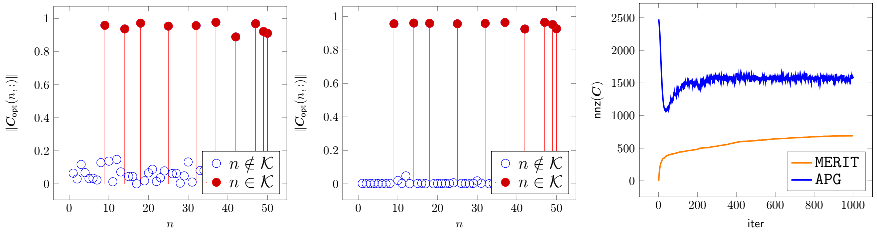
- We propose `MERIT`, a FW-based algorithm for solving:

$$\begin{aligned} \underset{\boldsymbol{C}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_{\mathrm{F}}^2 + \lambda \Phi_\mu(\boldsymbol{C}) \\ \text{subject to} \quad & \boldsymbol{C} \geq 0, \mathbf{1}^\top \boldsymbol{C} = \mathbf{1}^\top \end{aligned}$$

- We first achieve identifiability when working on this formulation. The result uses a similar idea to this work.
- Secondly, it's worth to stress that any convex method can be used to archive identifiability.
- To demonstrate, we run APG and FW on this problem. Both method produce similar optimal objective values, can both optimal solutions can be used to identify $\mathcal{K}$.
- However, it is noticeable that our method, MERIT, constantly keeps $\boldsymbol{C}$ being much more sparse compared to APG. That sparsity is the key of FW's memory efficiency.

Identifiability

- With regularization, we can guarantee the extraction of $\mathcal{K}$ exactly in the noisy case under some reasonable assumptions [Nguyen et al. 2021].
- This result is obtained using a similar idea to [Xiao Fu et al. 2015a].
- Any convex optimization method can be used to obtain $\mathcal{K}$ via solution $\boldsymbol{C}_{\text{opt}}$.



MERIT - Objective value: $3.58e{-}01$ APG - Objective value: $4.41e{-}01$ Number of nonzeros (nnz) of $\boldsymbol{C}$

$M = 40, K = 10, N = 50, \texttt{SNR} = 30\text{dB}, \mu = 1e{-}6, \lambda = 0.1.$

- Let's us examine how FW could save memory in this case.
- Recall the objective function now includes 2 terms: the fitting error and our regularization term.
- Thanks to the updating procedure of FW and also because the constraint being imposing on column-wise, we can perform the update on column by column sequentially. Note that we can run it simultaneously but it would result in $O(N^2)$ memory to store the whole gradient matrix.
- The gradient respect to 1 column of $\boldsymbol{C}$ is given by this.
- The key enables guarantee of memory is that: If at iteration t where we have supp$(\boldsymbol{c}^t) \in \mathcal{K}$, can FW pick $n^\star \in \mathcal{K}$ in the next iteration?
- Before showing why the statement holds, let's see what is the implication. This statement establishes a recursive relation on $\boldsymbol{C}$ between iterations. Hence, if we carefully initilize $\boldsymbol{C}$ to construct a base case suth that supp$(\boldsymbol{c}) \in \mathcal{K}$, then the index in next iteration is from $\mathcal{K}$, and hence we also have supp$(\boldsymbol{c}^{t+1}) \in \mathcal{K}$ holds in the next iteration. That means, the memory cost of saving $\boldsymbol{C}$ is again $O(KN)$ as was shown in noiseless case.
- Gradient is comprised of 2 terms. Let's us go through these terms.

Memory

- The objective function

$$h(\boldsymbol{C}) = \underbrace{\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_\mathrm{F}^2}_{f(\boldsymbol{C})} + \lambda \Phi_\mu(\boldsymbol{C}) = f(\boldsymbol{C}) + \lambda \Phi_\mu(\boldsymbol{C})$$

- FW's updating procedure on this problem can be executed column by column sequentially

$$\boldsymbol{s}_\ell^t \leftarrow \boldsymbol{e}_{n^\star}, \quad n^\star = \arg\min_n \, [\nabla h(\boldsymbol{c}_\ell)]_n$$
$$\boldsymbol{c}_\ell^{t+1} \leftarrow \boldsymbol{c}_\ell^t + \alpha(\boldsymbol{s}_\ell^t - \boldsymbol{c}_\ell^t), \quad \alpha^t = 2/(2+t)$$

- Gradient is given by

$$\nabla h(\boldsymbol{c}_\ell) = \nabla f(\boldsymbol{c}_\ell) + \lambda [\nabla \Phi_\mu(\boldsymbol{C})]_{:,\ell}$$

- Question: If at iteration t, supp$(\boldsymbol{c}_\ell^t) \in \mathcal{K}$, can FW pick

$$n^\star \in \mathcal{K},$$

where $n^\star := \arg\min_n \, [\nabla h(\boldsymbol{c}_\ell)]_n$ in iteration $t+1$?

- The first term is gradient of the fitting error.
- Recall that it is a sum of 2 terms:
- Because of noise, the smallest element which was shown to be inside of $\mathcal{K}$, now be outside of $\mathcal{K}$.

Effect of Noise

- Gradient of $f(\boldsymbol{c}_\ell) = 1/2 \, \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_{\mathrm{F}}^2$

$$\nabla f(\boldsymbol{c}_\ell) = [\boldsymbol{h}_1^\top \boldsymbol{q}_\ell, \ldots, \boldsymbol{h}_N^\top \boldsymbol{q}_\ell]^\top + \boldsymbol{n}_\ell$$

- A demonstration of effect of noise that causes

$$n^\star := \arg\min_n \, [\nabla f(\boldsymbol{c}_\ell)]_n \notin \mathcal{K}.$$

$$\nabla f(\boldsymbol{c}_\ell) = \begin{array}{r} n^\star \dashrightarrow \\ \\ \\ \\ \\ \\ \end{array} \!\! \begin{bmatrix} 0.5 \\ 1.5 \\ 1.5 \\ 2.0 \\ \vdots \\ 1.5 \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \mathcal{K} \\ \\ \\ \\ \end{array} + \begin{bmatrix} 0.5 \\ 1.5 \\ -0.5 \\ 1.0 \\ \vdots \\ -1.0 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 3.0 \\ 1.0 \\ 3.0 \\ \vdots \\ 0.5 \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \mathcal{K} \\ \\ \\ \\ \dashleftarrow n^\star \end{array}$$

- The second term is the gradient of regularization.
- Recall that we are assuming that at iteration t, we're having a "good" $\boldsymbol{c}$, in a sense that $\mathrm{supp}(\boldsymbol{c}^t)$ is a subset of $\mathcal{K}$.
- With such $\boldsymbol{c}$, gradient of regularization, which we denote $\boldsymbol{y}_\ell$ has a special structure. In particular,
  - For index $n \notin \mathcal{K}$, ...
  - For index n_0 such that ...
  - The existence of n_0 can be guarantee by some initialization.

Regularization

- ▶ Gradient of the regularization

$$\boldsymbol{y}_\ell = [\nabla \Phi_\mu(\boldsymbol{C})]_{:,\ell}, \quad y_{n,\ell} = \frac{\exp(c_{n,\ell}/\mu)}{\sum_{i=1}^{N} \exp(c_{n,i}/\mu)}$$

- ▶ Assume that at iteration t, $\mathrm{supp}(\boldsymbol{c}_\ell^t) \subseteq \mathcal{K}$ for all ℓ.
  - ▶ For $n \notin \mathcal{K}$, $y_{n,\ell} = 1/N$.
  - ▶ If $\exists n_0 \in \mathcal{K}$ such that $c_{n_0,\ell}$ is not the largest element in row n_0 (*), then $y_{n_0,\ell} < \exp((c_{n_0,\ell} - c_{n_0,\star})/\mu)$, $\quad c_{n_0,\star} = \max_i c_{n_0,i}$.
  - ▶ (*) can be enforced with some initialization.
- ▶ An example of $\boldsymbol{C}$ and $\boldsymbol{y}_\ell$,

$$\boldsymbol{C} = \begin{array}{cc} & \begin{array}{ccc} \ell & \quad & m \end{array} \\ \mathcal{K} & \begin{bmatrix} 0.3 & \dots & 0.1 \\ 0.2 & \dots & 0.6 \\ 0.5 & \dots & 0.3 \\ & \vdots & \\ \mathbf{0} & & \mathbf{0} \end{bmatrix} \begin{array}{c} \\ \leftarrow n_0 \\ \\ \\ \\ \end{array} \end{array} \implies \boldsymbol{y}_\ell = \begin{bmatrix} 0.5 \\ 0.001 \\ 0.9 \\ 1/N \\ \vdots \\ 1/N \end{bmatrix} \begin{array}{c} \\ \leftarrow n_0 \\ \\ \\ \\ \\ \end{array}$$

- Now we know that the gradient of regularization $\boldsymbol{y}_\ell$ has a special structure, such that ...
- That would lead to the picked $n^\star$ being inside $\mathcal{K}$ for some λ.

Effect of Regularization

Regularization can ensure $n^\star \in \mathcal{K}$ under some reasonable assumptions.

- Gradient

$$\nabla h(\boldsymbol{c}_\ell) = \nabla f(\boldsymbol{c}_\ell) + \lambda \boldsymbol{y}_\ell,$$

- We have

$$\begin{cases} y_{n,\ell} = 1/N & \text{if } n \notin \mathcal{K} \\ y_{n_0,\ell} \approx 0 & \text{for some } n_0 \in \mathcal{K} \end{cases}$$

$$\Rightarrow n^\star := \arg\min_n \, [\nabla h(\boldsymbol{c}_\ell)]_n = n_0 \quad \text{for some } \lambda$$

- An example of $\boldsymbol{C}$ and $\nabla h(\boldsymbol{c}_\ell)$,

$$\Longrightarrow \nabla h(\boldsymbol{c}_\ell) = \begin{array}{r} \mathcal{K}\left\{\begin{array}{l} \\ \\ \\ \end{array}\right. \\ \\ \\ \\ \end{array} \begin{bmatrix} 1.0 \\ 3.0 \\ 1.0 \\ 3.0 \\ \vdots \\ 0.5 \end{bmatrix} + \lambda \begin{bmatrix} 0.5 \\ 0.001 \\ 0.9 \\ 1/N \\ \vdots \\ 1/N \end{bmatrix}$$

the smallest element

`MERIT` in Noisy Case

To sum up, in the noisy case, under some reasonable assumptions, the proposed method `MERIT` can

- Extract $\mathcal{K}$ exactly.
- If $\boldsymbol{C}^t$ satisfies $\text{supp}(\boldsymbol{c}_\ell^t) \subseteq \mathcal{K}$ for all ℓ, then $\text{supp}(\boldsymbol{c}_\ell^{t+1}) \subseteq \mathcal{K}$ for all ℓ, and hence `MERIT` can guarantee a memory consumption of $O(KN)$.

- We first evaluate performance of the proposed method MERIT using synthetic settings.
- For identifiability evaluation, we measure the probability of exactly recovering $\mathcal{K}$.
- We compare MERIT with other 2 methods: SPA which is a very well-known representative from greedy approach, and FastGradient which is considered state-of-the-art method using convex relaxation approach.
- The first figure shows result on success rate. The 2 methods, MERIT and FastGradient are more noise reluctant compared to SPA.
- Within MERIT and FastGradient, we also measured their memory consumption. It can be seen in figure b that memory of FastGradient grows much faster compared to MERIT.

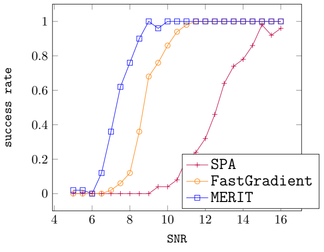
Synthetic Data

Data generation

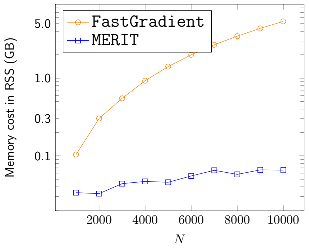
- $\boldsymbol{W} \sim \mathcal{U}(0,1)$
- $\boldsymbol{H} \sim \text{Dir}(\mathbf{1}), \boldsymbol{H}(:,1:K) = \boldsymbol{I}$
- $\boldsymbol{V} \sim \mathcal{N}(0,\sigma)$
- After shuffling $\boldsymbol{H}$, $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H} + \boldsymbol{V}$
- Noise level is measured in SNR $= 10\log_{10}(\sum_{\ell=1}^{N} \|\boldsymbol{W}\boldsymbol{h}_\ell\|_2^2)/(MN\sigma^2)$dB

Metric

- success rate $= P(\mathcal{K} = \widehat{\mathcal{K}})$
- Estimate success rate by 50 trials



(a) success rate under different SNRs; $N = 200$, $M = 50$, $K = 40$.



(b) Memory consumption under different N's; SNR = 10dB, $M = 50$, $K = 40$.

- In the second experiment, we evaluate MERIT on a topic modeling problem, using 2 real-world datasets: TDT2, Reuters.
- Since we have ground truth of topic labels on both dataset, we report Accuracy for performance comparison.
- The set of chosen baselines includes several representatives: SPA as a representative of greedy method, FastAnchor, XRAY are variants of SPA developed under topic modeling problem, LDA is a standard baseline for topic modeling. Lastly, FastGradient is method from convex relaxation is included as in the last exp.
- MERIT has a consistently good performance in both dataset compared to other baselines.

Real Data: Topic Modeling

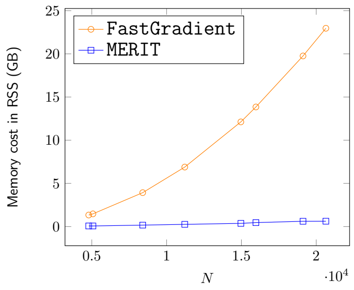
Accuracy

	Method $\backslash K$	3	4	5	6	7	8	9	10
TDT2	SPA	0.87	0.83	0.81	0.81	0.78	0.76	0.75	0.72
	FastAnchor	0.77	0.72	0.67	0.63	0.66	0.63	0.65	0.65
	XRAY	0.87	0.82	0.80	0.81	0.78	0.75	0.75	0.71
	LDA	0.78	0.77	0.74	0.75	0.73	0.72	0.68	0.70
	FastGradient	0.70	0.71	0.65	0.64	0.61	0.56	0.58	0.57
	MERIT	**0.88**	**0.88**	**0.85**	**0.86**	**0.84**	**0.82**	**0.80**	**0.77**
Reuters-21578	SPA	0.64	0.57	0.54	0.51	0.49	0.44	0.42	0.40
	FastAnchor	0.60	0.57	0.52	0.52	0.46	0.42	0.38	0.37
	XRAY	0.63	0.57	0.54	0.51	0.49	0.45	0.42	0.40
	LDA	0.63	0.57	0.53	0.51	0.46	0.44	0.41	0.42
	FastGradient	0.62	0.57	**0.56**	0.51	0.50	**0.48**	**0.44**	**0.46**
	MERIT	**0.66**	**0.62**	0.53	**0.53**	**0.51**	**0.48**	0.43	0.45

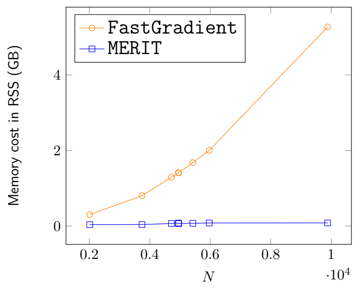
Bold, and blue indicate the best and second best scores, resp.

Real Data: Topic Modeling

- Moreover, we also measure and compare memory consumption between FastGradient and MERIT.
- Similar observation to the previous exp, amount of memory used by FastGradient is growing much faster than MERIT when N is increasing.



TDT2



Reuters-21578

Memory consumption of `FastGradient` and `MERIT`

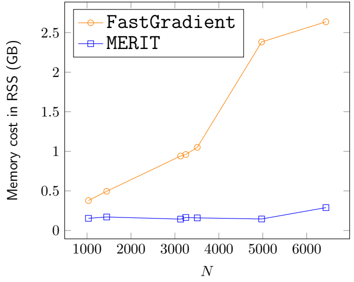
- In the last exp, we evaluate MERIT on a community detection problem using 2 common used dataset: BDLP and MAG.
- Since they both provide label on community membership, we compare performance based on SRC. Higher score means a better alignment between prediction and ground truth.
- For baseslines, we compare with GeoNMF and SPOC, both are popular greedy method for community detection problem, and FastGradient is also included as before.
- As we can see, MERIT has a good performance as it was in top 2 in 5 out of 7 cases.
- In terms of memory consumption, we observed a consistent result with previous experiment: FastGradient's memory grows much faster than MERIT.

Real Data: Community detection

- Metric: Spearman's rank correlation (SRC). SRC $\in [-1, 1]$, higher value is better.
- Data: co-authorship networks, a community ground truth is defined by
  - DBLP: group of conferences
  - MAG: "field of study" tag

Dataset	GeoNMF	SPOC	FastGradient	MERIT
DBLP1	0.2974	0.2996	**0.3145**	0.2937
DBLP2	0.2948	0.2126	0.3237	**0.3257**
DBLP3	0.2629	**0.2972**	0.1933	0.2763
DBLP4	0.2661	0.3479	0.1601	**0.3559**
DBLP5	0.1977	0.1720	0.0912	**0.1983**
MAG1	**0.1349**	0.1173	0.0441	0.1149
MAG2	0.1451	0.1531	**0.2426**	0.2414

SRC Performance on DBLP and MAG. **Bold** and blue indicate the best and second best scores.



Memory consumption of FastGradient and MERIT

Conclusion

- FW is proposed as a memory efficient method for solving separable simplex-structured matrix factorization via convex relaxation.
- When noise is absent, using FW can bring identification with memory $O(KN)$
- For the noisy case, we have proposed using a smooth regularization to guarantee identifiability.
- For the noisy case, we have also shown that running FW only cost $O(KN)$ using some initialization strategy.

The talk is based on [Tri Nguyen et al. "Memory-efficient convex optimization for self-dictionary separable nonnegative matrix factorization: A frank-wolfe approach". In: *arXiv preprint arXiv:2109.11135* [2021], IEEE TSP, revised. (2nd round revision).]

Reference I

[1] Edoardo M Airoldi et al. "Mixed membership stochastic blockmodels". In: *Journal of Machine Learning Research* 9.Sep (2008), pp. 1981–2014.

[2] R. Ammanouil et al. "Blind and Fully Constrained Unmixing of Hyperspectral Images". In: *IEEE Trans. Image Process.* 23.12 (Dec. 2014), pp. 5510–5518.

[3] U.M.C. Araújo et al. "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis". In: *Chemometrics and Intelligent Laboratory Systems* 57.2 (2001), pp. 65–73.

[4] S. Arora et al. "Learning topic models–going beyond SVD". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. 2012, pp. 1–10.

[5] Sanjeev Arora et al. "A practical algorithm for topic modeling with provable guarantees". In: *International Conference on Machine Learning*. 2013, pp. 280–288.

Reference II

[6] Sanjeev Arora et al. "Computing a Nonnegative Matrix Factorization—Provably". In: *SIAM Journal on Computing* 45.4 (2016), pp. 1582–1611. DOI: 10.1137/130913869.

[7] T.-H. Chan et al. "A Convex Analysis Framework for Blind Separation of Non-Negative Sources". In: *IEEE Trans. Signal Process.* 56.10 (Oct. 2008), pp. 5120–5134.

[8] Tsung-Han Chan et al. "A simplex volume maximization framework for hyperspectral endmember extraction". In: *IEEE Trans. Geosci. Remote Sens.* 49.11 (2011), pp. 4177–4193.

[9] D. Donoho et al. "When does non-negative matrix factorization give a correct decomposition into parts?" In: *Advances in Neural Information Processing Systems*. Vol. 16. 2003, pp. 1141–1148.

[10] E. Elhamifar et al. "See all by looking at a few: Sparse modeling for finding representative objects". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 1600–1607.

Reference III

[11] Ernie Esser et al. "A convex model for nonnegative matrix factorization and dimensionality reduction on physical space". In: *IEEE Trans. Image Process.* 21.7 (2012), pp. 3239–3252.

[12] Marguerite Frank et al. "An algorithm for quadratic programming". In: *Naval Research Logistics Quarterly* 3.1-2 (1956), pp. 95–110.

[13] X. Fu et al. "Robust Volume Minimization-Based Matrix Factorization for Remote Sensing and Document Clustering". In: *IEEE Trans. Signal Process.* 64.23 (Dec. 2016).

[14] Xiao Fu et al. "Nonnegative Matrix Factorization for Signal and Data Analytics: Identifiability, Algorithms, and Applications". In: *IEEE Signal Process. Mag.* 36.2 (Mar. 2019), pp. 59–80.

[15] Xiao Fu et al. "Robustness analysis of structured matrix factorization via self-dictionary mixed-norm optimization". In: *IEEE Signal Processing Letters* 23.1 (2015), pp. 60–64.

[16] Xiao Fu et al. "Self-Dictionary Sparse Regression for Hyperspectral Unmixing: Greedy Pursuit and Pure Pixel Search are Related". In: *IEEE J. Sel. Topics Signal Process.* 9.6 (2015), pp. 1128–1141.

Reference IV

[17] Nicolas Gillis. "Robustness analysis of hottopixx, a linear programming model for factoring nonnegative matrices". In: *SIAM Journal on Matrix Analysis and Applications* 34.3 (2013), pp. 1189–1212.

[18] Nicolas Gillis et al. "A fast gradient method for nonnegative sparse regression with self-dictionary". In: *IEEE Trans. Image Process.* 27.1 (2018), pp. 24–37.

[19] Nicolas Gillis et al. "Fast and robust recursive algorithms for separable nonnegative matrix factorization". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 36.4 (2014), pp. 698–714.

[20] Nicolas Gillis et al. "Robust near-separable nonnegative matrix factorization using linear optimization". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1249–1280.

[21] Kejun Huang et al. "Anchor-free correlated topic modeling: Identifiability and algorithm". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1786–1794.

Reference V

[22] Kejun Huang et al. “Detecting Overlapping and Correlated Communities without Pure Nodes: Identifiability and Algorithm”. In: *International Conference on Machine Learning*. Sept. 2019, pp. 2859–2868.

[23] Shahana Ibrahim et al. “Crowdsourcing via Pairwise Co-occurrences: Identifiability and Algorithms”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 7847–7857.

[24] Marian Daniel Iordache et al. “Collaborative sparse regression for hyperspectral unmixing”. In: *IEEE Trans. Geosci. Remote Sens.* 52.1 (2014), pp. 341–354.

[25] Nirmal Keshava et al. “Spectral unmixing”. In: *IEEE signal processing magazine* 19.1 (2002), pp. 44–57.

[26] Jing Lei et al. “Consistency of spectral clustering in stochastic block models”. In: *The Annals of Statistics* 43.1 (2015), pp. 215–237.

Reference VI

[27] Wing-Kin Ma et al. "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing". In: *IEEE Signal Process. Mag.* 31.1 (2014), pp. 67–81.

[28] Xueyu Mao et al. "Estimating Mixed Memberships with Sharp Eigenvector Deviations". In: *arXiv* (2017), pp. 1–46. ISSN: 23318422. arXiv: 1709.00407.

[29] Xueyu Mao et al. "On Mixed Memberships and Symmetric Nonnegative Matrix Factorizations". In: *International Conference on Machine Learning*. 2017, pp. 2324–2333.

[30] José MP Nascimento et al. "Vertex component analysis: A fast algorithm to unmix hyperspectral data". In: *IEEE Trans. Geosci. Remote Sens.* 43.4 (2005), pp. 898–910.

[31] Tri Nguyen et al. "Memory-efficient convex optimization for self-dictionary separable nonnegative matrix factorization: A frank-wolfe approach". In: *arXiv preprint arXiv:2109.11135* (2021).

Reference VII

[32] Maxim Panov et al. "Consistent estimation of mixed memberships with successive projections". In: *International Workshop on Complex Networks and their Applications* (2017), pp. 53–64.

[33] Ben Recht et al. "Factoring nonnegative matrices with linear programs". In: *Advances in Neural Information Processing Systems*. 2012, pp. 1214–1222.

Condition (*)

Claim:

$\exists n_0 \in \mathcal{K}$ such that $\boldsymbol{C}(n_0, :)$ is not a constant
$\Rightarrow \exists n_0 \in \mathcal{K}$ such that $c_{n_0,\ell}$ is not the largest element in row n_0 (*).

- Assume that for all $n \in \mathcal{K}$, $c_{n,\ell}$ is the largest element in row n.
- Then for row n_0 such that $\boldsymbol{C}(n_0, :)$ is not a constant,

$$\exists m, \quad c_{n_0,\ell} > c_{n_0,m}$$

- That leads to

$$1 = \mathbf{1}^\top \boldsymbol{c}_\ell > \mathbf{1}^\top \boldsymbol{c}_m = 1$$

- The contradiction concludes our claim.

An example of $\boldsymbol{C}$,

$$\boldsymbol{C} = \begin{array}{cc} & \begin{array}{ccc} \overbrace{}^{\ell} & & \overbrace{}^{m} \end{array} \\ \mathcal{K}\left\{ \phantom{\begin{array}{c} \vdots \\ 0.6 \\ \vdots \end{array}} \right. & \begin{bmatrix} \vdots & \cdots & \vdots \\ \boxed{0.6 \quad \cdots \quad 0.5} & & \\ \vdots & \cdots & \vdots \\ \mathbf{0} & \vdots & \mathbf{0} \end{bmatrix} \end{array}$$