

# Sum-of-norms regularized Nonnegative Matrix Factorization \*

<sup>1</sup>Andersen Ang      Waqas Bin Hamed      <sup>2</sup>Hans De Sterck

<sup>1</sup>School of Electronics and Computer Science, University of Southampton, United Kingdom

<sup>2</sup>Department of Applied Mathematics, University of Waterloo, Canada

July 2, 2024

## Abstract

When applying nonnegative matrix factorization (NMF), generally the rank parameter is unknown. Such rank in NMF, called the nonnegative rank, is usually estimated heuristically since computing the exact value of it is NP-hard. In this work, we propose an approximation method to estimate such rank while solving NMF on-the-fly. We use sum-of-norm (SON), a group-lasso structure that encourages pairwise similarity, to reduce the rank of a factor matrix where the rank is overestimated at the beginning. On various datasets, SON-NMF is able to reveal the correct nonnegative rank of the data without any prior knowledge nor tuning.

SON-NMF is a nonconvex nonsmooth non-separable non-proximable problem, solving it is nontrivial. First, as rank estimation in NMF is NP-hard, the proposed approach does not enjoy a lower computational complexity. Using a graph-theoretic argument, we prove that the complexity of the SON-NMF is almost irreducible. Second, the per-iteration cost of any algorithm solving SON-NMF is possibly high, which motivated us to propose a first-order BCD algorithm to approximately solve SON-NMF with a low per-iteration cost, in which we do so by the proximal average operator. Lastly, we propose a simple greedy method for post-processing.

SON-NMF exhibits favourable features for applications. Beside the ability to automatically estimate the rank from data, SON-NMF can deal with rank-deficient data matrix, can detect weak component with small energy. Furthermore, on the application of hyperspectral imaging, SON-NMF handle the issue of spectral variability naturally.

**Keywords:** nonnegative matrix factorization, rank, regularization, sum-of-norms, nonsmooth nonconvex optimization, algorithm, proximal gradient, proximal average, complete graph

## 1 Introduction

**Nonnegative Matrix Factorization (NMF)** We denote  $\text{NMF}(\mathbf{M}, r)$  [1, 2] the following problem: given a matrix  $\mathbf{M} \in \mathbb{R}_+^{m \times n}$ , find two factor matrices  $\mathbf{W} \in \mathbb{R}_+^{m \times r}$  and  $\mathbf{H} \in \mathbb{R}_+^{r \times n}$  such that  $\mathbf{M} = \mathbf{W}\mathbf{H}$ . NMF describes a cone:  $\mathbf{M}$  is a point cloud (of  $n$  points) in  $\mathbb{R}_+^m$ , contained in a polyhedral cone generated by the  $r$  columns of  $\mathbf{W}$  with nonnegative weights encoded in  $\mathbf{H}$ , where  $H_{ij}$  represents the contribution of column  $\mathbf{w}_i$  in representing the data column  $\mathbf{m}_j$ , e.g., see [3, Fig.1].

**Nonnegative rank** Let  $r = \text{rank}_+(\mathbf{M})$  denotes the nonnegative-rank of a matrix, where  $r$  represents the minimal number of nonnegative rank-1 components required to represent  $\mathbf{M}$  [4, Section 4], [2, Section 3], i.e.,

$$\text{NMF}(\mathbf{M}, r) : \mathbf{M} = \mathbf{W}\mathbf{H} = [\mathbf{w}_1 \dots \mathbf{w}_r] \begin{bmatrix} \mathbf{h}^1 \\ \vdots \\ \mathbf{h}^r \end{bmatrix} = \mathbf{w}_1\mathbf{h}^1 + \dots + \mathbf{w}_r\mathbf{h}^r = \sum_{\ell=1}^r \mathbf{w}_\ell\mathbf{h}^\ell, \quad \mathbf{w}_\ell \geq \mathbf{0}, \mathbf{h}^\ell \geq \mathbf{0}, \quad (1)$$

where  $\mathbf{w}_j$  is the  $j$ th column of  $\mathbf{W}$ , and  $\mathbf{h}^j$  is the  $j$ th row of  $\mathbf{H}$ . Here  $\mathbf{w}_j\mathbf{h}^j$  is the  $j$ th rank-1 factor in  $\mathbf{W}\mathbf{H}$ .

**$r$  is important** Parameter  $r$  controls the model complexity of NMF and plays a critical role in data analysis. In signal processing [5],  $r$  represents the number of sources in a audio. If  $r$  is over-estimated, over-fitting occurs where the over-estimated component in the models the noise (e.g. piano mechanical noise [6, Section 4.2]) instead of meaningful information.

\*Andersen Ang ([andersen.ang@soton.ac.uk](mailto:andersen.ang@soton.ac.uk)) is the corresponding author. Part of the work of this paper was done when Andersen Ang was a post-doctoral fellow and when Waqas Bin Hamed was a master student, both at the University of Waterloo. Funding: Andersen Ang acknowledge the supported in part by a joint postdoctoral fellowship by the Fields Institute for Research in Mathematical Sciences and the University of Waterloo, and in part by Discovery Grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

**$r$  is unknown** Generally  $r$  is unknown, finding  $r$  in  $\text{NMF}(\mathbf{M}, r)$  for  $\text{rank}_+(\mathbf{M}) \geq 3$  is NP-hard [7]<sup>1</sup>. In many cases  $\text{rank}(\mathbf{M})$  and/or  $\text{rank}_+(\mathbf{M})$  are small since  $\mathbf{M}$  is approximately low rank [8] and/or low nonnegative-rank [2, Section 9.2]. Many heuristics have been proposed to find  $r$  in the literature: beside trial-and-error, the two main groups of methods for finding  $r$  are stochastic/information-theoretic and algebraic/deterministic. The first group includes Bayesian method [9], cophenetic correlation coefficient [10] and minimum description length [11]. The second group includes fooling set [12] and  $f$ -vector in combinatorics [13]. See [2, Section 3] for a summary on the algebra of  $\text{rank}_+$ .

In this work, we focus on approximately solving  $\text{NMF}(\mathbf{M}, r)$ , without tuning nor knowing  $r$  in advance. This is achieved by imposing a “rank penalty” on NMF. Instead of using the nuclear norm nor the rank itself as a penalty term, we consider a clustering regularizer called Sum-of-norms (SON): we propose SON-NMF to “relax” the assumption of knowing  $r$ . Before we introduce SON-NMF, we first review the SON term.

**Matrix  $\ell_{p,q}$ -norm** The  $\ell_{p,q}$ -norm of a matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$  is defined as

$$\|\mathbf{X}\|_{p,q} := \left( \sum_{j=1}^n \left( \sqrt[p]{\sum_{i=1}^m X_{ij}^p} \right)^q \right)^{\frac{1}{q}} = \left\| \begin{bmatrix} \|\mathbf{x}_1\|_p \\ \vdots \\ \|\mathbf{x}_n\|_p \end{bmatrix} \right\|_q,$$

where in the last equality we take the  $p$ -norm on columns followed by taking  $q$ -norm on the resulting vector. A popular choice of the  $\ell_{p,q}$ -norm is  $\ell_{2,1}$ -norm, used in multiple measurement vector problem [14], sparse coding [15] and robust NMF [16].

**Sum-of-norms (SON)** We define the SON of a matrix  $\mathbf{X}$  as the  $\ell_{2,1}$ -norm of  $P(\mathbf{X})$ , where  $\mathbf{X} \mapsto P(\mathbf{X})$  is all the pairwise difference  $\mathbf{x}_i - \mathbf{x}_j$ . As  $\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \|\mathbf{x}_j - \mathbf{x}_i\|_2$ , there are  $\frac{n^2-n}{2}$  terms in SON of  $\mathbf{X}$ . In this work we propose to use  $\text{SON}_{2,1}(\mathbf{W})$  as a regularizer for the NMF, to be presented in the next section. Below we give remarks on  $\text{SON}_{2,q}(\mathbf{W})$  with other choices of  $q$ .

- $\text{SON}_{2,0}(\mathbf{W})$  with  $q = 0$ : it is trivial that  $\text{rank}(\mathbf{W}) \leq \text{SON}_{2,0}(\mathbf{W})$  because the set of linearly independent vectors is a subset of the set of unequal pair of vectors. Next, by the combinatorial nature of  $\ell_0$ -norm, minimizing  $\text{SON}_{2,0}(\mathbf{W})$  is NP-hard and its complexity scales with  $r$ , so  $\text{SON}_{2,0}(\mathbf{W})$  is computationally unfavourable to NMF for applications with a large  $r \approx (m, n)$ , which is the case in this work.
- $\text{SON}_{2,2}(\mathbf{W})$  with  $q = 2$ : it is the Frobenius norm of  $P(\mathbf{W})$  by definition. This SON has been used in graph-regularized NMF [17], which is different from (SON-NMF) for two reasons: 1. the graph regularizer is a weighted-squared-SON<sub>2,2</sub> norm which is everywhere differentiable, which is not the case for  $\text{SON}_{2,1}(\mathbf{W})$ , and 2.  $\text{SON}_{2,2}(\mathbf{W})$  does not induce sparsity while  $\text{SON}_{2,1}(\mathbf{W})$  does.
- $\text{SON}_{2,\infty}(\mathbf{W})$  with  $q \rightarrow \infty$ : this term focuses on the pair  $(\mathbf{w}_i, \mathbf{w}_j)$  that is mutually furthest away from each other, and ignoring the rest. This is unfavourable for removing the redundant  $\mathbf{w}_j$  in NMF for this work.

We are now ready to introduce SON-NMF.

**SON-NMF** In this work we propose to regularize NMF by  $\text{SON}_{2,1}(\mathbf{W}) = \|P(\mathbf{W})\|_{2,1} = \sum_{i \neq j} \|\mathbf{w}_i - \mathbf{w}_j\|_2$ :

$$\underset{\mathbf{W}, \mathbf{H}}{\operatorname{argmin}} F(\mathbf{W}, \mathbf{H}) := \frac{1}{2} \|\mathbf{W}\mathbf{H} - \mathbf{M}\|_F^2 + \lambda \sum_{i \neq j} \|\mathbf{w}_i - \mathbf{w}_j\|_2 + \gamma \sum_i \|\max\{-\mathbf{w}_i, \mathbf{0}\}\|_1 + \iota_{\Delta^r}(\mathbf{H}), \quad (\text{SON-NMF})$$

where  $\frac{1}{2} \|\mathbf{M} - \mathbf{W}\mathbf{H}\|_F^2 : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n} \rightarrow \mathbb{R}$  is a smooth nonconvex data fitting term, the constants  $\lambda > 0$  and  $\gamma > 0$  are parameters, the functions  $\sum_i \|\max\{-\mathbf{w}_i, \mathbf{0}\}\|_1$  and  $\iota_{\Delta^r}(\mathbf{H}) = \sum_j \iota_{\Delta^r}(\mathbf{h}_j)$  are nonsmooth lower-semicontinuous proper convex that represent model constraints: respectively the nonnegativity of  $\mathbf{w}_j$  (i.e.,  $\mathbf{W} \geq \mathbf{0}$ ) and  $\mathbf{h}_j$  is inside  $r$ -dimensional unit simplex (i.e.,  $\mathbf{H}$  is element-wise nonnegative and  $\mathbf{H}^\top \mathbf{1}_r \leq \mathbf{1}_n$  where  $\mathbf{1}_r \in \mathbb{R}^r$  denotes vector of ones). Note that in (SON-NMF) we use the penalty  $\sum_i \|\max\{-\mathbf{w}_i, \mathbf{0}\}\|_1$ , which is equivalent to the nonnegativity constraint  $\mathbf{W} \geq \mathbf{0}$  for sufficiently large  $\lambda$ , to be explained in section 4. We defer to the end of this section for the definition of symbols used in (SON-NMF).

<sup>1</sup>Note that  $\text{rank}_+(\mathbf{M})$  is not the same as  $\text{rank}(\mathbf{M})$ , which can be computed by eigendecomposition or singular value decomposition. See [2] on solving the problem  $\text{NMF}(\mathbf{M}, r)$  for the case  $\text{rank}_+(\mathbf{M}) \leq 2$ .

**Interpretation of SON: encouraging multicollinearity and rank-deficiency for NMF** The SON term encourages the pairwise difference in  $\|\mathbf{w}_i - \mathbf{w}_j\|_2$  to be small, possibly resulting in multicollinearity in the matrix  $\mathbf{W}$ . Note that in traditional regression models, multicollinearity is strongly discouraged due to its negative statistical effect on the variables [18]. In this work, we intentionally encourage the multicollinearity of  $\mathbf{W}$ , for the sake of encouraging rank deficiency in  $\mathbf{W}$  in order to reduce an overestimated rank for rank-estimation. I.e., SON-NMF can be seen as the ordinary NMF model under a multicollinearity regularizer where the rank of  $\mathbf{W}$  at the first iteration is overestimated and then it is the job of the regularizer to reduce the overestimated rank of  $\mathbf{W}$  to the correct value in the algorithmic process.

There is a “price to pay” for such multicollinearity. If  $\mathbf{W}$  is near-multicollinear, the conditional number of  $\mathbf{W}$  is large so  $\mathbf{W}^\top \mathbf{W}$  is ill-conditioned, negatively impacting the process of updating  $\mathbf{H}$ . See the discussion in Section 3.

**Contributions** We introduce a new problem (SON-NMF) with the following contributions.

- **Empirically rank-revealing.** On synthetic and real-world datasets, we empirically show that model (SON-NMF), free from tuning the rank  $r$ , will itself find the correct  $r$  in the data automatically when  $r$  is overestimated. This is due to the sparsity-inducing property of the  $\ell_{2,1}$  norm in  $\text{SON}_{2,1}$ .
- **Rank-deficient compatibility.** SON-NMF can work with rank-deficient problem, i.e., on data matrix with the true rank smaller than the over overestimated parameter  $r$ . This has two advantages. First, it means the model prevents over-fitting. Second, compared with existing NMF models such as the minimum-volume NMF [19, 5] (see below) which was shown to exhibit [3] rank-finding ability, SON-NMF is applicable to rank-deficient matrix.
- **Irreducible computational complexity.** As computing  $\text{rank}_+$  is NP-hard, the SON approach, as a “work-around” approach to estimate  $\text{rank}_+$ , cannot enjoy a lower complexity. We prove that (Theorem 1) the complexity of the SON term is *almost irreducible*. Precisely, we show that in the best case, to recover the  $r^*$  columns of the true  $\mathbf{W}^*$  using  $\mathbf{W}$  obtained from SON-NMF with a rank  $r > r^*$ , we cannot reduce the complexity of the SON term from  $r(r-1)/2$  to below  $r(r-\lceil r/r^* \rceil)/2$ .
- **Fast algorithm by proximal-average.** Solving (SON-NMF) is not trivial: the  $\mathbf{W}$ -subproblem is nonsmooth non-separable and non-proximable, meaning that proximal-based methods [20, 21, 22, 23, 24] cannot efficiently solve the problem. When dealing with non-proximal problem, dual approach like Lagrange multiplier and ADMM are usually used, however SON-NMF has  $\mathcal{O}(r^2)$  pairs of non-proximal terms and such complexity is irreducible (Theorem 1), the dual methods and second-order methods are inefficient since they have a very high per-iteration cost. We propose a low-cost proximal average [25] based on the Moreau-Yosida envelop [26].

We review the literature in the next paragraphs, on the background and the motivation of this work.

**Review of NMF: minimum-volume and rank-deficiency** SON-NMF has linkage to minvol NMF [19, 27]. Recently it has been observed in [3] that when using volume regularization in the form of  $\log \det(\mathbf{W}^\top \mathbf{W} + \delta \mathbf{I}_r)$ , minvol NMF on rank deficient matrix  $\mathbf{M}$  (i.e., overestimating the  $r$  parameter) has the ability to zeroing out extra components in  $\mathbf{W}, \mathbf{H}$ . This has also been observed in audio blind source separation [5], where a rank-7 factorization is used on a dataset with 3 sources, the minvol NMF is able to set the redundant components to zero. I.e., minvol NMF can automatically select the model order  $r$ . However minvol NMF is not suitable for rank-deficient  $\mathbf{W}$ : we have  $\log \det(\mathbf{W}^\top \mathbf{W}) = \log 0 = -\infty$  if  $\delta = 0$ . Even if  $\delta \neq 0$ , the rank-deficient  $\mathbf{W}$  provide no information in the logdet term. Furthermore, in the work [5] on using an overestimated rank in minvol NMF, it is the redundant components in matrix  $\mathbf{H}$  set to zero instead of  $\mathbf{W}$ . We remark that it is the rank-revealing observation of minvol NMF motivated the first author to propose SON-NMF.

**Review of clustering** SON was proposed in [28, 29] on clustering. Due to the interpretation that minimizing  $\text{SON}(\mathbf{W})$  will force the pairwise difference  $\mathbf{w}_i - \mathbf{w}_j$  to be small, SON is also called “fusion penalty” [30]. Later [31] considered SON with  $0 < p < 1$ , and recently [32] showed that SON clustering can provably recover the Gaussian mixture under some assumptions.  $\text{SON}_{2,0}$  is also used in graph trend filtering [33]. We remark that these works are different from SON-NMF: they are single-variable problem, and NMF is a bi-variate nonconvex problem with nonnegativity constraints.

**SON solution approaches** The approach we proposed to solve the SON problem is different from the existing approaches such as quadratic programming with convex hull [28], active-set [30], interior-point method [29], trust-region with smoothing [31], Lagrange multiplier [34, 12.3.8] and semi-smooth Newton’s method [35]. These approaches are all proposed for single-variable clustering (i.e.,  $\mathbf{W}$  only) with no nonnegativity constraint. What we proposed is to makes use of proximal average [26, 25] which is computationally cheap to compute (with a per-iteration cost  $\mathcal{O}(m)$  where  $m$  is the dimension of  $\mathbf{w}_j$ ) for SON and thus lowering the per-iteration cost. All the method mentioned above are either unable to solve the SON problem on  $\mathbf{W}$  with nonnegativity, or having a higher per-iteration cost. See details in section 4.

**History: the geometric median and the Fermat-Torricelli-Weber problem and** Although SON is proposed in 2000s [28, 30, 29], it is closely related to an old problem known as the Fermat-Torricelli-Weber problem [36, 37], [34, Example 3.66], also known as the geometric median. We note that the analysis of geometric median does not apply to SON-NMF, but it provides a geometric interpretation: SON-NMF produces a  $r^*$ -cluster of points with the smallest geometric median to the dataset.

**Rank estimation in NMF** Existing works on rank estimation for NMF is not applicable in the setting of this paper. The algebraic methods like fooling sets [12] and  $f$ -vector [13] only give a loose bound on  $\text{rank}_+(\mathbf{M})$  and are being expensive to implement. The statistical approaches [9, 11, 10] assume  $\mathbf{W}$  and  $\mathbf{H}$  follows some pre-defined distributions, or require on heavy post-processing. SON-NMF has none of these assumptions, restrictions nor post-processing.

**A “drawback” of SON-NMF** Finding the  $\text{rank}_+$  in NMF is NP-hard, the search space of  $r$  in NMF is the set of natural number  $\mathbb{N}$ , which has a cardinality of countably infinite. In SON-NMF we do not need to estimate the rank  $r$ , but we are required to provide a regularization parameter  $\lambda$ , in which its search space is the set of nonnegative real  $\mathbb{R}_+$ . By Cantor’s diagonal argument [38], the cardinality of real number is uncountably infinite. Hence, it seems in SON-NMF we are moving from NMF with a search space  $\mathbb{N}$  to SON-NMF with much larger search space  $\mathbb{R}_+$ , and thus SON-NMF is even more difficult to solve than the already NP-hard NMF. We remark that this is true theoretically, however it is not a problem practically because many datasets are hierarchically clustered in the latent space, and thus a simple tuning of  $\lambda$  can be used on SON-NMF to find the true rank.

**Paper organization** We provide theory of SON-NMF in section 2. We present how to solve SON-NMF in section 3 and section 4. We give experimental results in section 5. We conclude in section 6.

**Notation** The notation “ $\{x, y\}$  denotes  $\{X, Y\}$ ” means that we denote the object  $X$  by the symbol  $x$  and the object  $Y$  by the symbol  $y$  respectively (resp.). We use the symbols  $\{\mathbb{R}, \mathbb{R}_+, \bar{\mathbb{R}}, \mathbb{R}^m, \mathbb{R}^{m \times n}\}$  to denote {reals, nonnegative reals, extended reals,  $m$ -dimensional reals,  $m$ -by- $n$  reals}, we use {lowercase italic, bold lowercase italic, bold uppercase letters} to represent {scalar, vector, matrix}. Given a matrix  $\mathbf{M}$ , we denote  $\{\mathbf{m}^i, \mathbf{m}_j\}$  the  $\{i$ th row,  $j$ th column} of  $\mathbf{M}$ . Given a convex set  $C \subset \mathbb{R}^n$ , the indicator function of  $C$  at  $\mathbf{x}$  is defined as  $\iota_C(\mathbf{x}) = 0$  if  $\mathbf{x} \in C$  and  $\iota_C = +\infty$  if  $\mathbf{x} \notin C$ , and  $\text{proj}_C(\mathbf{x})$  denotes the projection of a point  $\mathbf{x}$  onto  $C$ . The projection of  $\{\mathbf{v} \in \mathbb{R}^n, \mathbf{V} \in \mathbb{R}^{m \times n}\}$  onto the nonnegative orthant  $\{\mathbb{R}_+^n, \mathbb{R}_+^{m \times n}\}$  is denoted by the element-wise max operator  $\{[\mathbf{v}]_+, [\mathbf{V}]_+\}$ . Lastly  $\Delta^r \in \mathbb{R}^r$  denotes the unit simplex and  $\mathbf{1}_r \in \mathbb{R}^r$  is the vector of 1s.

**Remark.** Note that the constraint on  $\mathbf{H}$  removes the scaling ambiguity of the factorization. That is, there do not exists a diagonal matrix  $\mathbf{D}$  that  $\mathbf{M} = \mathbf{W}_1 \mathbf{H}_1 = (\mathbf{W}_1 \mathbf{D})(\mathbf{D}^{-1} \mathbf{H}_1) =: \mathbf{W}_2 \mathbf{H}_2$  such that  $\mathbf{W}_1 \neq \mathbf{W}_2$  and  $\mathbf{H}_1 \neq \mathbf{H}_2$ .

## 2 Theory of SON-NMF

In this section we provide theories of SON<sub>2,1</sub>-NMF. First we give a closer look at SON<sub>2,1</sub>, then we give the motivation why one would like to reduce the complexity of SON<sub>2,1</sub>, next we give a bound showing that such complexity is irreducible. Lastly we discuss a greedy method utilising the property of SON<sub>2,1</sub>.

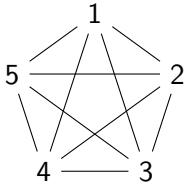
## 2.1 SON<sub>2,1</sub> has $r^2$ terms and its minimum occurs at maximal cluster-imbalance

Let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots]$  has five columns. Let  $\text{vec}$  denotes vectorization. The pair-wise difference  $P$  in SON can be expressed as

$$P(\mathbf{X}) = \text{Avec}(\mathbf{X}) = \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \cdots \\ \mathbf{I} & -\mathbf{I} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 - \mathbf{x}_2 \\ \mathbf{x}_1 - \mathbf{x}_3 \\ \vdots \end{bmatrix},$$

which has  $5^2 - 5 = 20$  pairs of  $(\mathbf{x}_i, \mathbf{x}_j)$ . In general, suppose  $\mathbf{X}$  has  $r$  columns, then the term  $P(\mathbf{X})$  contains  $r(r-1)$  pairs of  $(\mathbf{x}_i, \mathbf{x}_j)$ . By symmetry  $\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \|\mathbf{x}_j - \mathbf{x}_i\|_2$ , we drop the repeated terms so SON effectively has  $r(r-1)/2$  distinct pairs. We now switch to the language of graph theory. Denote  $G(V, E)$  a simple undirected unweighted graph of  $|V|$  nodes and  $|E|$  edges. Let  $K_r$  be complete graph of  $r$  nodes. Then  $\text{SON}_{2,1}(\mathbf{X}) = \sum \|\mathbf{x}_i - \mathbf{x}_j\|_2$  for  $(i, j) \in K_r$ . As  $|E(K_r)| = r(r-1)/2$ , so  $\text{SON}_{2,1}$  has  $\mathcal{O}(r^2)$  terms.

Back to the example of  $\mathbf{X}$  with five columns. Let  $\mathbf{X}$  be a rank-3 matrix with three clusters with centers  $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$  as  $\mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \mathbf{x}_3 \mathbf{x}_4 \mathbf{x}_5] = [\mathbf{c}_1 | \mathbf{c}_2 | \mathbf{c}_3 \mathbf{c}_3 \mathbf{c}_3]$  so  $\text{SON}_{2,1}(\mathbf{X}) = \|\mathbf{c}_1 - \mathbf{c}_2\|_2 + 3\|\mathbf{c}_1 - \mathbf{c}_3\|_2 + 3\|\mathbf{c}_2 - \mathbf{c}_3\|_2$ . We show the graph  $K_5$  and the pair-wise difference below.



	1	2	3	4	5
1	0	$\mathbf{c}_1 - \mathbf{c}_2$	$\mathbf{c}_1 - \mathbf{c}_3$	$\mathbf{c}_1 - \mathbf{c}_3$	$\mathbf{c}_1 - \mathbf{c}_3$
2		0	$\mathbf{c}_2 - \mathbf{c}_3$	$\mathbf{c}_2 - \mathbf{c}_3$	$\mathbf{c}_2 - \mathbf{c}_3$
3			0	0	0
4				0	0
5					0

Now we generalize: let  $\mathbf{X}$  with  $r$  columns has  $r^* \leq r$  clusters  $C_1, \dots, C_{r^*}$  with centers  $\mathbf{c}_1, \dots, \mathbf{c}_{r^*}$ . Let  $|C_i|$  denotes the cluster size of  $C_i$ . By  $|C_1| + \dots + |C_{r^*}| = r$  we have

$$\begin{aligned} \text{SON}_{2,1}(\mathbf{X}) &= \sum_{(i,j) \in K_r} |C_i||C_j| \|\mathbf{c}_i - \mathbf{c}_j\|_2 \\ &\leq \left( \max_{i \in [r]} |C_i| \right) \left( \max_{i,j} \|\mathbf{c}_i - \mathbf{c}_j\|_2 \right) \sum_{(i,j) \in K_r} |C_j| \leq \left( \max_{i \in [r]} |C_i| \right) \left( \max_{i,j} \|\mathbf{c}_i - \mathbf{c}_j\|_2 \right) r, \end{aligned}$$

giving a stopping criterion for the algorithm: we know  $r$  as an input, so we just need to track the product  $(\max_{i \in [r]} |C_i|)(\max_{i,j} \|\mathbf{c}_i - \mathbf{c}_j\|_2)$  for convergence. Furthermore, from the inequality we can focus on the cluster size instead of the norm  $\|\mathbf{c}_i - \mathbf{c}_j\|_2$ , and arrive at the following Lemma that characterizes the theoretical minimum of  $\text{SON}_{2,0}$  as a proxy of  $\text{SON}_{2,1}$ .

**Lemma 1** (Maximal cluster-imbalance gives the minimum of SON-2-0). *For a  $n$ -column matrix  $\mathbf{X}$  with  $K$  clusters  $C_1, \dots, C_K$  where  $\mathbf{x}_i \in C_i$  all takes the centroid  $\mathbf{c}_i$ , then  $\text{SON}_{2,0}(\mathbf{X}) = \sum_{i,j} |C_i||C_j|$  achieves the lowest value if a cluster takes  $n - K + 1$  columns in  $\mathbf{X}$  and the other  $K - 1$  clusters have a unit cluster size.*

*Proof.* Trivial by using the fact  $|C_1| + \dots + |C_K| = K$  with some inequality manipulations.  $\square$

### Remarks of Lemma 1

- As  $\ell_1$ -norm is a tight convex relaxation of the  $\ell_0$ -norm (over the unit ball), then Lemma 1 on the  $\text{SON}_{2,0}$  term gives a theoretical minimum for the  $\text{SON}_{2,1}$  term (if the input matrix is in a unit ball).
- For any cluster  $C_i$  the smallest cluster size is 1 and it is impossible for the  $\text{SON}_{2,0}$  term (similarly for the  $\text{SON}_{2,1}$  term) to “miss” a weak component in the data, if exist. This is observed in the experiment, see Fig. 7 and Fig. 5 in Section 5. Furthermore, if the centers  $\mathbf{c}_i$  have similar distance to each other:  $\|\mathbf{c}_i - \mathbf{c}_j\|_2 \approx \|\mathbf{c}_j - \mathbf{c}_k\|_2$ , then maximal cluster-imbalance will occur in the application. See Fig. 3, Fig. 4, Fig. 6, Fig. 7 in Section 5

## 2.2 SON complexity is irreducible

We now see that SON has  $\mathcal{O}(r^2)$  terms. In the application, we are using a large input rank  $r$  (possibly as large as the data-size  $m, n$ ) in the SON-NMF to estimate the true rank  $r^*$  of the data. This means the SON term has a high computational complexity and thus is impractical. So it natural to ask whether it is possible to reduce the complexity of the SON term by removing some edges in  $K_r$ , so that we can cut the per-iteration cost of

running SON-NMF, while retaining some recovery performance of SON-NMF. We give a negative result to this idea (Theorem 1). That is, the complexity of SON term is almost irreducible.

**Remark.** There are works on the literature with a similar idea. For example, [35, page 2] mentioned approaches using  $k$ -nearest neighbors. We remark that these are data-dependent approaches that use the data to learn a graph structure for reducing the complexity of the SON term. Our focus here is different. We are focusing on the possibility of reducing the complexity of the SON term purely from the graph perspective, independent of data. I.e., we are interested in finding the possible sparsest subgraph that such a reduced-SON is the “functionally the same” as the full-SON, and Theorem 1 below is saying that such sparsest subgraph basically does not exist.

We first give notation. Let  $r^*$  be the true NMF rank of  $\mathbf{W}$ . For a graph  $G(V, E)$ , let  $u, v$  be two nodes in  $V$  of  $G$  that there is an edge between them, i.e.,  $(u, v) \in E$ . The notation  $G \setminus (u, v)$  denotes the subgraph of  $G$  removing the edge  $(u, v)$ . The notion of *graph partition* of  $G$  is the set of subgraphs  $S_1, S_2, \dots$  of  $G$  where  $V(S_i)$  are the partition of  $V(G)$  that  $V(S_i)$  are mutually exclusive sets. Now, we have a trivial fact.

**Lemma 2.** Let  $\mathbf{W}$  has the true NMF rank  $r^*$ . Then the graph generated by the columns of  $\mathbf{W}$  must have the following property. For every possible partitioning of the nodes of the graph into  $r^*$  subgraphs, each subgraph needs to be connected.

This lemma can be proved easily by contradiction. Now we have the following lemma.

**Lemma 3.** The only graph that satisfies the condition of Lemma 2 is the complete graph.

*Proof.* Let  $G$  be a graph whose nodes correspond to columns of  $\mathbf{W}$ . Suppose that some particular edge  $(u, v)$  is omitted from  $G$ . Then it is possible that the ‘true’ partition is:  $(u, v)$  is one partition, while the remaining nodes of  $G \setminus (u, v)$  are divided arbitrarily among the other  $r^* - 1$  partitions. In this case,  $u$  and  $v$  are not connected by edges except through  $G \setminus (u, v)$ . Since  $u, v$  were arbitrary, the conclusion is that no edge can be omitted from  $G$ .  $\square$

The lemma means that apart from the complete graph, we cannot consider other graph structure. The following theorem quantify the amount of edge we can remove from the complete graph.

**Theorem 1.** Let  $(\mathbf{W}^*, \mathbf{H}^*) = \text{NMF}(\mathbf{M}, r^*)$  be the true solution, and let  $(\mathbf{W}, \mathbf{H}) = \text{SON-NMF}(\mathbf{M}, r)$  with  $r \geq r^*$  be another solution. If we want to recover  $\mathbf{W}^*$  using  $r^*$  clusters of the  $r$  columns of  $\mathbf{W}$ , we cannot reduce the number of pairwise difference term  $\|\mathbf{w}_i - \mathbf{w}_j\|_2$  in SON below  $r(r - \lceil r/r^* \rceil)/2$ .

*Proof.* We construct a simple undirected unweighted graph  $G(V, E)$  with  $|V| = r$  nodes where each node  $v \in V$  represents a vector  $\mathbf{w}_i$  of  $\mathbf{W}$  produced by SON-NMF( $\mathbf{M}, r$ ). Here an edge  $e(u, v)$  denotes the pairwise difference  $\|\mathbf{w}_i - \mathbf{w}_j\|_2$ . Now, recovering  $\mathbf{W}^*$  by  $\mathbf{W}$  with fewer terms in the SON regularizer can be translated as:

we can identify  $r^*$  disjoint clusters in  $G$  with  $|V| = r$  using a subgraph of  $K_r$  with fewer edges. (2)

We are going to show that the statement (2) is true, and at best such an improvement is from  $r(r - 1)/2$  to  $r(r - \lceil r/r^* \rceil)/2$ .

Assuming, in the best case that, each of these  $r^*$  columns of  $\mathbf{W}^*$  is associated with exactly  $\lceil r/r^* \rceil$  nodes in  $\mathbf{W}$ , represented by the nodes in the graph  $G$ . Consider a node  $v \in V$ , denote  $S(v) \subset V$  be the set of nodes that are disconnected to  $v$  (i.e., there is no path between  $u \in S(v)$  and  $v$ ), and let  $T$  be a nonempty subset of  $S(v)$ . Then the recovery of the  $r^*$  clusters in  $G$  is impossible if a cluster in  $G$  is of the form  $\{v\} \cup T$ . The negation of this very last statement gives:

To recover the  $r^*$  clusters for all subset of nodes of size at least  $r/r^*$ , we need  $|T| < r/r^*$  for any such  $T$ .

The inequality  $|T| < r/r^*$  has to hold for any subset  $T$  of  $S(v)$ , this implies  $|S(v)| < r/r^*$ . I.e.,  $v$  has to connect to at least  $r - \lceil r/r^* \rceil$  other nodes  $u \notin S(v)$  in the graph  $G$ . This connectivity holds for every node  $v \in V$ , meaning that at best the graph has  $\frac{r}{2} \left( r - \lceil \frac{r}{r^*} \rceil \right)$  number of edges.  $\square$

Theorem 1 tells that not much improvement can be made on reducing the number of edges from  $r(r - 1)/2$  of  $K_r$  for the SON. We can also look at this from another angle. First, we define the reduction factor  $R(r^*, r)$  as

$$R(r; r^*) := \frac{\text{full number of terms} - \text{reduced number of terms}}{\text{full number of terms}} = \frac{\frac{r}{2}(r - 1) - \frac{r}{2}(r - \lceil \frac{r}{r^*} \rceil)}{\frac{r}{2}(r - 1)}.$$

The following lemma tells that the reduction factor is small.

**Lemma 4.** For fix  $r^*$ , the value  $R(r^*, r)$  approaches to  $1/r^*$  for increasing  $r$ . I.e.,  $\lim_{r \rightarrow \infty} R(r; r^*) = 1/r^*$ .

*Proof.* Take the limit of  $R(r; r^*)$  gives  $\lim_{r \rightarrow \infty} R(r; r^*) = \lim_{r \rightarrow \infty} \frac{\lceil r/r^* \rceil - 1}{r - 1} = \lim_{r \rightarrow \infty} \frac{\lceil r/r^* \rceil}{r - 1}$ . Using  $r \leq \lceil r \rceil \leq r + 1$ , we have  $\lim_{r \rightarrow \infty} \frac{r/\lceil r^* \rceil}{r - 1} \leq \lim_{r \rightarrow \infty} R(r; r^*) \leq \lim_{r \rightarrow \infty} \frac{(r + 1)/\lceil r^* \rceil}{r - 1}$ . By squeeze theorem,  $\lim_{r \rightarrow \infty} R(r; r^*) = \frac{1}{\lceil r^* \rceil} \leq \frac{1}{r^*}$ . By the fact that ceiling function is lower semicontinuous, the limit touches the upper bound  $1/r^*$ .  $\square$

The lemma shows that we can only reduce the complexity of the SON term by a small amount. For  $r^* \geq 3$  (NMF is trivial for  $r^* \leq 2$  [2]), we achieve a reduction of 33%. The reduction decreases quickly to zero as  $r$  and/or  $r^*$  increases. For example, with  $(r, r^*) = (1000, 25)$ , i.e., using 1000 nodes to find 25 clusters, we can at best reducing the number of edges of  $K_{1000}$  only by 5%, i.e., from  $|K_{1000}| = 499500$  edges to  $500(1000 - \lceil 1000/25 \rceil) = 480000$ .

### 3 BCD algorithm and the H-subproblem

We now discuss how to solve the nonsmooth nonconvex nonseparable non-proximable minimization problem (SON-NMF) by block coordinate descent (BCD) [39, 40]. Let  $k$  denotes the iteration counter. Starting with an initial guess  $(\mathbf{W}_1, \mathbf{H}_1)$ , we perform alternating update as  $\mathbf{H}_{k+1} \leftarrow \text{update}(\mathbf{H}_k; \mathbf{W}_k)$ ,  $\mathbf{W}_{k+1} \leftarrow \text{update}(\mathbf{W}_k; \mathbf{H}_{k+1})$ , where  $\text{update}()$  is performed by approximately solving a subproblem. Here we discuss the BCD framework and how we update  $\mathbf{H}$ . We discuss how we handle the subproblem on  $\mathbf{W}$  in the next section.

Algorithm 1 shows the pseudo-code of the BCD method for solving SON-NMF.

**Algorithm 1:** (Inexact) BCD for solving SON-NMF

---

**Input:**  $\mathbf{M}, \mathbf{W}_1, \mathbf{H}_1, \lambda, \gamma$

- 1 **for**  $k = 1, 2, \dots$  **do**
- 2    $\mathbf{H}_{k+1} = \text{proj}_{\Delta^r}(\mathbf{Q}\mathbf{H}_k + \mathbf{R})$  with  $\mathbf{Q} = \mathbf{I}_n - \mathbf{W}_k^\top \mathbf{W}_k / \|\mathbf{W}_k^\top \mathbf{W}_k\|_2$  and  $\mathbf{R} = \mathbf{W}_k^\top \mathbf{M} / \|\mathbf{W}_k^\top \mathbf{W}_k\|_2$
- 3   **for**  $\ell = 1, 2, \dots, \ell_{\max}$ , (e.g., 10) **do**
- 4      $\mathbf{W}_{k+1} = \text{update}(\mathbf{W}_k; \mathbf{H}_{k+1}, \mathbf{M}, \lambda, \gamma)$ , see section 4.

---

We now explain Step 2 in Algorithm 1.

**H-subproblem: projection onto unit simplex** The step  $\text{update}(\mathbf{H}_k; \mathbf{W}_k)$  is performed by solving the subproblem on  $\mathbf{H}$ , which contains  $n$  parallel problems as

$$\underset{\mathbf{h}_1, \dots, \mathbf{h}_n}{\operatorname{argmin}} \frac{1}{2} \sum_{j=1}^n \|\mathbf{W}_k \mathbf{h}_j - \mathbf{m}_j\|_2^2 \text{ s.t. } \mathbf{h}_j \in \Delta^r := \left\{ \mathbf{x} \in \mathbb{R}_+^r : \sum_i x_i \leq 1 \right\} \text{ for } j = 1, 2, \dots, n. \quad (3)$$

The subproblem on each column  $\mathbf{h}_j$  is a constrained least squares in the form

$$\underset{\mathbf{x} \in \Delta^r}{\operatorname{argmin}} f(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \frac{1}{2} \langle \mathbf{A}^\top \mathbf{A}\mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{A}^\top \mathbf{b}, \mathbf{x} \rangle, \quad (4)$$

where  $\mathbf{x}$  is the variable  $\mathbf{h}_j$ , and we have  $\mathbf{A} = \mathbf{W}^\top \mathbf{W}$  with  $\mathbf{b} = \mathbf{W}^\top \mathbf{m}_j$ . We use proximal gradient method (details in the next section) to update  $\mathbf{x}$  in (4) iteratively as

$$\mathbf{x}_k^{\ell+1} = \text{proj}_{\Delta^r} \left( \mathbf{x}_k^\ell - \frac{\mathbf{A}^\top \mathbf{A}\mathbf{x}_k^\ell - \mathbf{A}^\top \mathbf{b}}{\|\mathbf{A}^\top \mathbf{A}\|_2} \right) = \text{proj}_{\Delta^r} \left( \left( \mathbf{I} - \frac{\mathbf{A}^\top \mathbf{A}}{\|\mathbf{A}^\top \mathbf{A}\|_2} \right) \mathbf{x}_k^\ell + \frac{\mathbf{A}^\top \mathbf{b}}{\|\mathbf{A}^\top \mathbf{A}\|_2} \right), \quad (5)$$

where  $\mathbf{x}_k^\ell$  is the variable at iteration- $k$  and inner-iteration- $\ell$ . In short, for each column  $\mathbf{h}_j$  in  $\mathbf{H}$ , running (5) several times over the counter  $\ell$  will solve (4) at iteration  $k$ . We take  $\ell = 1$  to achieve an update scheme with low per-iteration cost.

**Projection**  $\text{proj}_{\Delta^r}(\mathbf{x})$  projects a vector  $\mathbf{x} \in \mathbb{R}^r$  onto  $\Delta^r$  with a cost  $\mathcal{O}(r \log r)$  [41] comes from the sorting procedure for finding the Lagrangian multiplier when solving the projection subproblem.

**The overall update** The aforementioned column-wise update can be combined into a matrix update as

$$\mathbf{H}_{k+1} = \text{proj}_{\Delta^r} \left( \mathbf{H}_k - \frac{\mathbf{W}_k^\top \mathbf{W}_k \mathbf{H}_k - \mathbf{W}_k^\top \mathbf{M}}{\|\mathbf{W}_k^\top \mathbf{W}_k\|_2} \right),$$

where  $\text{proj}_{\Delta^r}$  is implemented in parallel for the  $n$  columns. The total cost of  $\text{proj}_{\Delta^r}(\mathbf{H})$  is  $\mathcal{O}(nr \log r)$ , or  $\mathcal{O}(n^2 \log n)$  if  $r \approx n$ . This high cost partly explains why we do not consider 2nd-order method for updating  $\mathbf{H}$ . Below we give another reason for not considering 2nd-order method for updating  $\mathbf{H}$ : the  $\mathbf{W}$  is multicollinear.

**On the price to pay for the multicollinearity of  $\mathbf{W}$**  We now give an important remark regarding the matrix  $\mathbf{W}_k^\top \mathbf{W}_k$ . As stated in the introduction, the SON terms encourage the multicollinearity of  $\mathbf{W}$ , hence possibly  $\mathbf{W}$  is ill-conditioned, and  $\mathbf{W}^\top \mathbf{W}$  has a huge condition number. This has negative effects on Problem (3):

1. Now Problem (3) is not a strongly-convex, leading to the possibility of having multiple global minima.
2. When applying Nesterov's acceleration [42] in the update of  $\mathbf{H}$ , the optimal scheme became less effective since the acceleration slow down for a huge conditional number of  $\mathbf{W}_k^\top \mathbf{W}_k$ .
3. We cannot use 2nd-order method for updating  $\mathbf{H}$  because  $(\mathbf{W}_k^\top \mathbf{W}_k)^{-1}$  may not even exists.
4. Tools from duality cannot be efficiently utilized on Problem (3). E.g., to design a stopping criterion.

## 4 Proximal averaging on the $\mathbf{W}$ -subproblem

In this section we focus on solving the  $\mathbf{W}$ -subproblem, the line update( $\mathbf{W}; \mathbf{H}, \mathbf{M}, \lambda, \gamma$ ) in Algorithm 1:

$$\underset{\mathbf{W}}{\operatorname{argmin}} F(\mathbf{W}) := \frac{1}{2} \|\mathbf{W}\mathbf{H} - \mathbf{M}\|_F^2 + \lambda \sum_{i \neq j} \|\mathbf{w}_i - \mathbf{w}_j\|_2 + \gamma \sum_{j=1}^r \|\max\{-\mathbf{w}_j, \mathbf{0}\}\|_1. \quad (6)$$

We remark that  $F(\mathbf{W})$  in (6) is convex, nonsmooth, Lipschitz-continuous, non-separable and non-proximable.

- $F(\mathbf{W})$  is convex and continuous: the terms in  $F$  are norms under some convex-preserving maps, hence  $F$  is convex. Furthermore, norms are continuous, and  $\|\mathbf{w}_i - \mathbf{w}_j\|_2$ ,  $\|\max\{-\mathbf{w}_j, \mathbf{0}\}\|_1$  are 1-Lipschitz.
- $F(\mathbf{W})$  is nonsmooth:  $\|\mathbf{w}_i - \mathbf{w}_j\|_2$  is not differentiable at  $\mathbf{w}_i = \mathbf{w}_j$  and  $\|\max\{-\mathbf{w}_j, \mathbf{0}\}\|_1$  is not differentiable when any component of  $\mathbf{w}$  is negative.
- $F(\mathbf{W})$  is non-separable:  $\mathbf{w}_i, \mathbf{w}_j$  are lumped together in the SON term, the function  $F(\mathbf{W})$  cannot be separated into component-wise  $F(\mathbf{w}_j)$  that solely contains one column  $\mathbf{w}_j$ .
- $F(\mathbf{W})$  is non-proximable: the prox operator (see details below) for  $\lambda \sum \|\mathbf{w}_i - \mathbf{w}_j\|_2 + \gamma \sum \|\max\{-\mathbf{w}_j, \mathbf{0}\}\|_1$  has no closed-form solution nor can be solved efficiently.
- $F(\mathbf{W})$  is “not-dualizable”: the value  $r$ , the number of columns in  $\mathbf{W}$ , is possibly as large as  $m, n$ . If we introduce dual variable / Lagrangian multiplier in  $F$  and apply dual methods (e.g., augmented Lagrangian, ADMM), the number of vector variables will explode from  $r$  ( $\sim \mathcal{O}(m)$ ) to  $r^2$  ( $\sim \mathcal{O}(m^2)$ ).
- $F(\mathbf{W})$  is “not 2nd-order friendly”: based on the same reason stated above, we do not consider 2nd-order method here as the per-iteration cost of updating  $\mathbf{W}$  is high, between  $\mathcal{O}(m^4)$  to  $\mathcal{O}(m^5)$ .

As  $F(\mathbf{W})$  is non-separable and non-proximable, proximal gradient methods [20, 21, 22, 23, 24] can not be applied to efficiently solve (6). We solve (6) by a technique of Moreau-Yosida envelop called proximal averaging [25], which is more efficient than inexact proximal step [43] and smoothing [44] that both requires parameter tuning.

**Remark.** The penalty  $\sum_i \|\max\{-\mathbf{w}_i, \mathbf{0}\}\|_1$  guarantees  $\mathbf{W} \geq \mathbf{0}$  if  $\gamma > 0$  is sufficiently large.

**Column-wise update** We solve (6) column-by-column. Consider the  $j$ th component of the rank-1 factor in (1), i.e.,  $\mathbf{w}_j \mathbf{h}^j$ . Let  $\mathbf{M}_j = \mathbf{M} - \mathbf{W}_{-j} \mathbf{H}^{-j}$  where  $\mathbf{W}_{-j}$  is  $\mathbf{W}$  without the column  $\mathbf{w}_j$  and  $\mathbf{H}^{-j}$  is  $\mathbf{H}$  without the row  $\mathbf{h}^j$ . After some algebra the subproblem (6) on one column  $\mathbf{w}_j$  becomes

$$\mathbf{w}_j^* := \operatorname{argmin}_{\mathbf{w}} \frac{\|\mathbf{h}^j\|_2^2}{2} \|\mathbf{w}\|_2^2 - \langle \mathbf{M}_j \mathbf{h}^{j\top}, \mathbf{w} \rangle + \lambda \sum_{i \neq j} \|\mathbf{w} - \mathbf{w}_i\|_2 + \gamma \|\max\{-\mathbf{w}, \mathbf{0}\}\|_1. \quad (7)$$

which is in the form

$$\operatorname{argmin}_{\mathbf{x}} \phi(\mathbf{x}) + \psi(\mathbf{x}), \quad \text{where } \psi(\mathbf{x}) := \sum_{i=1}^N \alpha_i \psi_i(\mathbf{x}), \quad (8)$$

where  $\phi : \mathbb{R}^m \rightarrow \mathbb{R}$  is a closed proper convex smooth function, all  $\psi_i : \mathbb{R}^m \rightarrow \overline{\mathbb{R}}$  are convex closed proper functions that are (possibly) nonsmooth, and  $\alpha_i \geq 0$ ,  $\sum \alpha_i = 1$  are normalized averaging coefficients, i.e., we normalize  $\lambda$  and  $\gamma$  to obtain  $\alpha_i$ . Note that  $\psi_i$  are non-separable, i.e., they share the same global variable  $\mathbf{x}$ .

**Proximal gradient method** A popular approach to solve minimization (8) is the proximal gradient method [45, 46, 47], in which the update under a stepsize  $\mu > 0$  is  $\mathbf{x}^+ = \mathbf{P}_\psi^\mu(\mathbf{x} - \mu \nabla \phi(\mathbf{x}))$ , where  $\mathbf{P}_\psi^\mu$  denotes the proximal operator associated with  $\psi$ , see (9) for the expression. By the fact that  $\psi(\mathbf{x}) := \sum_{i=1}^N \alpha_i \psi_i(\mathbf{x})$  in (8), we have  $\mathbf{P}_\psi^\mu = \mathbf{P}_{\sum \alpha_i \psi_i}^\mu$  in which currently there is no efficient way to compute, and this is what we mean that  $\psi$  is “non-proximable”. To handle this we make use of the idea of proximal average [26, 25]. Below we give the background of proximal average for solving (8) and then we discuss how to apply proximal average to solve (7).

## 4.1 Proximal average

Given a point  $\mathbf{v} \in \mathbb{R}^n$ , a convex closed proper function  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  and a parameter  $\mu > 0$ , the proximal operator of  $f$  at  $\mathbf{v}$ , denoted as  $\mathbf{P}_f^\mu(\mathbf{v})$ , and the Moreau-Yosida envelope, or in short Moreau envelope, of  $f$  at  $\mathbf{v}$ , denoted as  $\mathbf{M}_f^\mu(\mathbf{v})$ , are defined as

$$\begin{aligned} \mathbf{P}_f^\mu(\mathbf{v}) &:= \operatorname{argmin}_{\xi} f(\xi) + \frac{1}{2\mu} \|\xi - \mathbf{v}\|_2^2, \\ \mathbf{M}_f^\mu(\mathbf{v}) &:= \min_{\xi} f(\xi) + \frac{1}{2\mu} \|\xi - \mathbf{v}\|_2^2. \end{aligned} \quad (9)$$

**Algorithm 2:** Proximal averaging for solving (8)

```

1 for  $k = 1, 2, \dots$  do
2    $\bar{\mathbf{x}} = \mathbf{x}_k - \mu \nabla \phi(\mathbf{x}_k)$            gradient step
3    $\mathbf{x}_{k+1} = \sum_{i=1}^N \alpha_i \mathbf{P}_{\psi_i}^\mu(\bar{\mathbf{x}})$     proximal average

```

The idea of proximal average is that  $\mathbf{P}_\psi^\mu = \mathbf{P}_{\sum \alpha_i \psi_i}^\mu$  is hard to compute but  $\mathbf{P}_{\psi_i}^\mu$  for each  $i$  is easy to compute, so we replace  $\mathbf{P}_\psi^\mu$  by  $\sum_{i=1}^N \alpha_i \mathbf{P}_{\psi_i}^\mu$ . Algorithm 2 shows the proximal averaging approach for solving (8). On the convergence, under the assumptions that  $\phi$  is  $L$ -smooth and  $\psi_i$  are all  $M_i$ -Lipschitz, then the sequence  $\{\mathbf{x}_k\}_{k \in \mathbb{N}}$  produced by Algorithm 2 converges to the solution of

$$\operatorname{argmin}_{\mathbf{x}} \phi(\mathbf{x}) + A(\mathbf{x}), \quad \text{where } \mathbf{M}_A^\mu = \sum_i \alpha_i \mathbf{M}_{\psi_i}^\mu, \quad (\#)$$

i.e.,  $A$  with Moreau envelope equals to the average of the Moreau envelope of  $\psi_i$  is called the proximal average of  $\{\psi_1, \dots, \psi_n\}$  [25]. Furthermore, we have  $0 \leq \psi - A \leq \frac{\mu}{2} \sum_i \alpha_i M_i^2 < +\infty$  and that an  $\epsilon$ -solution for (#) is an  $2\epsilon$ -solution for (8).

## 4.2 Update on $\mathbf{w}$

Now we discuss how to use proximal average (Algorithm 2) to solve the  $\mathbf{W}$ -subproblem. First the subproblem satisfies the assumptions of the theory of proximal average. Then, let  $\sigma = (r-1)\lambda + \gamma$  be a normalization factor. Rewrite (7) as

$$\operatorname{argmin}_{\mathbf{w}} \frac{\|\mathbf{h}^j\|_2^2}{2} \|\mathbf{w}\|_2^2 - \langle \mathbf{M}_j \mathbf{h}^{j\top}, \mathbf{w} \rangle + \sigma \left( \sum_{1 \leq i \neq j \leq r}^r \frac{\lambda}{\sigma} \|\mathbf{w} - \mathbf{w}_i\|_2 + \frac{\gamma}{\sigma} \|\max\{-\mathbf{w}, \mathbf{0}\}\|_1 \right). \quad (10)$$

Since  $\text{argmin}$  is invariant to scaling, i.e.,  $\text{argmin } F = \text{argmin } \alpha F$  for all  $\alpha > 0$ , we rewrite (10) as

$$\underset{\mathbf{w}}{\text{argmin}} \underbrace{\frac{\|\mathbf{h}^j\|_2^2}{2\sigma} \|\mathbf{w}\|_2^2 - \left\langle \frac{\mathbf{M}_j \mathbf{h}^{j\top}}{\sigma}, \mathbf{w} \right\rangle}_{\phi} + \sum_{1 \leq i \neq j \leq r}^r \frac{\lambda}{\sigma} \|\mathbf{w} - \mathbf{w}_i\|_2 + \frac{\gamma}{\sigma} \|\max\{-\mathbf{w}, \mathbf{0}\}\|_1. \quad (11)$$

The reason we scale (10) to get (11) is to make sure the assumption of proximal average for solving problems in the form of (8) is satisfied in (11). Then we have the gradient  $\nabla \phi(\mathbf{w}) = \|\mathbf{h}^j\|_2^2 \mathbf{w}/\sigma - \mathbf{M}_j \mathbf{h}^{j\top}/\sigma$  and it is  $(\|\mathbf{h}^j\|_2^2/\sigma)$ -Lipschitz. The gradient descent step (line 2 of Algorithm 2) is thus

$$\bar{\mathbf{w}} = \mathbf{w} - \frac{1}{L} \nabla \phi(\mathbf{w}) = \mathbf{w} - \frac{1}{\|\mathbf{h}^j\|_2^2/\sigma} \left( \frac{\|\mathbf{h}^j\|_2^2}{\sigma} \mathbf{w} - \frac{\mathbf{M}_j \mathbf{h}^{j\top}}{\sigma} \right) = \frac{\mathbf{M}_j \mathbf{h}^{j\top}}{\|\mathbf{h}^j\|_2^2}.$$

Next we recall three useful lemmas for computing the prox of each nondifferentiable terms:

**Lemma 5** (Scaling). *If  $\nu > 0, \mu > 0$  then  $P_{\nu\psi}^\mu = P_\psi^{\nu\mu}$ .*

**Lemma 6.** *The proximal operator of  $\|\mathbf{x} - \mathbf{c}\|_2$  with parameter  $\mu$  is  $P_{\|\mathbf{x}-\mathbf{c}\|_2}^\mu(\mathbf{v}) = \mathbf{v} - \frac{\mathbf{v} - \mathbf{c}}{\max\{1, \|\frac{\mathbf{v}-\mathbf{c}}{\mu}\|_2\}}$ .*

**Lemma 7.** *Let  $\mathbf{1}$  be the vector of ones, the proximal operator of  $\mu \|\max\{-\mathbf{x}, \mathbf{0}\}\|_1$  has the closed-form expression  $\text{median}(\mathbf{v} + \mu \mathbf{1}, \mathbf{0}, \mathbf{v})$ , i.e.,*

$$\left[ P_{\mu \|\max\{-\cdot, \mathbf{0}\}\|_1}^1(\mathbf{v}) \right]_i = \begin{cases} v_i + \mu & v_i + \mu < 0, \\ 0 & v_i \leq 0 \leq v_i + \mu, \\ v_i & v_i > 0. \end{cases}$$

Based on the three lemmas, the proximal step for the SON terms is

$$P_{\|\cdot - \mathbf{w}_i\|_2}^{\frac{1}{L_j} \frac{\lambda}{\sigma}}(\bar{\mathbf{w}}) = P_{\|\cdot - \mathbf{w}_i\|_2}^{\frac{\lambda}{\|\mathbf{h}^j\|_2^2}}(\bar{\mathbf{w}}) = \bar{\mathbf{w}} - \frac{\bar{\mathbf{w}} - \mathbf{w}_i}{\max\{1, \left\| \frac{\bar{\mathbf{w}} - \mathbf{w}_i}{\lambda/\|\mathbf{h}^j\|_2^2} \right\|_2\}},$$

and the proximal step for the penalty term is

$$P_{\frac{1}{L_j} \frac{\gamma}{\sigma} \|\max\{-\cdot, \mathbf{0}\}\|_1}^1(\bar{\mathbf{w}}) = \text{median}\left(\bar{\mathbf{w}} + \frac{1}{L_j} \frac{\gamma}{\sigma} \mathbf{1}, \mathbf{0}, \bar{\mathbf{w}}\right) = \text{median}\left(\bar{\mathbf{w}} + \frac{\gamma}{\|\mathbf{h}^j\|_2^2} \mathbf{1}, \mathbf{0}, \bar{\mathbf{w}}\right).$$

Algorithm 3 uses the proximal average as one iteration of update( $\mathbf{W}_k; \mathbf{H}_{k+1}$ ) in the BCD framework. Repeating this steps in Algorithm 3 will eventually solve the W-subproblem (6). In terms of per-iteration cost, one complete for-loop in Algorithm 3 has the cost  $\mathcal{O}(r^2m)$ , or  $\mathcal{O}(m^3)$  if  $r \approx m$ .

**Algorithm 3:** One iteration of update( $\mathbf{W}_k; \mathbf{H}_{k+1}; \mathbf{M}, \lambda, \gamma$ ) as a proximal averaging step

---

```

1 for  $j = 1, 2, \dots, r$  do
2   Compute  $\|\mathbf{h}^j\|_2^2$ ,  $\mathbf{M}_j = \mathbf{M} - \mathbf{W}\mathbf{H} + \mathbf{w}_j \mathbf{h}^j$ 
3   Update  $\mathbf{w}_j$  by solving (7) using one iteration of proximal-average as follows:
4     Compute  $\bar{\mathbf{w}} = \mathbf{M}_j \mathbf{h}^{j\top} / \|\mathbf{h}^j\|_2^2$ 
5     For  $i \neq j$ , compute  $P_{\|\cdot - \mathbf{w}_i\|_2}^{\frac{1}{L_j} \frac{\lambda}{\sigma}}(\bar{\mathbf{w}}) = P_{\|\cdot - \mathbf{w}_i\|_2}^{\frac{\lambda}{\|\mathbf{h}^j\|_2^2}}(\bar{\mathbf{w}}) = \bar{\mathbf{w}} - \frac{\bar{\mathbf{w}} - \mathbf{w}_i}{\max\{1, \left\| \frac{\bar{\mathbf{w}} - \mathbf{w}_i}{\lambda/\|\mathbf{h}^j\|_2^2} \right\|_2\}}$ 
6     Compute  $P_{\frac{1}{L_j} \frac{\gamma}{\sigma} \|\max\{-\cdot, \mathbf{0}\}\|_1}^1(\bar{\mathbf{w}}) = \text{median}\left(\bar{\mathbf{w}} + \frac{\gamma}{\|\mathbf{h}^j\|_2^2} \mathbf{1}, \mathbf{0}, \bar{\mathbf{w}}\right)$ 
7      $\mathbf{w} = \sum_{i \neq j}^r \frac{\lambda}{\sigma} P_{\|\cdot - \mathbf{w}_i\|_2}^{\frac{1}{L_j} \frac{\lambda}{\sigma}}(\bar{\mathbf{w}}) + \frac{\gamma}{\sigma} P_{\frac{1}{L_j} \frac{\gamma}{\sigma} \|\max\{-\cdot, \mathbf{0}\}\|_1}^1(\bar{\mathbf{w}})$ 

```

---

**Remark.** Existing approaches like quadratic programming with convex hull [28], active-set [30], interior-point method [29], trust-region with smoothing [31] and semi-smooth Newton's method [35] are solving SON-clustering, not SON-NMF. Modifying these approaches for SON-NMF is out of the scope of this work.

**Remark** (Why not using hard constraints for  $\mathbf{W} \geq \mathbf{0}$ ?). In NMF, the nonnegativity constraint  $\mathbf{W} \geq \mathbf{0}$  is normally enforced by adding an indicator function  $\iota_+(\mathbf{W})$  into the objective, where  $\iota_+$  is applied on  $\mathbf{W}$  element-wise that  $\iota_+(W_{ij}) = 0$  if  $W_{ij} \geq 0$  and  $\iota_+(W_{ij}) = +\infty$  if  $W_{ij} < 0$ . If we consider SON-NMF with the hard constraints  $\mathbf{W} \geq \mathbf{0}$ , it is possible that the output of the proximal average step is not strictly feasible, and thus making the objective function value at that iteration go to  $+\infty$ , and destroy the convergence of the whole method.

**Post-processing to extract columns of  $\mathbf{W}$**  Once the  $\text{SON}_{2,1}$  norm of  $\mathbf{W}$  with overestimated rank is minimized, we pick the columns of  $\mathbf{W}$  in each cluster to form the final rank-reduced solution matrix  $\mathbf{W}$ . We do so on the rows on  $\mathbf{H}$ .

## 5 Experiment

In this section we present numerical results to

- support the effectiveness of the algorithm for solving SON-NMF.
- showcase the ability of the SON-NMF in identifying the rank without prior knowledge.

Section organization. In section 5.1 we showcase the ability of SON-NMF in identifying the rank parameter without prior knowledge. In section 5.2 we showcase that the proposed algorithm is much faster than ADMM approach and Nesterov’s smoothing.

All the experiments were conducted on a Apple MacBook Air (M2 chipset, 8 CPU cores, 8 GPU cores) with a 3.5GHz CPU and 8 GB RAM. A Python library is available<sup>2</sup>.

### 5.1 SON-NMF identifies the rank parameter without prior knowledge

Here we solve SON-NMF on a datasets that we know the true NMF factorization rank  $r^*$ . In the experiment we intentionally set the rank parameter  $r$  higher than  $r^*$ , to show that SON-NMF is able to identify  $r^*$ .

#### 5.1.1 Synthetic data

First we use a synthetic data [3] that the data matrix  $\mathbf{Z} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$  with  $\text{rank}(\mathbf{Z}) = 3 < 4 = \text{rank}_+(\mathbf{Z})$ .

**Dataset generation** We follows [3]. In the experiment, we use  $\mathbf{Z}$  as the ground truth  $\mathbf{W}$ , denoted as  $\mathbf{W}_{\text{true}}$ , we generate the ground truth  $\mathbf{H}$ , denoted as  $\mathbf{H}_{\text{true}}$ , by sampling from a Dirichlet distribution with distribution parameter  $\alpha = 0.05$  for each element in a column vector. Then we generate the data matrix  $\mathbf{M} = \mathbf{W}_{\text{true}} \mathbf{H}_{\text{true}} + \mathbf{N}$  where  $\mathbf{N} \sim \mathcal{N}(0, 1)$  is random noise generated by sampling from normal distribution using `numpy.random.randn`<sup>3</sup>.

**Experiment** We solve (SON-NMF) using the inexact-BCD (Algorithm 1) with proximal average (Algorithm (3)) with the following setting

- We initialize  $\mathbf{W}, \mathbf{H}$  randomly under uniform distribution over interval  $[0, 1]$  by `numpy.random.rand`<sup>4</sup>
- We run 1 update iteration on  $\mathbf{H}$  and 10 iterations on  $\mathbf{W}$ . I.e., we repeat Algorithm (3) 10 times before switching to updating  $\mathbf{H}$ .
- We stop the algorithm when the relative error between iterations, defined as  $(F_k - F_{k-1})/F_{k-1}$ , is less than  $10^{-6}$ , or the iteration counter reaches the maximum number of iteration.
- Table 1 shows the parameters used in the experiments.

<sup>2</sup><https://github.com/waqasbinhamed/sonnmf>

<sup>3</sup><https://numpy.org/doc/stable/reference/random/generated/numpy.random.randn.html>

<sup>4</sup><https://numpy.org/doc/stable/reference/random/generated/numpy.random.rand.html>

Table 1: Parameters used in the algorithm in the experiments

	$r$	$\lambda$	$\gamma$	max iteration
synthetic data experiment 1	4	$10^{-6}$	10	1000
synthetic data experiment 2	8	$10^{-6}$	1.5	1000
swimmer	50	0.5	10	1000
Jasper experiment 1	64	40000	10000	2000
Jasper experiment 2	100	1000	0.001	1000
Jasper experiment 3	20	1000000	1000000	1000
Urban	20	1000000	1000000	1000

**Result** Fig. 1 shows the result of the reconstruction. The reconstruction provided by SON-NMF fits better than the one provided by NMF. Fig. 2 shows the convergence speed of solving the problem using BCD with proximal averaging on solving W-subproblem, compared with the BCD with ADMM and BCD with Nesterov’s smoothing.

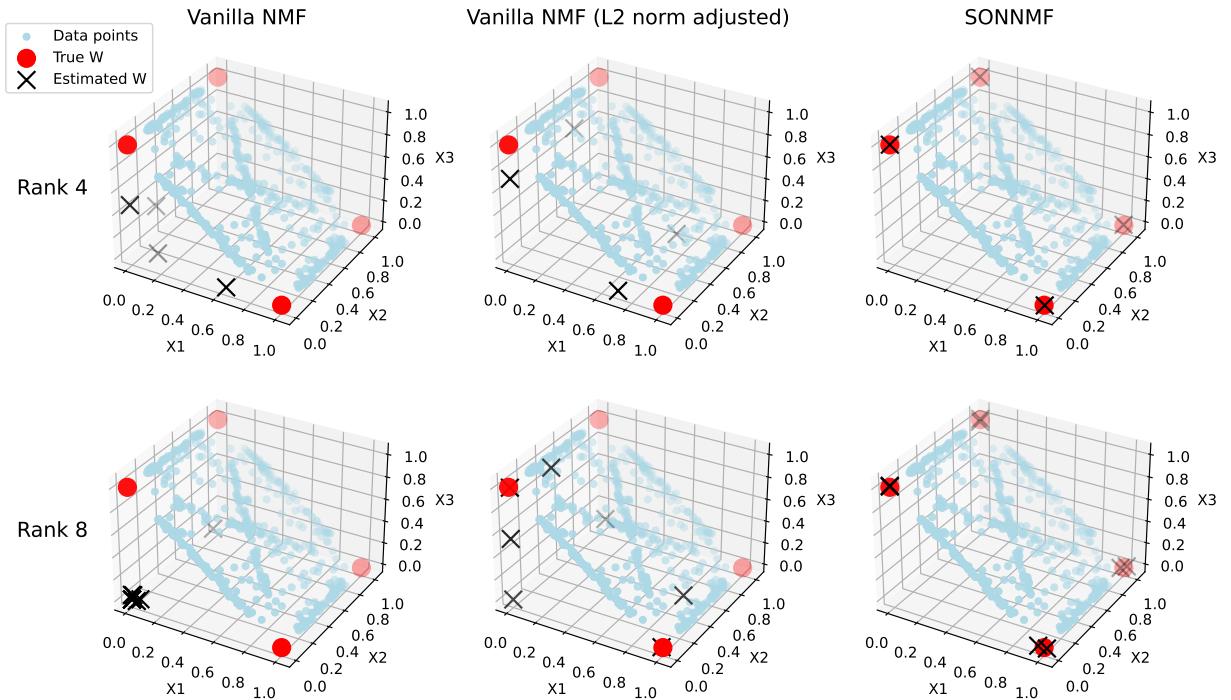


Figure 1: The reconstructed columns of  $\mathbf{W}$  (cross) by NMF and SON-NMF together with the ground truth columns of  $\mathbf{W}$  (red dots). **Left:**  $\mathbf{W}$  given by NMF; **middle:**  $\mathbf{W}$  given by NMF, with column normalized to 1. **Right:**  $\mathbf{W}$  from SON-NMF. In both cases  $r = 4$  and  $r = 8$ , the crosses given by SON-NMF fit numerically with the red dots.

### 5.1.2 The swimmer dataset

Now we use the **swimmer** dataset<sup>5</sup> introduced by [48]. The dataset consists of 256 figures with each 20-by-11 pixel of a skeleton body “swimming”, see the top row of Fig. 3. By inspection, the dataset consists of a rank-17 NMF: 1 for the torso, 16 for the 4 limbs with each limb corresponding to 4 different movement. A rank-50 (with  $r = 50 > r^* = 17$ ) SON-NMF is used in this dataset and we successfully recover all the 17 components. The redundant components are all captured as noise with small energy. Furthermore, if we perform a simple greedy search to determine the columns of  $\mathbf{W}$  to be extracted, the right figure of Fig. 2 shows the score with a cut-off point exactly at  $r = 17$ .

<sup>5</sup>We use the version available at <https://gitlab.com/ngillis/nmfbook/>

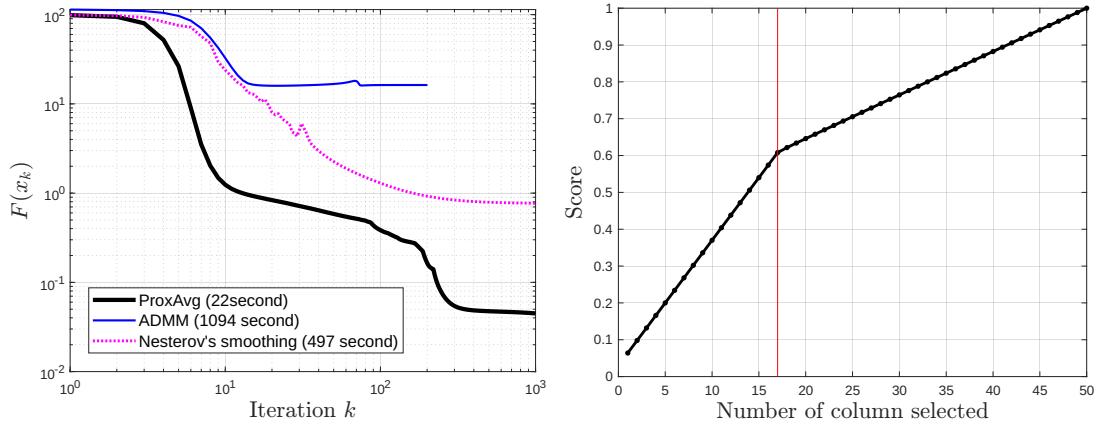


Figure 2: **Righ:** The convergence plot of SON-NMF cost function on synthetic data in experiment 2. Here we compare the convergence of three BCD algorithms with different method on solving the W-subproblem: proximal average (this work), ADMM and Nesterov’s smoothing. In the plot we also shows the computation time in second. The result here shows that compared with ADMM and smoothing, proximal average has the fastest convergence. **Right:** The score (the SON term) of selecting columns in  $\mathbf{W}$  on the swimmer dataset, based on simple greedy search. The red line  $r = 17$  indicates a cut-off point, which is exactly the number of component in the dataset.

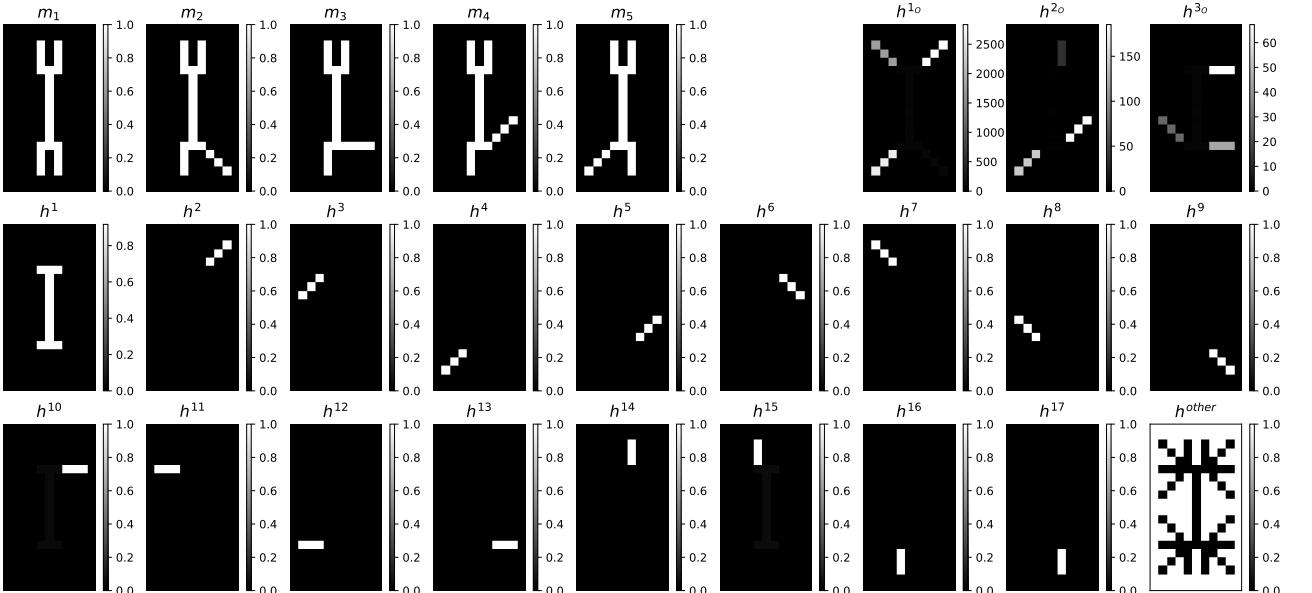


Figure 3: Top row, left: first 5 images ( $\mathbf{m}^1, \mathbf{m}^2, \dots, \mathbf{m}^5$ ) in the swimmer dataset, showing a swimmer “swimming”. Top row, right: 3  $\mathbf{h}_O^j$  obtained from rank-50 vanilla NMF, where the subscript in  $\mathbf{h}_O^j$  denotes the standard NMF. We can see that  $\mathbf{h}_O^j$  contains mixed result. Bottom rows: The decomposition result of rank-50 SON-NMF. Here  $\mathbf{h}^1$  captures the torso,  $\mathbf{h}^2, \mathbf{h}^3, \dots, \mathbf{h}^{17}$  capture the four limbs, and  $\mathbf{h}^{\text{other}}$ , which denotes the sum of all the other components, represent the noise, with a clear illustration that  $\mathbf{h}^{\text{other}}$  is complementary to all  $\mathbf{h}^1, \dots, \mathbf{h}^{17}$ . We remark that the  $\mathbf{w}$  corresponding to  $\mathbf{h}^{\text{other}}$  has a very small energy (not plotted here). For the full decomposition result of the vanilla NMF, see appendix.

### 5.1.3 Jasper ridge hyperspectral dataset

In this section we conduct experiment on the Jasper Ridge dataset<sup>6</sup>, which is a 100-by-100-by-198 dataset with pixel dimensions  $100 \times 100$  (number of pixels in each row and each column) and wavelength dimension of 198 (the dataset consists of 198 bandwidth of wavelengths). We refer to [2, Section 1.3.2] for the background of applying NMF on hyperspectral image. Fig.4 shows the photo of the Jasper Ridge and the three regions used in experiments. We remark that, due to the large numerical value of the entries of the dataset, we have to scale  $\lambda$

<sup>6</sup>From MATLAB <https://uk.mathworks.com/help/images/explore-hyperspectral-data-in-the-hyperspectral-viewer.html>

(the SON regularization parameter) to a large value (as shown in Table 1).

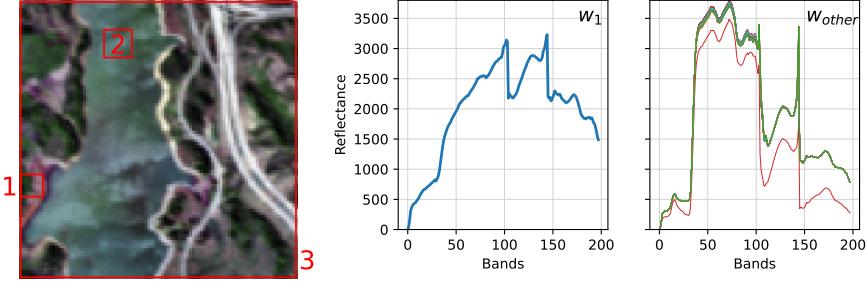


Figure 4: **Left:** The photo of the Jasper Ridge dataset, with the three regions of the dataset used in three experiments labeled in red. **Right:** Result for Jasper experiment 1. SON-NMF identifies the two material: soil ( $w_1$ ) and vegetation ( $w_{\text{other}}$ , refers to all the columns in  $\mathbf{W}$  except  $w_1$ ).

**Jasper experiment 1** We run a rank-64 SON-NMF on a 8-by-8 region consists of vegetation and soil. Here we use  $r = 64 = mn$  in SON-NMF, where  $r$  is as large as the size of the dataset. Fig.4 shows the matrix  $\mathbf{W}$  obtained from the SON-NMF. By inspection, region 1 consists of two end-member material: soil and vegetation. SON-NMF identified the two material, see Fig.4. This experiment showcases the ability of SON-NMF to correctly identify the correct number of components in the data without knowing the factorization rank.

**Jasper experiment 2** We run a rank-100 SON-NMF on a 10-by-10 water region. This region contains only water so it expected there is only one component in the decomposition. SON-NMF successfully identify the water component from the data and reduced a rank-100 NMF to a rank-1 NMF.

We remark that, by Perron-Frobenius theorem, the rank-1 solution here can also be obtained algebraically by the leading component in the eigendecomposition of the covariance matrix of the data. I.e., we have exact solution for rank-1 NMF by eigendecomposition, see the following proposition.

**Proposition 1.** Given a data matrix  $\mathbf{M} \in \mathbb{R}_+^{m \times n}$  and assume  $\mathbf{M} = \mathbf{WH}$  is the NMF of  $\mathbf{M}$ . Assume the columns of  $\mathbf{W}$ , denoted by  $\mathbf{w}_j$ , is ordered according to the norm of  $\mathbf{w}_j \mathbf{h}^j$  contributing to  $\mathbf{M}$ . Then, for the case  $r = 1$  (the data has a rank-1 NMF), the vector  $\mathbf{w}_1$  (the leading column of  $\mathbf{W}$ ) can be given by the leading eigenvector of the eigendecomposition of  $\mathbf{MM}^\top$ .

*Proof.* By  $\mathbf{M} = \mathbf{WH}$  we have  $\mathbf{MM}^\top = \mathbf{W}\mathbf{HH}^\top\mathbf{W}^\top = \mathbf{WG}\mathbf{W}^\top$  where  $\mathbf{G} := \mathbf{HH}^\top$ . Let the eigendecomposition of  $\mathbf{G}$  and  $\mathbf{MM}^\top$  as  $\mathbf{G} = \mathbf{V}\Sigma\mathbf{V}^\top$  and  $\mathbf{MM}^\top = \mathbf{U}\Lambda\mathbf{U}^\top$ . Then

$$\mathbf{WV}\Sigma\mathbf{V}^\top\mathbf{W}^\top = \mathbf{U}\Lambda\mathbf{U}^\top \implies \mathbf{WV} = \mathbf{U} \implies (\mathbf{WV})_{:,1} = \mathbf{U}_{:,1} \iff \mathbf{Wv}_1 = \mathbf{u}_1.$$

Both  $\mathbf{G} = \mathbf{HH}^\top$  and  $\mathbf{MM}^\top$  are nonnegative square matrices, by Perron-Frobenius theorem, both  $\mathbf{u}_1$  and  $\mathbf{v}_1$  are nonnegative vectors. Thus  $\mathbf{Wv}_1 = \mathbf{u}_1$  means  $\mathbf{u}_1 \in \text{cone}(\mathbf{W})$ . Lastly  $\mathbf{u}_1 = \mathbf{w}_1$  if  $\text{rank}(\mathbf{W}) = 1$ .  $\square$

Fig.5 shows that the result obtained from SON-NMF agree with the exact solution provided by eigendecomposition, and has a relative error of 0.006.

**Jasper experiment 3** In this experiment, we run a rank-20 SON-NMF on the whole Jasper Ridge dataset. Four material are extracted, see Fig.6. The materials extracted agree with the results obtained from other methods.

#### 5.1.4 The Urban hyperspectral dataset

In this section we conduct experiment on big data with  $1.5 \times 10^7$  data points. We use a dataset named Urban<sup>7</sup> that is a 307-by-307-by-162 data cube with pixel dimensions 307-by-307 (number of pixels in each row and each column) and wavelength dimension of 162 (the dataset consists of 162 bandwidth of wavelengths). We run a rank-20 SON-NMF with the following parameters:  $\lambda = \gamma = 10^6$ . We run at most 1000 iterations. SON-NMF successfully identified 5 clusters of material, see Fig.7.

<sup>7</sup>Available at <https://gitlab.com/ngillis/nmfbook/>

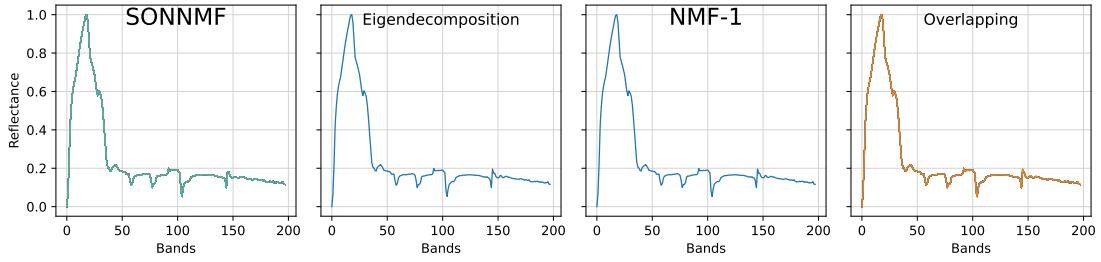


Figure 5: Result for Jasper experiment 2. The rank-100 SON-NMF (with  $r = 100$  that is much larger than the ground truth  $r^*$ ) identifies the water spectrum in the decomposition. **Left:** the plotting of all the 100 columns in  $\mathbf{w}$  share the same waveform. The middle two figures: the  $\mathbf{W}$  obtained by eigendecomposition and rank-1 vanilla NMF. **Right:** the plot of overlapping all the  $\mathbf{W}$ , showing that SON-NMF is producing result agreeing with vanilla NMF. For clarification we have normalized all the  $\mathbf{w}$  here to unit  $\ell_\infty$ -norm.

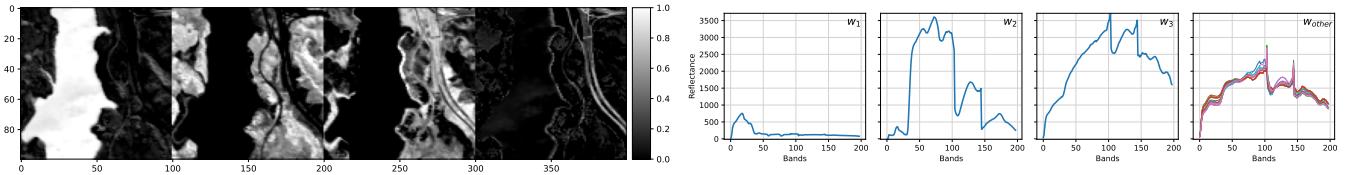


Figure 6: Result for Jasper experiment 3. Four material are extracted: (from left to right) water, vegetation, soil and road.

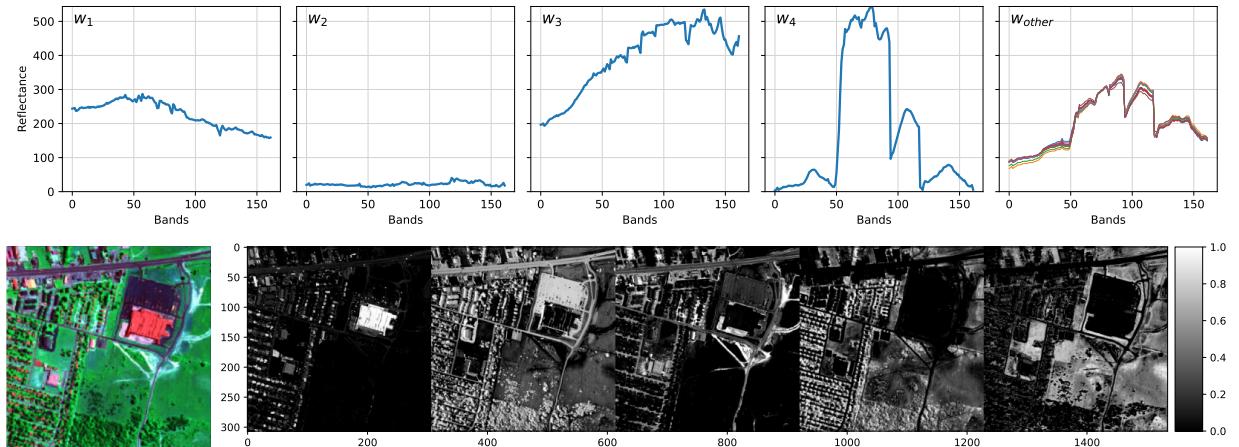


Figure 7: The reconstruction of a rank-20 SON-NMF on the Urban dataset. Here SON-NMF identified 5 clusters of material: (from left to right) roof, asphalt, soil, tree, and grass. Bottom left: the photo of the Urban dataset. We remark that the weak component asphalt in the dataset, is extracted by the SON-NMF. This is not the case by the classical NMF or with other rank estimation approach.

## 5.2 Speed of the algorithm

In Fig. 2 we showed the convergence of the BCD (Algorithm 1) with proximal average on solving the  $\mathbf{W}$ -subproblem (Algorithm 3) compared with BCD with ADMM to solve the  $\mathbf{W}$ -subproblem and BCD with Nesterov's smoothing to solve the  $\mathbf{W}$ -subproblem. The result shown in Fig. 2 tells that proximal average has the best performance. We refer the reader to [25] for the discussion why proximal average performs better than smoothing. In the following we discuss why proximal average perform much better than ADMM.

**Why ADMM is not suitable for SON-NMF: expensive per-iteration cost** Problem (8) with problem size  $n \times 1$  can be solved by multi-block ADMM, which introduces  $N$  auxiliary variables and  $N$  Lagrangian multipliers, and the augmented Lagrangian has a problem size of  $n \times (1 + 2N)$ . Such explosion of size makes the ADMM expensive for designing fast algorithm. To be exact,  $\mathbf{W} \mapsto P(\mathbf{W})$  is a  $m$ -by- $r$  to  $m$ -by- $r(r - 1)/2$  mapping, i.e., there are many nonsmooth terms  $\|\mathbf{w}_i - \mathbf{w}_j\|_2$  in SON. For each  $\mathbf{w}_i$ , the number of non-smooth terms in the optimization subproblem is  $r$ , and thus for the multi-block ADMM, the per-iteration complexity for each subproblem is  $m(1 + 2r)$ , and for all the  $r$  columns in  $\mathbf{W}$ , the multi-block ADMM has a per-iteration complexity

of  $\mathcal{O}(2mr^2 + mr)$ . In contrast, proximal-average has a per-iteration complexity of  $\mathcal{O}(mr)$ . In this work  $r$  is possibly as large as  $m$ , hence a per-iteration complexity of  $\mathcal{O}(2mr^2 + mr)|_{r=m} = \mathcal{O}(2m^3 + m^2)$  for ADMM is very expensive for solving the  $\mathbf{W}$ -subproblem, compared to a  $\mathcal{O}(mr)|_{r=m} = \mathcal{O}(m^2)$  cost for proximal-average. Furthermore, it is well known that ADMM has a slow convergence and therefore it may take even more iterations to solve SON-NMF.

### 5.3 Discussion: favourable features of SON-NMF for applications

Lastly we discuss favourable features of SON-NMF for applications that we have shown or observed.

**Is empirically rank-revealing** All the seven experiments in section 5 shows that SON-NMF can effectively learn the rank of the NMF without prior knowledge.

**Can deal with rank deficiency** SON-NMF is especially good at dealing with dataset with rank deficiency. This ability is not presented in othe regularized NMF model such as minvol NMF [3], which also have an empirically rank-revealing ability.

**Can detect weak component in the dataset** Due to the clustering nature of the SON term, SON-NMF is better at detecting weak component in the dataset than the vanilla NMF.

- In the Jasper dataset in section 5, the water component has small energy relative to other components: it only contribute to  $\|\mathbf{w}_{\text{water}} \mathbf{h}^{\text{water}}\|_F / \|\mathbf{M}\|_F = 9\%$  energy in the dataset, compared with 54% for the vegetation (tree/grass) component.
- The squared-F-norm in the expression  $\|\mathbf{M} - \mathbf{WH}\|_F^2$  will raise the importance of the large energy component in NMF, and thus making the algorithm emphasizing large component in the iteration and ignoring the weak component.

Thus, with  $r = 4$ , the vanilla NMF failed to extract the water component (see the full result in the Appendix). However, for SON-NMF, as the cost function contains the term  $\|\mathbf{w}_{\text{other}} - \mathbf{w}_{\text{water}}\|_2$ , SON-NMF will extract the water component.

The ability of SON-NMF to extract weak component is also supported by Lemma 1 that the smallest possible cluster size of any cluster identified in the SON term is bounded below by 1.

**Can handle spectral variability** Note that the solutions of SON-NMF hyperspectral images (i.e., the  $\mathbf{W}$  plots in both Fig.4, Fig.5, Fig.6, Fig.7) exhibit the phenomenon of spectral variability [49], and hence we argue that SON-NMF can be potentially useful in hyperspectral imaging: instead of using a sophisticated data processing pipeline as described in [49], SON alone is enough to deal with the spectral variability.

**Is a hierarchically clustering** In the case of SON clustering, different values of the regularization parameter  $\lambda$  yield different numbers of clusters. This is beneficial for dataset that is hierarchically clustered, so that one value of  $\lambda$  yields the coarse clustering while another yields the finer clustering. In the experiments on hyperspectral images, different values of  $\lambda$  give different but useful results.

## 6 Conclusion

In this paper we proposed a sum-of-norm regularized NMF model, aimed at estimating the rank in NMF on-the-fly. The proposed SON-NMF is a nonconvex nonsmooth non-separable non-proximal optimization problem, and we develope a BCD algorithm with proximal-average for solving SON-NMF. Theoretically we show that the complexity of the SON term in SON-NMF is irreducible, meaning that the complexity of solving SON-NMF is possibly very high. This is expected since rank estimation is an NP-hard problem in NMF. Lastly we empirically show that SON-NMF is capable to detect the correct factorization rank in NMF, and potentially applicable to imaging applications with some favourable features.

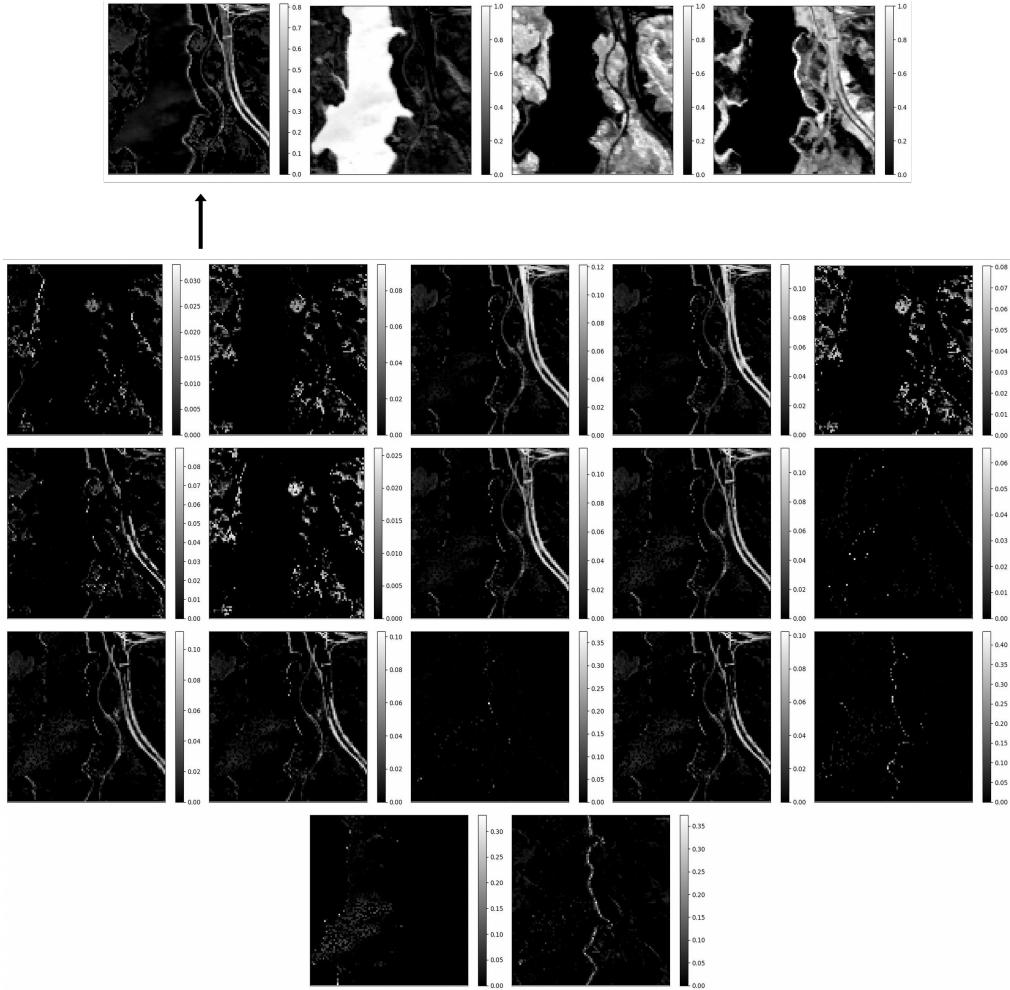


Figure 8: The full decomposition map of SON-NMF ( $r = 20$ ) on Jasper dataset (with  $r^*$  expected to be around 4). Here the road endmember consists of 17 components.

## Acknowledgement

Andersen Ang thanks Steve Vavasis for the discussion on graph theory and the complexity of SON-NMF.

## References

- [1] P. Paatero and U. Tapper, “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values,” *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.
- [2] N. Gillis, *Nonnegative matrix factorization*. SIAM, 2020.
- [3] V. Leplat, A. M. Ang, and N. Gillis, “Minimum-volume rank-deficient nonnegative matrix factorizations,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3402–3406, IEEE, 2019.
- [4] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.
- [5] V. Leplat, N. Gillis, and A. M. Ang, “Blind audio source separation with minimum-volume beta-divergence nmf,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 3400–3410, 2020.
- [6] M. S. Ang, “Nonnegative matrix and tensor factorizations: Models, algorithms and applications,” *Ph. D. thesis*, 2020.
- [7] S. A. Vavasis, “On the complexity of nonnegative matrix factorization,” *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1364–1377, 2010.

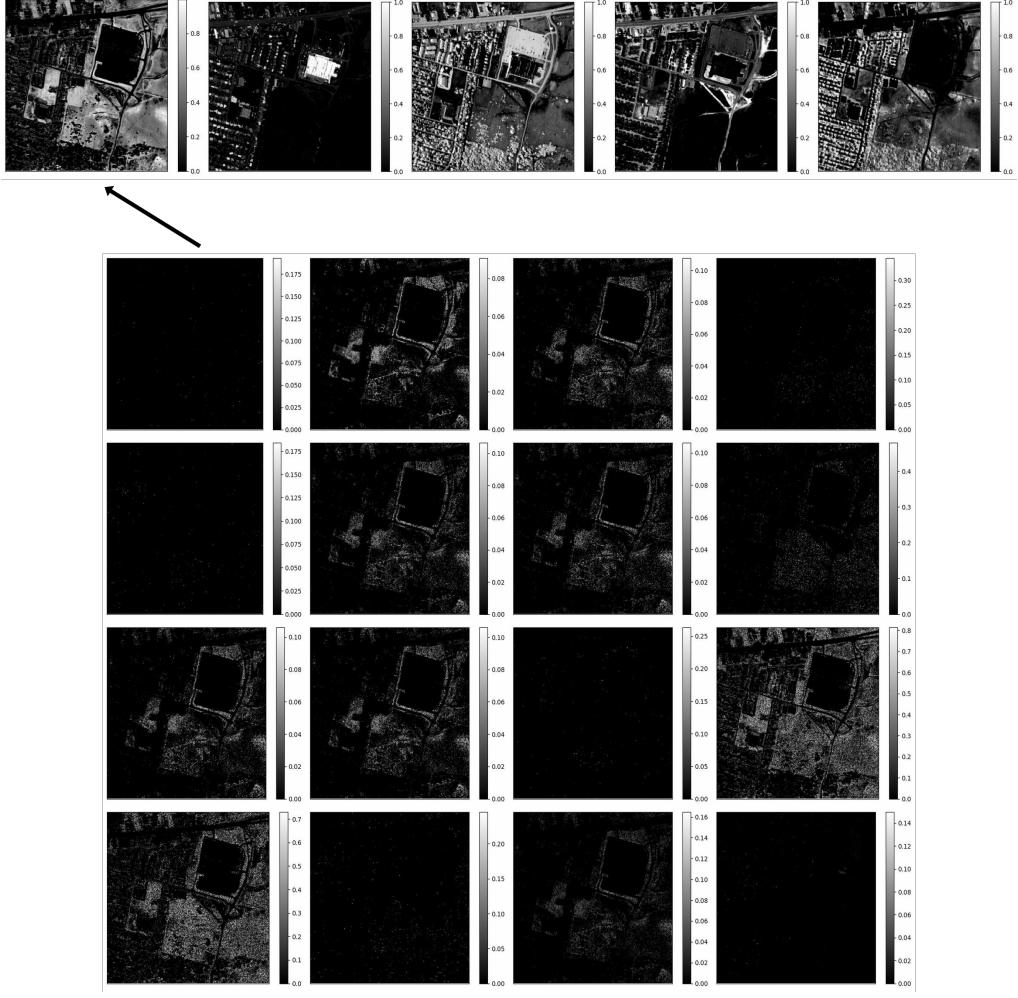


Figure 9: The full decomposition map of SON-NMF ( $r = 20$ ) on urban dataset (with  $r^*$  expected to be around 5). Here the grass endmember consists of 16 components.

- [8] M. Udell and A. Townsend, "Why are big data matrices approximately low rank?," *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 1, pp. 144–160, 2019.
- [9] V. Y. Tan and C. Févotte, "Automatic relevance determination in nonnegative matrix factorization with the  $\beta$ -divergence," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1592–1605, 2012.
- [10] F. Esposito, A. Boccarelli, and N. Del Buono, "An NMF-Based Methodology for Selecting Biomarkers in the Landscape of Genes of Heterogeneous Cancer-Associated Fibroblast Populations," *Bioinformatics and Biology Insights*, vol. 14, p. 1177932220906827, 2020.
- [11] S. Squires, A. Prügel-Bennett, and M. Niranjan, "Rank selection in nonnegative matrix factorization using minimum description length," *Neural computation*, vol. 29, no. 8, pp. 2164–2176, 2017.
- [12] J. E. Cohen and U. G. Rothblum, "Nonnegative ranks, decompositions, and factorizations of nonnegative matrices," *Linear Algebra and its Applications*, vol. 190, pp. 149–168, 1993.
- [13] J. Dewez, N. Gillis, and F. Glineur, "A geometric lower bound on the extension complexity of polytopes based on the f-vector," *Discrete Applied Mathematics*, vol. 303, pp. 22–38, 2021.
- [14] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.
- [15] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint  $\ell_2$ , 1-norms minimization," *Advances in neural information processing systems*, vol. 23, 2010.

- [16] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using l21-norm," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 673–682, 2011.
- [17] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1548–1560, 2010.
- [18] D. E. Farrar and R. R. Glauber, "Multicollinearity in regression analysis: the problem revisited," *The Review of Economic and Statistics*, pp. 92–107, 1967.
- [19] M. A. Ang and N. Gillis, "Volume regularized non-negative matrix factorizations," in *2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–5, IEEE, 2018.
- [20] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Mathematical Programming*, vol. 117, pp. 387–423, 2009.
- [21] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.
- [22] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.
- [23] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.
- [24] H. Le, N. Gillis, and P. Patrinos, "Inertial block proximal methods for non-convex non-smooth optimization," in *International Conference on Machine Learning*, pp. 5671–5681, PMLR, 2020.
- [25] Y.-L. Yu, "Better approximation and faster algorithm using the proximal average," *Advances in neural information processing systems*, vol. 26, 2013.
- [26] H. H. Bauschke, R. Goebel, Y. Lucet, and X. Wang, "The proximal average: basic theory," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 766–785, 2008.
- [27] A. M. S. Ang and N. Gillis, "Algorithms and comparisons of nonnegative matrix factorizations with volume regularization for hyperspectral unmixing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 4843–4853, 2019.
- [28] K. Pelckmans, J. De Brabanter, J. A. Suykens, and B. De Moor, "Convex clustering shrinkage," in *PASCAL workshop on statistics and optimization of clustering workshop*, 2005.
- [29] F. Lindsten, H. Ohlsson, and L. Ljung, "Clustering using sum-of-norms regularization: With application to particle filter output computation," in *2011 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 201–204, IEEE, 2011.
- [30] T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert, "Clusterpath: an algorithm for clustering using convex fusion penalties," in *28th international conference on machine learning*, p. 1, 2011.
- [31] L. Niu, R. Zhou, Y. Tian, Z. Qi, and P. Zhang, "Nonsmooth penalized clustering via  $\ell_p$  regularized sparse regression," *IEEE transactions on cybernetics*, vol. 47, no. 6, pp. 1423–1433, 2016.
- [32] T. Jiang and S. Vavasis, "Certifying clusters from sum-of-norms clustering," *arXiv preprint arXiv:2006.11355*, 2020.
- [33] X. Huang, A. Ang, J. Zhang, and Y. Wang, "Inhomogeneous graph trend filtering via a  $\ell_{2,0}$  cardinality penalty," *arXiv preprint arXiv:2304.05223*, 2023.
- [34] A. Beck, *First-order methods in optimization*. SIAM, 2017.

- [35] Y. Yuan, D. Sun, and K.-C. Toh, "An efficient semismooth newton based algorithm for convex clustering," in *International Conference on Machine Learning*, pp. 5718–5726, PMLR, 2018.
- [36] J. Krarup and S. Vajda, "On torricelli's geometrical solution to a problem of fermat," *IMA Journal of Management Mathematics*, vol. 8, no. 3, pp. 215–224, 1997.
- [37] N. M. Nam, N. T. An, R. B. Rector, and J. Sun, "Nonsmooth algorithms and nesterov's smoothing technique for generalized fermat–torricelli problems," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 1815–1839, 2014.
- [38] G. Cantor, "Ueber eine elementare frage der mannigfaltigketislehre.," *Jahresbericht der Deutschen Mathematiker-Vereinigung*, vol. 1, pp. 72–78, 1890.
- [39] C. Hildreth, "A quadratic programming procedure," *Naval research logistics quarterly*, vol. 4, no. 1, pp. 79–85, 1957.
- [40] S. J. Wright, "Coordinate descent algorithms," *Mathematical programming*, vol. 151, no. 1, pp. 3–34, 2015.
- [41] L. Condat, "Fast projection onto the simplex and the l1 ball," *Mathematical Programming*, vol. 158, no. 1-2, pp. 575–585, 2016.
- [42] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87. Springer Science & Business Media, 2003.
- [43] M. Schmidt, N. Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," *Advances in neural information processing systems*, vol. 24, 2011.
- [44] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, pp. 127–152, 2005.
- [45] G. B. Passty, "Ergodic convergence to a zero of the sum of monotone operators in hilbert space," *Journal of Mathematical Analysis and Applications*, vol. 72, no. 2, pp. 383–390, 1979.
- [46] M. Fukushima and H. Mine, "A generalized proximal point algorithm for certain non-convex minimization problems," *International Journal of Systems Science*, vol. 12, no. 8, pp. 989–1000, 1981.
- [47] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale modeling & simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.
- [48] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?," *Advances in neural information processing systems*, vol. 16, 2003.
- [49] R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, C. Richard, J. Chanussot, L. Drumetz, J.-Y. Tourneret, A. Zare, and C. Jutten, "Spectral variability in hyperspectral data unmixing: A comprehensive review," *IEEE geoscience and remote sensing magazine*, vol. 9, no. 4, pp. 223–270, 2021.

## Additional experimental results

**Vanilla NMF on the swimmer dataset** Fig. 10 shows the decomposition result of swimmer dataset by rank-50 vanilla NMF.

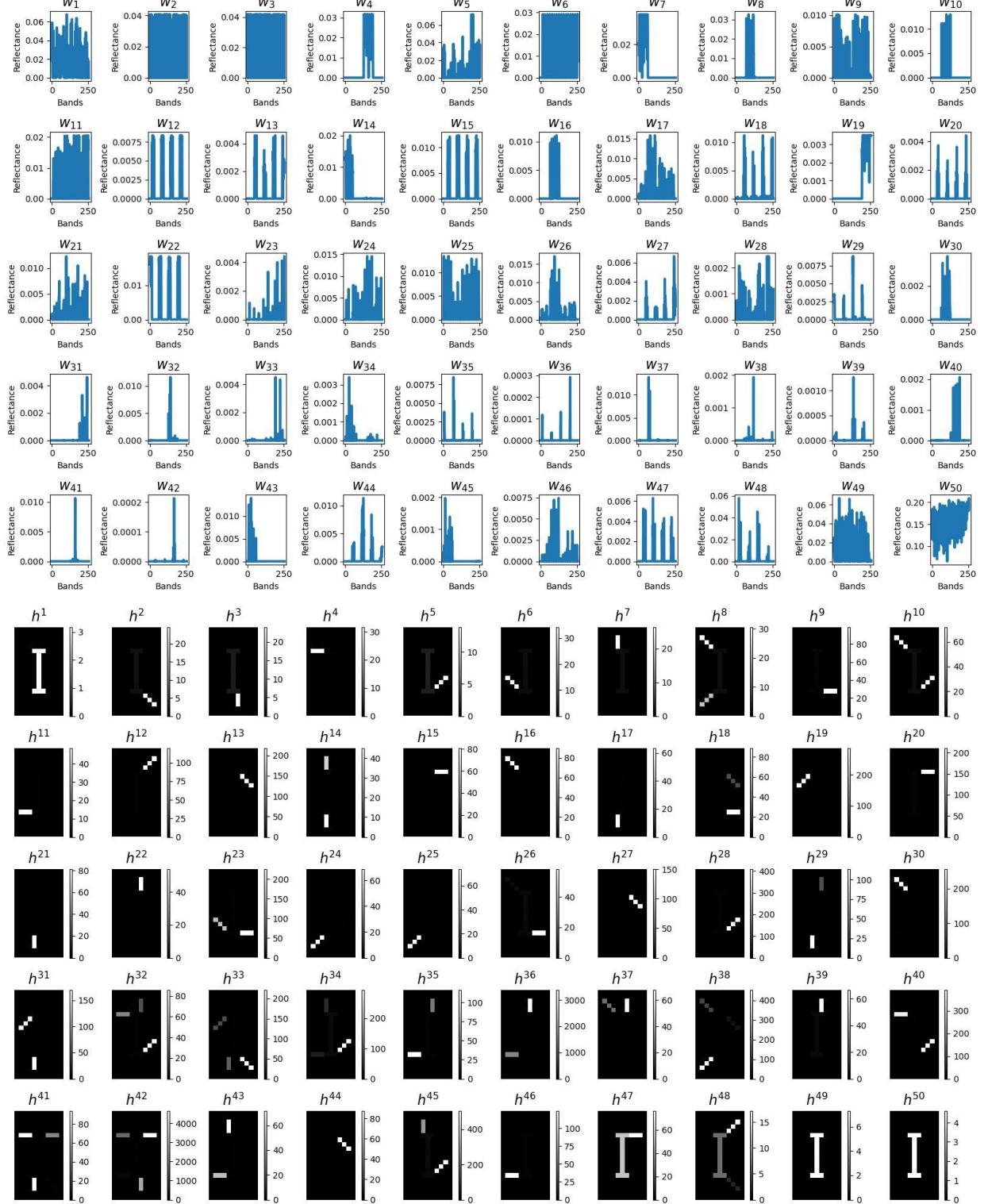


Figure 10: The decomposition result of the swimmer dataset by rank-50 vanilla NMF. We can see that the result do not produce component-wise decomposition, the limbs and torsos are mixed. Furthermore, each limb is represented by several component. For example,  $h^{17}$  and  $h^{21}$  represent the same left leg.

**Vanilla NMF on the Jasper dataset** Fig. 11 shows the decomposition result of the full Jasper dataset by rank-20 vanilla NMF.

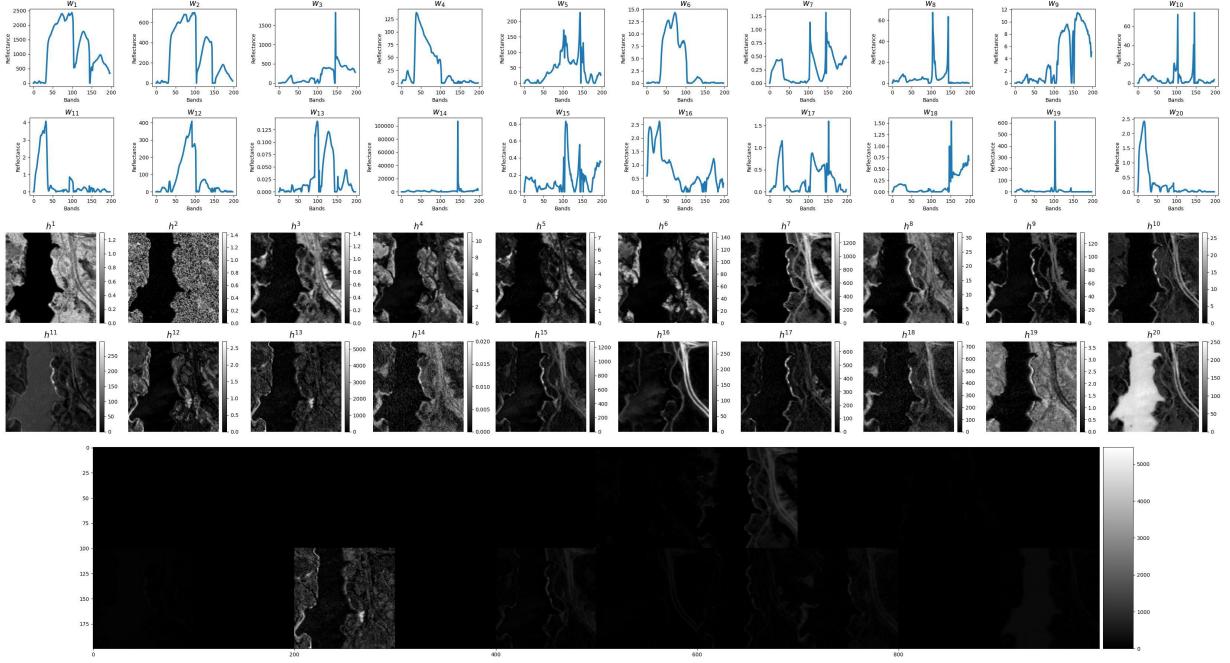


Figure 11: The decomposition result of the full Jasper dataset by rank-20 vanilla NMF. We can see that the result do not produce naturally-looking spectrum for  $W$  for many components. Compared with SON-NMF, the water component is not separated from other components, meaning that vanilla NMF failed to separate water component from other material.

**Vanilla NMF on the Urban dataset** Fig. 12 shows the decomposition result of the full Urban dataset by rank-20 vanilla NMF.

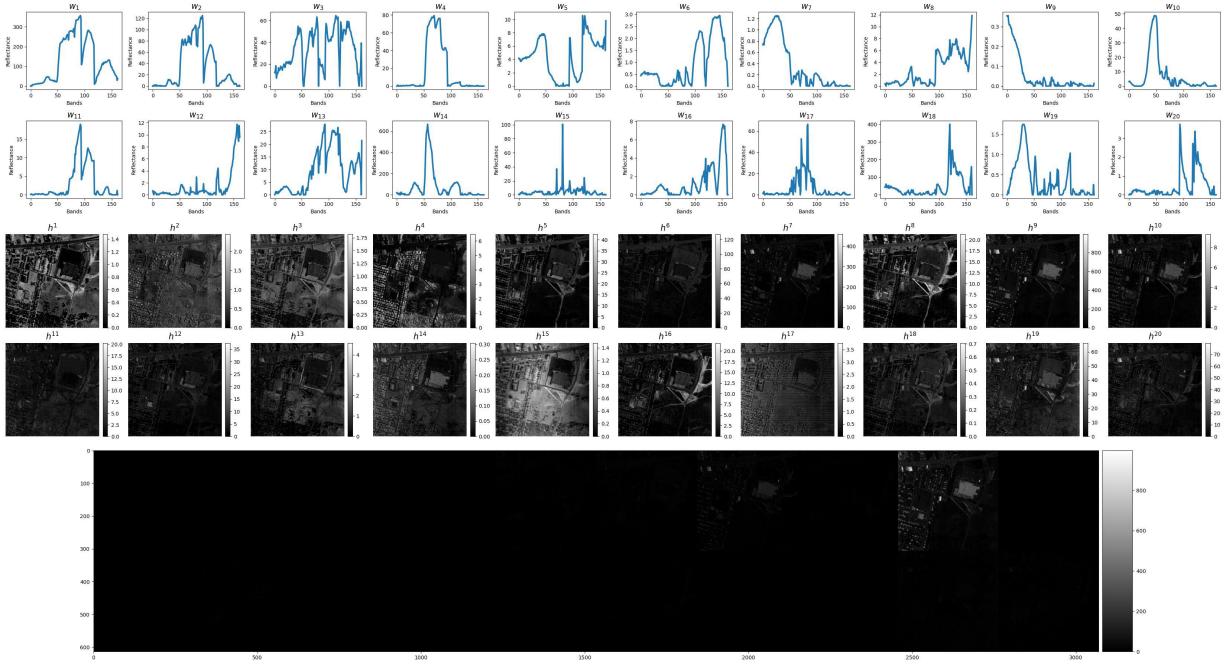


Figure 12: The decomposition result of the full Urban dataset by rank-20 vanilla NMF. We can see that the result do not produce naturally-looking spectrum for  $W$  for many components.