Refinement of Hottopixx Method for Nonnegative Matrix Factorization Under Noisy Separability

Tomohiko Mizutani *

January 11, 2022

Abstract

Hottopixx, proposed by Bittorf et al. at NIPS 2012, is an algorithm for solving nonnegative matrix factorization (NMF) problems under the separability assumption. Separable NMFs have important applications, such as topic extraction from documents and unmixing of hyperspectral images. In such applications, the robustness of the algorithm to noise is the key to the success. Hottopixx has been shown to be robust to noise, and its robustness can be further enhanced through postprocessing. However, there is a drawback. Hottopixx and its postprocessing require us to estimate the noise level involved in the matrix we want to factorize before running, since they use it as part of the input data. The noise-level estimation is not an easy task. In this paper, we overcome this drawback. We present a refinement of Hottopixx and its postprocessing that runs without prior knowledge of the noise level. We show that the refinement has almost the same robustness to noise as the original algorithm.

Keywords: nonnegative matrix factorization, separability, robustness to noise, linear programming

1 Introduction

Let $\mathbb{R}_+^{d \times n}$ denote the set of all nonnegative matrices of size $d \times n$. We are given $\boldsymbol{V} \in \mathbb{R}_+^{d \times n}$ and the factorization rank r. The nonnegative matrix factorization (NMF) problem asks us to find the factors $\boldsymbol{W} \in \mathbb{R}_+^{d \times r}$ and $\boldsymbol{H} \in \mathbb{R}_+^{r \times n}$ of $\boldsymbol{V}$ minimizing the gap between $\boldsymbol{V}$ and the product $\boldsymbol{WH}$. NMFs have many applications in diverse fields and thus have drawn the attention of researchers and practitioners. The problem is that the computation is intractable; it was shown to be NP-hard by Vavasis [19].

Arora et al. [3] further investigated the complexity of the NMF problem. They proposed to use an assumption, called separability, to remedy the issue. The notion of separability was originally introduced by Donoho and Stodden in [6] as a way of discussing the uniqueness of NMFs. Arora et al. showed that, if we place the separability assumption on the input matrix $\boldsymbol{V}$, then the NMF problem turns out to be tractable; we can find the factors $\boldsymbol{W}$ and $\boldsymbol{H}$ without much effort such that $\boldsymbol{V} = \boldsymbol{WH}$. Let us say that a matrix is separable if it satisfies the separability assumption. The application range of separable NMFs is restricted in comparison to NMFs, but they have still

---

*Department of Mathematical and Systems Engineering, Shizuoka University, 3-5-1 Johoku, Naka, Hamamatsu, 432-8561, Japan. `mizutani.t@shizuoka.ac.jp`

important applications, such as topic extraction from documents [4, 2] and unmixing of hyperspectral images [15, 16]. Other applications can be found in [7, 12]. So far, several algorithms have been developed for solving separable NMF problems. Separable matrices arising from applications should be perturbed by noise. Hence, it is desirable that an algorithm is robust against noise; even if noise is added to a separable matrix, the algorithm should be able to find the factors whose product well approximates the noisy separable matrix.

Bittorf et al. [5] proposed an algorithm, referred to as Hottopixx, for separable NMF problems. Their development is based on the observation that a certain feature of a separable matrix can be captured using linear programming (LP), and an optimal solution of the LP serves as a guide for solving the separable NMF problem. They showed that Hottopixx is robust to noise. Their result needs somewhat strong assumption. Roughly speaking, they assume that the columns of the separable matrix do not overlap. The assumption is not reasonable when dealing with applications such as topic extraction from documents and unmixing of hyperspectral images. Gillis [9] pointed out this issue and suggested a resolution. He developed postprocessing for Hottopixx and showed that with it Hottopixx is robust to noise without the assumption Bittorf et al. [5] put.

There is a drawback with Hottopixx and its postprocessing. They require three input data: a noisy separable matrix, the factorization rank, and the noise level. In the applications we mentioned above, we often encounter the situation in which the factorization rank can be estimated in advance. Meanwhile, it is unlikely that the noise level can be estimated in advance; we thus need to estimate it and its estimation is not an easy task. For that reason, most of the algorithms for solving separable NMF problems, such as VCA [18], SPA [15], SNPA [10] and ER [17], are designed to receive two input data: a noisy separable matrix and the factorization rank. Several drawbacks of Hottopixx are listed by Gillis and Luce in [13] and the drawback we mentioned above is one of them.

The main contribution of this paper is to overcome the drawback. We present a refinement of Hottopixx and its postprocessing that takes a noisy separable matrix and the factorization rank as input, but does not need prior knowledge of the noise level. We show that the refinement has almost the same robustness to noise as the original algorithm. The results are summarized in Theorems 1 and 2 of Section 3. In addition, we demonstrate in experiments the effectiveness of our refinement.

This paper is organized as follows. In Section 2, we formulate the separable NMF problem and explain the assumption and parameters used in our analysis. Section 3 presents the main results and compares them with the results of previous studies. Sections 4 and 5 describe the proposed algorithms and examine their robustness to noise; the refinement of Hottopixx is in Section 4 and the refinement of postprocessing is in Section 5. Section 6 describes experiments.

1.1 Notation and Symbols

We write $\mathbf{0}$ for a vector of all zeros, $\mathbf{1}$ for a vector of all ones, $\boldsymbol{e}_i$ for the ith unit vector, and $\boldsymbol{I}$ for the identity matrix. The symbol $\mathbf{0}$ is also used for a matrix of all zeros; in particular, $\mathbf{0}_{m \times n}$ for an $m \times n$ matrix of all zeros.

The notation $\boldsymbol{a}(i)$ denotes the ith element of $\boldsymbol{a} \in \mathbb{R}^n$. Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. The rows, columns and elements are denoted as follows: $\boldsymbol{A}(i,:)$ for the ith row, $\boldsymbol{A}(:,j)$ or $\boldsymbol{a}_j$ for the jth column, and $\boldsymbol{A}(i,j)$ for the (i,j)th element. Let $I \subset \{1, \ldots m\}$ and $J \subset \{1, \ldots, n\}$. The notation $\boldsymbol{A}(I,:)$ denotes the submatrix obtained by eliminating rows $\boldsymbol{A}(i,:)$ for all indices i in the complement of I, and $\boldsymbol{A}(:,J)$ that by eliminating columns $\boldsymbol{A}(:,j)$ for all indices j in the complement of J.

The notation $\| \cdot \|_p$ denotes the L_p norm of a vector or a matrix, $\| \cdot \|_F$ the Frobenius norm of

a matrix, $\text{tr}(\cdot)$ the trace of a square matrix, and $\text{diag}(\cdot)$ a vector composed of diagonal elements of a square matrix. i.e., $\text{diag}(\boldsymbol{B}) = [\boldsymbol{B}(1,1), \ldots, \boldsymbol{B}(n,n)]^\top$ for $\boldsymbol{B} \in \mathbb{R}^{n \times n}$. For positive integers r and n, the symbol R denotes the set of consecutive integers from 1 to r, and N that from 1 to n. For $S \subset N$, we denote by S^c the complement of S. For $a, b \in \mathbb{R}$ with $a < b$, the notation (a,b) denotes the open interval $\{x \in \mathbb{R} : a < x < b\}$, and $[a,b]$ the closed interval $\{x \in \mathbb{R} : a \leq x \leq b\}$.

2 Problem and Preliminaries

Let $\boldsymbol{V} \in \mathbb{R}_+^{d \times n}$ have an exact NMF $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H}$ for $\boldsymbol{W} \in \mathbb{R}_+^{d \times r}$ and $\boldsymbol{H} \in \mathbb{R}_+^{r \times n}$. *Separability* assumes that it can be further written as

$$\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H} \text{ for } \boldsymbol{W} \in \mathbb{R}_+^{d \times r} \text{ and } \boldsymbol{H} = [\boldsymbol{I}, \bar{\boldsymbol{H}}]\boldsymbol{\Pi} \in \mathbb{R}_+^{r \times n} \tag{1}$$

where $\boldsymbol{I}$ is an $r \times r$ identity matrix, $\bar{\boldsymbol{H}}$ is an $r \times (n-r)$ nonnegative matrix, and $\boldsymbol{\Pi}$ is an $n \times n$ permutation matrix. When a nonnegative matrix is written in the form shown in (1), we say that it is *r-separable* or simply *separable*. Separability means that all columns of $\boldsymbol{W}$ appear in those of $\boldsymbol{V}$; that is, there is a map $\phi : R \to N$ such that $\boldsymbol{w}_j = \boldsymbol{v}_{\phi(j)}$ for each $j = 1, \ldots, r$. We call the matrix $[\boldsymbol{v}_{\phi(1)}, \ldots, \boldsymbol{v}_{\phi(r)}]$, which is equivalent to $\boldsymbol{W}$, the *basis* of $\boldsymbol{V}$: in particular, $\boldsymbol{v}_{\phi(j)}$ is the *basis column* and $\phi(j)$ the *basis index*. We call r the *factorization rank* of $\boldsymbol{V}$. We formulate the separable NMF problem as follows:

Problem 1. *Given a separable matrix $\boldsymbol{V}$ and factorization rank r, find the basis of $\boldsymbol{V}$.*

Separable matrices arising from applications would contain noise. *Noisy separability* assumes that $\boldsymbol{N} \in \mathbb{R}^{d \times n}$ is added to a separable matrix $\boldsymbol{V} \in \mathbb{R}_+^{d \times n}$ such that

$$\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N}.$$

We call $\boldsymbol{N}$ the *noise* added to the separable matrix $\boldsymbol{V}$. If a matrix is in the form above, we say that it is *noisy separable*. When dealing with applications, it is desirable that, even if a separable matrix contains noise, an algorithm for solving separable NMF problems can still find a near-basis. Given a noisy separable matrix $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N}$, we say that the algorithm is *robust to noise* if it can find the column index set J such that $\boldsymbol{A}(:, J)$ is close to the basis of $\boldsymbol{V}$.

Our analysis put the following assumption on a matrix $\boldsymbol{A}$.

Assumption 1. $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N} \in \mathbb{R}^{d \times n}$, *where $\boldsymbol{V} \in \mathbb{R}_+^{d \times n}$ is r-separable of the form $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H} = \boldsymbol{W}[\boldsymbol{I}, \bar{\boldsymbol{H}}]\boldsymbol{\Pi}$ shown in (1) and $\boldsymbol{N} \in \mathbb{R}^{d \times n}$ is noise. Moreover,*

(a) *every column of $\boldsymbol{V}, \boldsymbol{W}$ and $\boldsymbol{H}$ has unit L_1 norm, and*

(b) *the noise $\boldsymbol{N}$ satisfies $\|\boldsymbol{N}\|_1 \leq \epsilon$ for some real number ϵ satisfying $0 \leq \epsilon < 1$.*

We call ϵ the *noise level* involved in $\boldsymbol{A}$. As described in [3, 8], we can assume without loss of generality that part (a) holds. Our analysis uses parameters κ, ω and β, which were introduced by Gillis [9] for the analysis of Hottopixx. Let $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N} \in \mathbb{R}^{d \times n}$ where $\boldsymbol{V} \in \mathbb{R}_+^{d \times n}$ is r-separable of the form $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H} = \boldsymbol{W}[\boldsymbol{I}, \bar{\boldsymbol{H}}]\boldsymbol{\Pi}$ shown in (1) and $\boldsymbol{N} \in \mathbb{R}^{d \times n}$ is noise. The parameters κ and ω are defined in terms of $\boldsymbol{W}$ by

$$\kappa = \min_{1 \leq j \leq r} \min_{\boldsymbol{z} \geq \boldsymbol{0}} \|\boldsymbol{w}_j - \boldsymbol{W}(:, R \setminus \{j\})\boldsymbol{z}\|_1,$$

$$\omega = \min_{1 \leq j_1 \neq j_2 \leq r} \|\boldsymbol{w}_{j_1} - \boldsymbol{w}_{j_2}\|_1.$$

They satisfy the relation

$$\kappa \leq \omega. \tag{2}$$

It is easy to verify that it holds. Let $j_1, j_2 \in R$ with $j_1 \neq j_2$ satisfy $\omega = \|\boldsymbol{w}_{j_1} - \boldsymbol{w}_{j_2}\|_1$. Then, there exists an integer $\ell \in R$ such that $\boldsymbol{W}(:, R \setminus \{j_1\})\boldsymbol{e}_\ell = \boldsymbol{w}_{j_2}$. Hence,

$$\kappa \leq \|\boldsymbol{w}_{j_1} - \boldsymbol{W}(:, R \setminus \{j_1\})\boldsymbol{e}_\ell\|_1 = \|\boldsymbol{w}_{j_1} - \boldsymbol{w}_{j_2}\|_1 = \omega.$$

Let Assumption 1(a) hold. Then, we can bound κ and ω as

$$0 \leq \kappa \leq 1, \tag{3}$$
$$0 \leq \omega \leq 2. \tag{4}$$

The lower bounds come from the definitions of κ and ω. For the upper bounds, we find that $\kappa \leq \|\boldsymbol{w}_j\|_1 = 1$ for any $j \in R$, and $\omega \leq \|\boldsymbol{w}_{j_1} - \boldsymbol{w}_{j_2}\|_1 \leq \|\boldsymbol{w}_{j_1}\|_1 + \|\boldsymbol{w}_{j_2}\|_1 = 2$ for any different $j_1, j_2 \in R$. The parameter β is defined in terms of the submatrix $\bar{\boldsymbol{H}}$ of $\boldsymbol{H}$ by

$$\beta = \max_{1 \leq i \leq r, \ 1 \leq j \leq n-r} \bar{\boldsymbol{H}}(i,j).$$

Let Assumption 1(a) hold. Then, β satisfies $0 \leq \beta \leq 1$. In particular, if $\beta = 1$, there are columns of $\bar{\boldsymbol{H}}$ such that one element is 1 and the others are 0. This means that there are duplicate basis columns.

3 Main Results

Here, we present the main results in the form of Theorems 1 and 2. We refine Hottopixx of Bittorf et al. [5]. Our refinement uses the optimization model P, which is shown in Section 4.1. Algorithm 1 of the section describes the details of the refinement. Our first result, which states the robustness of Algorithm 1 to noise, is as follows:

Theorem 1. *Let $\boldsymbol{A}$ satisfy Assumption 1. Assume $\kappa > 0$. Run the refinement of Hottopixx, i.e., Algorithm 1, on the input $(\boldsymbol{A}, r)$. If*

$$\epsilon \leq \frac{\kappa(1-\beta)}{9(r+1)},$$

then, after suitably rearranging the columns of $\boldsymbol{W}$, the output $\boldsymbol{W}_{\text{out}}$ satisfies $\|\boldsymbol{W} - \boldsymbol{W}_{\text{out}}\|_1 \leq \epsilon$.

If the noise level ϵ is positive and the basis columns overlap, i.e., $\beta = 1$, the theorem is invalid and does not say anything about the robustness of Algorithm 1 to noise. To cope with this issue, we develop postprocessing that ensures the algorithm's robustness to noise even in such a case. Postprocessing for that purpose was proposed by Gillis [9], and here, we refine it. A detailed description of the refinement is given in Algorithm 2 of Section 5.1. Our second result, which states the robustness of Algorithm 2 to noise, is as follows:

Theorem 2. *Let $\boldsymbol{A}$ satisfy Assumption 1. Run the refinement of Hottopixx with postprocessing, i.e., Algorithm 2, on the input $(\boldsymbol{A}, r)$. If*

$$\epsilon < \frac{\kappa\omega}{578(r+1)},$$

then, after suitably rearranging the columns of $\boldsymbol{W}$, the output $\boldsymbol{W}_{\mathrm{out}}$ satisfies

$$\|\boldsymbol{W} - \boldsymbol{W}_{\mathrm{out}}\|_1 \le \frac{136(r+1)}{\kappa}\epsilon.$$

In particular, if

$$\epsilon < \frac{\kappa^2}{289(r+1)^2},$$

then, after suitably rearranging the columns of $\boldsymbol{W}$, the output $\boldsymbol{W}_{\mathrm{out}}$ satisfies

$$\|\boldsymbol{W} - \boldsymbol{W}_{\mathrm{out}}\|_1 \le 8\sqrt{\epsilon}.$$

Theorem 2 tells us that there is a range of noise intensity that Algorithm 2 is robust to even if there are duplicate basis columns. From the relation $\kappa \le \omega$ shown in (2), we can see that $\frac{\kappa^2}{289(r+1)^2} \le \frac{\kappa\omega}{578(r+1)}$ holds, and $\frac{\kappa^2}{289(r+1)^2}$ is $\frac{2}{r+1}$ times smaller than $\frac{\kappa\omega}{578(r+1)}$. The theorem tells us that, if ϵ satisfies $\epsilon \le \frac{\kappa\omega}{578(r+1)}$, the error of the output $\boldsymbol{W}_{\mathrm{out}}$ relative to the basis $\boldsymbol{W}$ can be bounded by using ϵ, r, κ; in particular, if ϵ is small and satisfies $\epsilon \le \frac{\kappa^2}{289(r+1)^2}$, the error bound depends on only ϵ. Note that it remains an open question how tight the bounds shown in Theorems 1 and 2 are. This is a topic for further research.

Now, let us review the previous work on Hottopixx and compare our results with the previous ones. Arora et al. [3] proposed the first algorithm with provable guarantees for solving separable NMF problems. Motivated by that work, Bittorf et al. [5] developed Hottopixx. Let $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N}$ where $\boldsymbol{V} \in \mathbb{R}_+^{d \times n}$ is r-separable of the form $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H}$ shown in (1) and $\boldsymbol{N} \in \mathbb{R}^{d \times n}$ is noise satisfying $\|\boldsymbol{N}\|_1 \le \epsilon$ for some nonnegative real number ϵ. Hottopixx is based on the optimization model Q and require $(\boldsymbol{A}, r, \epsilon)$ as its input. The details of the algorithm and Q are given in Section 4.1. Bittorf et al. showed that Hottopixx is robust to noise. However, it was unclear whether one can ensure its robustness in the case that there are duplicate basis columns.

Gillis [9] and Gillis and Luce [13] pursued a line of research that examined the robustness of Hottopixx. Tables 1 and 2 summarize their results as well as ours. The first column lists the input data of the algorithms; the second one lists the optimization model whose details are given in Section 4.1; the third one lists the assumptions imposed on the analysis; and the fourth and fifth ones list the robustness results obtained by the analysis, i.e., the bound on the noise level and the error of the output relative to the basis.

Gillis [9] investigated the robustness of Hottopixx. He started by analyzing the case where there are no duplicate basis columns. The analysis suggested that the use of postprocessing makes it possible to enhance its robustness. He then developed postprocessing and showed that Hottopixx with the postprocessing is robust to noise even when there are duplicate basis columns. The results on Hottopixx (Theorem 2.3 of [9]) are summarized in the second row of Table 1, and those on Hottopixx with the postprocessing (Theorem 3.5 of [9]) are in the second row of Table 2.

Gillis and Luce [13] developed a refinement of Hottopixx. Their refinement is based on the optimization model R, and it requires $(\boldsymbol{A}, \epsilon)$ as input. The details of the algorithm and R are given

Table 1: Comparison of our result (Theorem 1) with those of Gillis (Theorem 2.3 of [9]) and Gillis and Luce (Theorem 2 of [13]) for algorithms without postprocessing. The algorithm of Gillis and Luce uses a parameter ρ that is set to a positive real number.

	Input	Model	Assumption	Noise level	Error
Our result	$\boldsymbol{A}, r$	P	Assumption 1, $\kappa > 0$	$\frac{\kappa(1-\beta)}{9(r+1)}$	ϵ
Gillis	$\boldsymbol{A}, r, \epsilon$	Q	Assumption 1, $\kappa > 0$	$\frac{\kappa(1-\beta)}{9(r+1)}$	ϵ
Gillis and Luce	$\boldsymbol{A}, \epsilon$	R	Assumption 1, $\kappa > 0$	$\frac{\kappa(1-\beta)\min\{1,\rho\}}{5(\rho+2)}$	ϵ

Table 2: Comparison of our result (Theorem 2) with those of Gillis (Theorem 3.5 of [9]) and Gillis and Luce (Theorem 7 of [13]) for algorithms with postprocessing.

	Input	Model	Assumption	Noise level	Error
Our result	$\boldsymbol{A}, r$	P	Assumption 1	$\frac{\kappa\omega}{578(r+1)}$	$\frac{136(r+1)}{\kappa}\epsilon$
Gillis	$\boldsymbol{A}, r, \epsilon$	Q	Assumption 1	$\frac{\kappa\omega}{99(r+1)}$	$\frac{49(r+1)}{\kappa}\epsilon + 2\epsilon$
Gillis and Luce	$\boldsymbol{A}, r, \epsilon$	R	Assumption 1	$\frac{\kappa\omega}{99(r+1)}$	$\frac{49(r+1)}{\kappa}\epsilon + 2\epsilon$

in Section 4.1. They showed that the refinement is robust to noise. The results (Theorem 2 of [13]) are summarized in the third row of Table 1. Here, ρ is a parameter that is set to a positive real number. The advantage of the refinement over Hottopixx is that it does not require prior knowledge of the factorization rank r of the matrix $\boldsymbol{A}$ we want to factorize, and the robustness result does not depend on r. They also incorporated the postprocessing of Gillis [9] into the refinement, and showed that the same result as Theorem 3.5 of [9] holds for the refinement with the postprocessing. The results (Theorem 7 of [13]) are summarized in the third row of Table 2.

Let us compare our results with those of Gillis [9] and Gillis and Luce [13]. We can see from Tables 1 and 2 that Algorithm 1 is as robust as Hottopixx, and Algorithm 2 is almost as robust as Hottopixx and the refinement of Gillis and Luce with the postprocessing of Gillis. The assumptions of our analysis are the same as theirs. There is a difference in the input data: $(\boldsymbol{A}, r)$ for our algorithms and $(\boldsymbol{A}, r, \epsilon)$ or $(\boldsymbol{A}, \epsilon)$ for the existing algorithms. We often encounter a situation in which the factorization rank r is available in advance in applications such as topic extraction from documents and unmixing of hyperspectral images. Hence, it is reasonable to assume that a noisy separable matrix $\boldsymbol{A}$ and the factorization rank r will be given as input. As mentioned in Section 1, most of the algorithms for solving separable NMF problems are designed to take $(\boldsymbol{A}, r)$ as input. The advantage of our algorithms over the existing ones is they run on $(\boldsymbol{A}, r)$ that does not include prior knowledge of the noise level ϵ and yet have almost the same robustness to noise as the existing ones.

4 Refinement of Hottopixx

4.1 Algorithm

Our refinement of Hottopixx is described in Algorithm 1. For the input $\boldsymbol{A}$ and r, step 2 constructs

Algorithm 1 Refinement of Hottopixx

Input: $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ and a positive integer r.
Output: $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{d \times r}$.

1. If there are duplicate columns in $\boldsymbol{A}$, keep one of them and remove all the rest.

2. Compute the optimal solution $\boldsymbol{X}_{\text{opt}}$ of the problem $\mathsf{P}(\boldsymbol{A}, r)$. Set $\boldsymbol{p} = \text{diag}(\boldsymbol{X}_{\text{opt}})$.

3. Let $\boldsymbol{W}_{\text{out}} = \boldsymbol{A}(:, J)$ for the index set J corresponding to the r largest elements of $\boldsymbol{p}$, and return $\boldsymbol{W}_{\text{out}}$.

---

and solves the optimization problem with variable $\boldsymbol{X} \in \mathbb{R}^{n \times n}$,

$$
\mathsf{P}(\boldsymbol{A}, r): \quad \begin{aligned}
\text{Minimize} \quad & \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{X}\|_1 \\
\text{subject to} \quad & \text{tr}(\boldsymbol{X}) = r, \\
& \boldsymbol{X}(i, i) \leq 1 && \text{for all } i \in N, \\
& \boldsymbol{X}(i, j) \leq \boldsymbol{X}(i, i) && \text{for all } i, j \in N, \\
& \boldsymbol{X}(i, j) \geq 0 && \text{for all } i, j \in N.
\end{aligned}
$$

Throughout this paper, we use $\boldsymbol{X}_{\text{opt}}$ to denote the optimal solution and θ to denote the optimal value $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{X}_{\text{opt}}\|_1$. By introducing new variables $\boldsymbol{Y} \in \mathbb{R}^{d \times n}$ and $z \in \mathbb{R}$, the problem above can be reduced to an LP problem, since the minimization of $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{X}\|_1$ is equivalent to the minimization of z under the constraints: $-\boldsymbol{Y} \leq \boldsymbol{A} - \boldsymbol{A}\boldsymbol{X} \leq \boldsymbol{Y}$ and $\sum_{i=1}^{d} \boldsymbol{Y}(i, j) \leq z$ for all $j \in N$. We use P' to denote the LP problem. It should be noted that P' has $n^2 + dn + 1$ variables and $2n^2 + 2dn + n + 1$ constraints. Hence, the size of P' may be rather large. Step 1 performs the preprocessing on the input matrix. Although Hottopixx does not contain this step, Algorithm 1 must have it. See Remark 1 at the end of this section for the reason.

Here, let us recall Hottopixx of Bittorf et al. [5] and the refinement of Gillis and Luce [13]. Bittorf et al. looked at a certain feature of separable matrices and developed Hottopixx on the basis of that observation. Let $\boldsymbol{A}$ satisfy Assumption 1. Then, it can be written as $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N}$, where $\boldsymbol{V} \in \mathbb{R}_+^{d \times n}$ is r-separable of the form $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H} = \boldsymbol{W}[\boldsymbol{I}, \bar{\boldsymbol{H}}]\boldsymbol{\Pi}$ shown in (1) and $\boldsymbol{N} \in \mathbb{R}^{d \times n}$ is noise. Using an $n \times n$ permutation matrix $\boldsymbol{\Pi}$ and the $r \times (d - r)$ nonnegative matrix $\bar{\boldsymbol{H}}$, we construct the matrix

$$
\boldsymbol{X}_0 = \boldsymbol{\Pi}^{-1} \left[\begin{array}{c|c} \boldsymbol{I} & \bar{\boldsymbol{H}} \\ \hline \boldsymbol{0}_{(n-r) \times r} & \boldsymbol{0}_{(n-r) \times (n-r)} \end{array} \right] \boldsymbol{\Pi} \in \mathbb{R}^{n \times n} \tag{5}
$$

where $\boldsymbol{I}$ is an identity matrix of size r. We make the following observations:

- The basis of $\boldsymbol{V}$ can be identified by using $\boldsymbol{X}_0$, since the diagonal entries of $\boldsymbol{X}_0$ are 0 or 1 and the positions with 1 correspond to the basis indices of $\boldsymbol{V}$.

- $\boldsymbol{X}_0$ satisfies

$$
\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{X}_0\|_1 \leq 2\epsilon. \tag{6}
$$

The second observation comes from the fact that we have

$$\boldsymbol{V}\boldsymbol{X}_0 = \boldsymbol{W}[\boldsymbol{I},\bar{\boldsymbol{H}}]\boldsymbol{\Pi}\boldsymbol{\Pi}^{-1}\left[\begin{array}{c|c}\boldsymbol{I} & \bar{\boldsymbol{H}} \\ \hline \boldsymbol{0} & \boldsymbol{0}\end{array}\right]\boldsymbol{\Pi} = \boldsymbol{W}[\boldsymbol{I},\bar{\boldsymbol{H}}]\boldsymbol{\Pi} = \boldsymbol{V},$$

which gives

$$\begin{aligned}
\|\boldsymbol{A}-\boldsymbol{A}\boldsymbol{X}_0\|_1 = \|\boldsymbol{V}+\boldsymbol{N}-(\boldsymbol{V}+\boldsymbol{N})\boldsymbol{X}_0\|_1 &= \|\boldsymbol{N}-\boldsymbol{N}\boldsymbol{X}_0\|_1 \\
&\leq \|\boldsymbol{N}\|_1 + \|\boldsymbol{N}\|_1\|\boldsymbol{X}_0\|_1 \\
&\leq 2\epsilon \qquad\qquad\qquad \text{(by Assumption 1).}
\end{aligned}$$

To compute $\boldsymbol{X}_0$ approximately, Bittorf et al. proposed to solve an optimization problem with variable $\boldsymbol{X} \in \mathbb{R}^{n\times n}$,

$$\begin{aligned}
\mathsf{Q}(\boldsymbol{A},r,\epsilon): \quad &\text{Minimize} \quad &&\boldsymbol{f}^\top\text{diag}(\boldsymbol{X}) \\
&\text{subject to} \quad &&\|\boldsymbol{A}-\boldsymbol{A}\boldsymbol{X}\|_1 \leq 2\epsilon, \\
& &&\text{tr}(\boldsymbol{X}) = r, \\
& &&\boldsymbol{X}(i,i) \leq 1 && \text{for all } i \in N, \\
& &&\boldsymbol{X}(i,j) \leq \boldsymbol{X}(i,i) && \text{for all } i,j \in N, \\
& &&\boldsymbol{X}(i,j) \geq 0 && \text{for all } i,j \in N.
\end{aligned}$$

Here, $\boldsymbol{f}$ is a parameter set by the user: it can be chosen to be any n-dimensional vector with distinct elements. The problem Q can be reduced to an LP. Hottopixx is the same as performing steps 2 and 3 of Algorithm 1 with a replacement of $\mathsf{P}(\boldsymbol{A},r)$ in step 2 by $\mathsf{Q}(\boldsymbol{A},r,\epsilon)$. It thus requires $(\boldsymbol{A},r,\epsilon)$ as input.

Gillis and Luce [13] refined Hottopixx. They proposed to solve an optimization problem with variable $\boldsymbol{X} \in \mathbb{R}^{n\times n}$,

$$\begin{aligned}
\mathsf{R}(\boldsymbol{A},\epsilon): \quad &\text{Minimize} \quad &&\boldsymbol{g}^\top\text{diag}(\boldsymbol{X}) \\
&\text{subject to} \quad &&\|\boldsymbol{A}-\boldsymbol{A}\boldsymbol{X}\|_1 \leq \rho\epsilon, \\
& &&\boldsymbol{X}(i,i) \leq 1 && \text{for all } i \in N, \\
& &&\boldsymbol{X}(i,j) \leq \boldsymbol{X}(i,i) && \text{for all } i,j \in N, \\
& &&\boldsymbol{X}(i,j) \geq 0 && \text{for all } i,j \in N.
\end{aligned}$$

Here, $\boldsymbol{g}$ and ρ are parameters set by the user: $\boldsymbol{g}$ can be chosen to be any n-dimensional vector with distinct positive elements and ρ a positive value. As in the case of Q, the problem R can be reduced to an LP. Their algorithm computes the optimal solution of R and constructs an index set corresponding to diagonal entries larger than $1 - \frac{\min\{1,\rho\}}{2}$. Hence, it takes as input $(\boldsymbol{A},\epsilon)$ and does not require r as input.

Remark 1. *If Algorithm 1 does not contain step 1, it may fail to find a basis from separable matrices with duplicate basis columns. For instance, consider*

$$\boldsymbol{V} = \left[\begin{array}{ccccc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right].$$

This is 2-separable with $\beta = 1$ since it can be written as $\boldsymbol{V} = \boldsymbol{WH}$ by letting $\boldsymbol{W} = \boldsymbol{I}$ and $\boldsymbol{H} = \boldsymbol{V}$. Suppose that the algorithm receives $(\boldsymbol{A}, r)$ by letting $\boldsymbol{A} = \boldsymbol{V}$ and $r = 2$ as input. Consider the two matrices,

$$\boldsymbol{X}_1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad and \quad \boldsymbol{X}_2 = \begin{bmatrix} 1/3 & 0 & 1/3 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 1/2 \end{bmatrix}.$$

Both $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are optimal solutions of problem $\mathsf{P}(\boldsymbol{A}, r)$, since they satisfy all the constraints and $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{X}_1\|_1 = \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{X}_2\|_1 = 0$. If Algorithm 1 skips step 1 and finds $\boldsymbol{X}_2$ in step 2, then it constructs $J = \{2, 5\}$ in step 3. We have $\boldsymbol{W}_{\mathrm{out}} \neq \boldsymbol{W}$, since $\boldsymbol{W}_{\mathrm{out}} = \boldsymbol{A}(:, J) = \boldsymbol{V}(:, J)$ and $\boldsymbol{W} = \boldsymbol{I}$.

4.2 Analysis

The optimal value θ of problem P is related to the noise level ϵ involved in separable matrices. Actually, from the observation Bittorf et al. made in [5], we can easily see that $\theta \leq 2\epsilon$ holds.

Lemma 1. *Let $\boldsymbol{A}$ satisfy Assumption 1. Then, the optimal value θ of problem $\mathsf{P}(\boldsymbol{A}, r)$ satisfies $\theta \leq 2\epsilon$.*

Proof. Since $\boldsymbol{A}$ satisfies Assumption 1, it is given by $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N}$ where $\boldsymbol{V} \in \mathbb{R}_+^{d \times n}$ is r-separable of the form $\boldsymbol{V} = \boldsymbol{WH} = \boldsymbol{W}[\boldsymbol{I}, \bar{\boldsymbol{H}}]\boldsymbol{\Pi}$ shown in (1) and $\boldsymbol{N} \in \mathbb{R}^{d \times n}$ is noise. Using the permutation matrix $\boldsymbol{\Pi}$ and the nonnegative matrix $\bar{\boldsymbol{H}}$, we construct the matrix $\boldsymbol{X}_0$ that is shown in (5), i.e.,

$$\boldsymbol{X}_0 = \boldsymbol{\Pi}^{-1} \left[\begin{array}{c|c} \boldsymbol{I} & \bar{\boldsymbol{H}} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array} \right] \boldsymbol{\Pi} \in \mathbb{R}^{n \times n}.$$

Since Assumption 1(a) holds, we can check that $\boldsymbol{X}_0$ is a feasible solution of $\mathsf{P}(\boldsymbol{A}, r)$. Hence, the objective function value at $\boldsymbol{X}_0$ satisfies $\theta \leq \|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{X}_0\|_1$. In addition, as shown in (6), we have $\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{X}_0\|_1 \leq 2\epsilon$. Consequently, $\theta \leq 2\epsilon$ holds. □

Let $\boldsymbol{A}$ satisfy Assumption 1. Let I be a set of basis indices of $\boldsymbol{V}$. Gillis showed in Lemma 2.1 of [9] that a feasible solution $\boldsymbol{X}$ of problem Q has the following properties: the L_1 norm of each column of $\boldsymbol{X}$ is less than about 1, and $\boldsymbol{VX}$ serves as a good approximation to $\boldsymbol{V}$. Using the results, Gillis showed in Lemma 2.2 of [9] that the diagonal elements of $\boldsymbol{X}$ indexed by I take higher values than the others. Hence, we can construct I by checking the values of the diagonal elements of $\boldsymbol{X}$. Lemma 1 implies that the optimal solution of problem P is feasible for problem Q. Hence, the same results as in Lemmas 2.1 and 2.2 of [9] hold for the optimal solution of problem P. Here, we formally describe these results as Lemmas 2 and 3.

Lemma 2. *Let $\boldsymbol{A}$ satisfy Assumption 1. Then, the optimal solution $\boldsymbol{X}_{\mathrm{opt}} \in \mathbb{R}^{n \times n}$ of problem $\mathsf{P}(\boldsymbol{A}, r)$ satisfies*

$$\|\boldsymbol{X}_{\mathrm{opt}}(:, i)\|_1 \leq 1 + \frac{4\epsilon}{1 - \epsilon} \quad and \quad \|\boldsymbol{v}_i - \boldsymbol{V}\boldsymbol{X}_{\mathrm{opt}}(:, i)\|_1 \leq \frac{4\epsilon}{1 - \epsilon}$$

for $i \in N$.

The proof is almost the same as the one of Lemma 2.1 in [9]. We have included it in Appendix A to make the discussion self-contained.

Lemma 3. *Let $\boldsymbol{A}$ satisfy Assumption 1. Assume $\kappa > 0$ and $\beta < 1$. Let I be a set of basis indices of $\boldsymbol{V}$. Let $\boldsymbol{p} = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$ for the optimal solution $\boldsymbol{X}_{\mathrm{opt}}$ of problem $\mathsf{P}(\boldsymbol{A}, r)$. Then, the elements of $\boldsymbol{p}$ indexed by I satisfy*

$$\boldsymbol{p}(i) \geq 1 - \frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)}$$

for every $i \in I$.

We have included the proof in Appendix B. Our proof follows the one of Lemma 2.2 in [9], although additional considerations are made; see Remark 2. The key idea of the proof is as follows. Since $\boldsymbol{A}$ satisfies Assumption 1, it can be written as $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N}$ where $\boldsymbol{V}$ is r-separable of the form $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H} = \boldsymbol{W}[\boldsymbol{I}, \bar{\boldsymbol{H}}]\boldsymbol{\Pi}$ shown in (1) and there is a map $\phi : R \rightarrow N$ such that $\boldsymbol{w}_j = \boldsymbol{v}_{\phi(j)}$ for each $j \in R$. For $j \in R$ and $i = \phi(j) \in N$, let

$$\eta = \boldsymbol{H}(j, :)\boldsymbol{X}_{\mathrm{opt}}(:, i).$$

We can see that η is rewritten by using $\boldsymbol{X}_{\mathrm{opt}}(i, i)$, which is equivalent to $\boldsymbol{p}(i)$, due to $\boldsymbol{H}(j, i) = 1$, and evaluate the lower and upper bounds on η. The result of the lemma follows from the bounds.

Now, we can prove Theorem 1. It follows from Lemma 3.

(Proof of Theorem 1). Let us consider the case of $\beta = 1$. Here, we only have to show that, if $\boldsymbol{A}$ is separable with an overlap of basis columns, then the algorithm finds a set of basis indices. Separability means that duplicate basis columns appear in the columns of $\boldsymbol{A}$. Hence, after conducting step 1, the resulting matrix is separable with no overlapping basis columns. This reduces to the case of $\beta < 1$.

Let us move on to the case of $\beta < 1$. Step 2 solves problem $\mathsf{P}(\boldsymbol{A}, r)$ and sets $\boldsymbol{p} = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$ for the optimal solution $\boldsymbol{X}_{\mathrm{opt}}$. Let I be a set of basis indices of $\boldsymbol{V}$. Lemma 3 tells us that

$$\boldsymbol{p}(i) \geq 1 - \underbrace{\frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)}}_{(A)}$$

holds for every $i \in I$. Since

$$\epsilon \leq \frac{\kappa(1-\beta)}{9(r+1)} \leq \frac{1}{18},$$

we have $1 - \epsilon \geq 17/18 > 8/9$. In light of this, the term (A) is bounded as follows:

$$(A) < \frac{9\epsilon}{\kappa(1-\beta)} \leq \frac{1}{r+1}.$$

We thus obtain

$$\boldsymbol{p}(i) > \frac{r}{r+1} \quad \text{for } i \in I. \tag{7}$$

10

The first constraint of problem $\mathsf{P}(\boldsymbol{A}, r)$ requires $\boldsymbol{X}_{\mathrm{opt}}$ to satisfy $\mathrm{tr}(\boldsymbol{X}_{\mathrm{opt}}) = r \Leftrightarrow \sum_{i \in N} \boldsymbol{p}(i) = r$. Hence,

$$r = \sum_{i \in N} \boldsymbol{p}(i) = \sum_{i \in I} \boldsymbol{p}(i) + \sum_{i \in N \setminus I} \boldsymbol{p}(i).$$

Combining it with inequality (7) gives

$$\boldsymbol{p}(i) \leq \frac{r}{r+1} \quad \text{for } i \in N \setminus I. \tag{8}$$

Since I has r elements, inequalities (7) and (8) ensure that the index set corresponding to the r largest elements of $\boldsymbol{p}$ coincides with I. Hence, the index set J constructed in step 3 coincides with I, which is the set of basis indices of $\boldsymbol{V}$. Consequently, after suitably rearranging the columns of $\boldsymbol{W}$, the output $\boldsymbol{W}_{\mathrm{out}} = \boldsymbol{A}(:, J)$ satisfies $\|\boldsymbol{W} - \boldsymbol{W}_{\mathrm{out}}\|_1 \leq \epsilon$. $\qquad\blacksquare$

5   Refinement of Hottopixx with Postprocessing

5.1   Algorithm

We explore the case where there are duplicate basis columns in the input matrix of Algorithm 1. As shown in Section 4.2, the algorithm's guarantee of robustness to noise is founded upon Lemma 3. However, the lemma does not hold any more, because $\beta = 1$ in this case. To address this issue, we develop and incorporate postprocessing in the algorithm.

Let us outline our postprocessing first and give the details at the end of this section. In what follows, we will assume that we are given $\boldsymbol{A}$ satisfying Assumption 1. We use the term *cluster* to refer to the set of column indices of $\boldsymbol{A}$. Although Lemma 3 does not hold in the case where there are duplicate basis columns, the optimal solution of problem P still provides us with clues to finding clusters from which we can obtain near-basis columns. For a cluster $S \subset N$ and $\boldsymbol{p} \in \mathbb{R}_+^n$, define the *score* of cluster S by

$$\mathrm{score}(S, \boldsymbol{p}) = \sum_{u \in S} \boldsymbol{p}(u),$$

and we call $\boldsymbol{p}$ a *point list*.

Let $\mu > 0$ be a parameter and define

$$T_j = \{u \in N : \|\boldsymbol{a}_u - \boldsymbol{w}_j\|_1 \leq 2\mu\} \tag{9}$$

for each $j \in R$. Here, $\boldsymbol{a}_u$ is the uth column of $\boldsymbol{A}$ and $\boldsymbol{w}_j$ is the jth column of $\boldsymbol{W}$. We call $T_1, \ldots, T_r$ *anchors* with parameter μ. Let $\boldsymbol{p} = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$ for the optimal solution $\boldsymbol{X}_{\mathrm{opt}}$ of problem P, and choose the parameter μ of T_j, depending on the noise level ϵ involved in $\boldsymbol{A}$. We show in Corollary 1 that the anchors $T_1, \ldots, T_r$ have high scores, i.e.,

$$\mathrm{score}(T_j, \boldsymbol{p}) > \frac{r}{r+1}$$

for every $j \in R$. Lemma 3.3 of [9] by Gillis implies that the same result holds for a feasible solution of problem Q.

If we find all the anchors, then near-basis columns can be obtained by choosing one element from each anchor. However, even if we use the point list obtained from the optimal solution of P, it is not an easy task to find anchors exactly. We thus construct a collection $\mathcal{F}$ of clusters that contains all the anchors, and observe the structure of $\mathcal{F}$. Our postprocessing algorithm is designed on the basis of this observation. To describe $\mathcal{F}$, we introduce Ω, which is a collection of clusters, that will serve as the foundation of $\mathcal{F}$. Sort columns $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ of $\boldsymbol{A}$ by their L_1 distance to $\boldsymbol{a}_i$ in ascending order so that

$$\|\boldsymbol{a}_i - \boldsymbol{a}_{u_1}\|_1 \leq \|\boldsymbol{a}_i - \boldsymbol{a}_{u_2}\|_1 \leq \cdots \leq \|\boldsymbol{a}_i - \boldsymbol{a}_{u_{n-1}}\|_1$$

where $\{i, u_1, \ldots, u_{n-1}\} = N$. Then, construct

$$\Omega_i = \{\{i\}, \{i, u_1\}, \{i, u_1, u_2\}, \ldots, \{i, u_1, u_2, \ldots, u_{n-1}\}\}$$

and let

$$\Omega = \bigcup_{i \in N} \Omega_i.$$

For a cluster $S \in \Omega_i$, define the diameter of S in Ω_i by

$$\mathrm{diam}(S) = \max_{u \in S} \|\boldsymbol{a}_i - \boldsymbol{a}_u\|_1.$$

For a point list $\boldsymbol{p} \in \mathbb{R}_+^n$ and parameter μ used for constructing the anchors $T_1, \ldots, T_r$, let

$$\mathcal{F}_i(\boldsymbol{p}) = \left\{ S \in \Omega_i : \mathrm{diam}(S) \leq 3\mu, \ \mathrm{score}(S, \boldsymbol{p}) > \frac{r}{r+1} \right\} \tag{10}$$

and

$$\mathcal{F}(\boldsymbol{p}) = \bigcup_{i \in N} \mathcal{F}_i(\boldsymbol{p}).$$

In particular, if $\boldsymbol{p}$ is set as $\boldsymbol{p} = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$ for the optimal solution $\boldsymbol{X}_{\mathrm{opt}}$ of problem P, we use the abbreviation $\mathcal{F}_i$ for $\mathcal{F}_i(\boldsymbol{p})$ and the abbreviation $\mathcal{F}$ for $\mathcal{F}(\boldsymbol{p})$. That is, for $\boldsymbol{p} = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$,

$$\mathcal{F}_i = \mathcal{F}_i(\boldsymbol{p}) \quad \text{and} \quad \mathcal{F} = \mathcal{F}(\boldsymbol{p}).$$

As mentioned above, we have to choose μ depending on the noise level ϵ to ensure that the anchors can have high scores. Hence, it is impossible to construct $\mathcal{F}(\boldsymbol{p})$. But, it is possible to compute some of the clusters in $\mathcal{F}(\boldsymbol{p})$. Consider a collection $\mathcal{G}_i(\boldsymbol{p})$ of clusters obtained by removing the condition $\mathrm{diam}(S) \leq 3\mu$ in $\mathcal{F}_i(\boldsymbol{p})$:

$$\mathcal{G}_i(\boldsymbol{p}) = \left\{ S \in \Omega_i : \mathrm{score}(S, \boldsymbol{p}) > \frac{r}{r+1} \right\}. \tag{11}$$

Let

$$\mathcal{G}(\boldsymbol{p}) = \bigcup_{i \in N} \mathcal{G}_i(\boldsymbol{p}).$$

Figure 1: Illustration of $\mathcal{F} = \mathcal{F}(\boldsymbol{p})$ and $\mathcal{F}(\boldsymbol{q})$ where $\boldsymbol{p}$ is a point list obtained from the optimal solution of problem P, and $\boldsymbol{q}$ is a point list obtained by updating $\boldsymbol{p}$ such that $\boldsymbol{q}(u) = 0$ if u belongs to the red-colored cluster S; otherwise, $\boldsymbol{q}(u) = \boldsymbol{p}(u)$: clusters (set of points surrounded by an oval), anchors (set of points surrounded by an oval filled with gray color), the components $\bar{\mathcal{F}}_i$ of $\mathcal{F}$ (collection of clusters surrounded by a dotted oval), and basis columns (star).

Unlike $\mathcal{F}(\boldsymbol{p})$, we can construct $\mathcal{G}(\boldsymbol{p})$. Let $\hat{S} = \arg\min_{S \in \mathcal{G}(\boldsymbol{p})} \operatorname{diam}(S)$. Since $\mathcal{F}(\boldsymbol{p}) \subset \mathcal{G}(\boldsymbol{p})$, we have

$$\operatorname{diam}(\hat{S}) = \min_{S \in \mathcal{G}(\boldsymbol{p})} \operatorname{diam}(S) \leq \min_{S \in \mathcal{F}(\boldsymbol{p})} \operatorname{diam}(S) \leq 3\mu.$$

Hence, $\hat{S}$ belongs to $\mathcal{F}(\boldsymbol{p})$. We can get it through $\mathcal{G}(\boldsymbol{p})$.

Let us look at $\mathcal{F}$, which is an abbreviation of $\mathcal{F}(\boldsymbol{p})$ with the point list $\boldsymbol{p}$ obtained from the optimal solution of problem P. We show in Lemma 7 that any cluster in $\mathcal{F}$ always has a common element with some anchor. This means that clusters in $\mathcal{F}$ are localized around each anchor, and anchors are the cores of $\mathcal{F}$. Hence, using the components $\bar{\mathcal{F}}_1, \ldots, \bar{\mathcal{F}}_r$ of $\mathcal{F}$, given as

$$\bar{\mathcal{F}}_j = \{S \in \mathcal{F} : \max_{u \in S} \|\boldsymbol{a}_u - \boldsymbol{w}_j\|_1 \leq 8\mu\}, \tag{12}$$

we can write $\mathcal{F}$ as

$$\mathcal{F} = \bar{\mathcal{F}}_1 \cup \cdots \cup \bar{\mathcal{F}}_r.$$

If anchors are far from each other; in other words, ω is large, then the components $\bar{\mathcal{F}}_1, \ldots, \bar{\mathcal{F}}_r$ are disjoint from each other. The left of Figure 1 illustrates $\mathcal{F}$.

According to the observations made so far, it turns out that we can find a cluster belonging to one of $\bar{\mathcal{F}}_1, \ldots, \bar{\mathcal{F}}_r$. A cluster of $\mathcal{F}$ is obtained by using $\mathcal{G}(\boldsymbol{p})$, and it belongs to one of $\bar{\mathcal{F}}_1, \ldots, \bar{\mathcal{F}}_r$ because $\mathcal{F}$ can be written as $\mathcal{F} = \bar{\mathcal{F}}_1 \cup \cdots \cup \bar{\mathcal{F}}_r$. Let us denote the obtained cluster by S_1, and assume that S_1 belongs to $\bar{\mathcal{F}}_1$ in order to simplify the subsequent description. By updating the point list $\boldsymbol{p}$, we can find a cluster belonging to one of the remaining components $\bar{\mathcal{F}}_2, \ldots, \bar{\mathcal{F}}_r$. Let $\boldsymbol{q}$ be a point list made by updating $\boldsymbol{p}$ as

$$\boldsymbol{q}(u) = \begin{cases} 0 & \text{if } u \in S_1, \\ \boldsymbol{p}(u) & \text{otherwise.} \end{cases}$$

13

Algorithm 2 Refinement of Hottopixx with postprocessing

Input: $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ and a positive integer r.
Output: $\boldsymbol{W}_{\mathrm{out}} \in \mathbb{R}^{d \times r}$.

1. Compute the optimal solution $\boldsymbol{X}_{\mathrm{opt}} \in \mathbb{R}^{n \times n}$ of problem $\mathsf{P}(\boldsymbol{A}, r)$.

2. Set $\boldsymbol{p}_1 = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$, $J = \emptyset$ and $\ell = 1$. Perform the following procedure.

   2-1. Find S_ℓ such that

   $$S_\ell = \arg \min_{S \in \mathcal{G}(\boldsymbol{p}_\ell)} \mathrm{diam}(S).$$

   2-2. Choose one element from S_ℓ and add it to J. Increase ℓ by 1.

   2-3. If $\ell = r$, then return $\boldsymbol{W}_{\mathrm{out}} = \boldsymbol{A}(:, J)$ and terminate; otherwise, construct $\boldsymbol{p}_\ell \in \mathbb{R}_+^n$ as

   $$\boldsymbol{p}_\ell(u) = \begin{cases} 0 & \text{if } u \in S_1 \cup \cdots \cup S_{\ell-1}, \\ \boldsymbol{p}_1(u) & \text{otherwise}, \end{cases}$$

   and go to step 2-1.

We show in Lemma 10 that $\mathcal{F}(\boldsymbol{q})$ can be written as

$$\mathcal{F}(\boldsymbol{q}) = \bar{\mathcal{F}}_2 \cup \cdots \cup \bar{\mathcal{F}}_r.$$

The right of Figure 1 illustrates $\mathcal{F}(\boldsymbol{q})$. A cluster, denoted by S_2, of $\mathcal{F}(\boldsymbol{q})$ is obtained by using $\mathcal{G}(\boldsymbol{q})$, and it belongs to one of $\bar{\mathcal{F}}_2, \ldots, \bar{\mathcal{F}}_r$. By repeating the procedure, we can find r clusters $S_1, \ldots, S_r$ such that $S_j \in \bar{\mathcal{F}}_j$ for each $j \in R$ by rearranging the indices of $\bar{\mathcal{F}}_1, \ldots, \bar{\mathcal{F}}_r$. The obtained clusters provide near-basis columns. We choose one element from each cluster and construct the set J. Rearranging the columns of $\boldsymbol{W}$, we find that it satisfies

$$\|\boldsymbol{W} - \boldsymbol{A}(:, J)\|_1 \leq 8\mu.$$

This leads to Theorem 2.

Algorithm 2 is a formal description of our algorithm. It takes as input $(\boldsymbol{A}, r)$. Step 2 is the postprocessing. The cost of step 2 is dominated by step 2-1. The cost of step 2-1 is in turn dominated by the computation of the L_1 distance between any two columns of $\boldsymbol{A} \in \mathbb{R}^{d \times n}$, which takes $O(n^2 d)$ flops. We below summarize the definition and role of $T_j, \mathcal{F}(\boldsymbol{p}), \mathcal{G}(\boldsymbol{p})$ and $\bar{\mathcal{F}}_j$, which are used for analyzing Algorithm 2 in Section 5.2.

- T_j is a cluster, called anchor, which is defined as in (9).

- $\mathcal{F}(\boldsymbol{p})$ is a collection of clusters constructed by using a point list $\boldsymbol{p}$. This is formed as $\mathcal{F}(\boldsymbol{p}) = \cup_{i \in N} \mathcal{F}_i(\boldsymbol{p})$ where $\mathcal{F}_i(\boldsymbol{p})$ is defined as in (10). If $\boldsymbol{p}$ is a point list obtained from the optimal solution of problem P, we abbreviate $\mathcal{F}(\boldsymbol{p})$ and $\mathcal{F}_i(\boldsymbol{p})$ as $\mathcal{F}$ and $\mathcal{F}_i$, respectively.

- $\mathcal{G}(\boldsymbol{p})$ is a collection of clusters constructed by using a point list $\boldsymbol{p}$. This is formed as $\mathcal{G}(\boldsymbol{p}) = \cup_{i \in N} \mathcal{G}_i(\boldsymbol{p})$ where $\mathcal{G}_i(\boldsymbol{p})$ is defined as in (11), which is obtained by discarding some condition imposed on $\mathcal{F}_i(\boldsymbol{p})$.

- $\bar{\mathcal{F}}_j$ is the component of $\mathcal{F}$, defined as in (12). We show in Lemma 8 that $\mathcal{F}$ is written as $\mathcal{F} = \cup_{j \in R} \bar{\mathcal{F}}_j$.

5.2 Analysis

5.2.1 Scores of Anchors

We show that anchors have high scores by using the point list obtained by solving problem P.

Lemma 4. *Let $\boldsymbol{A}$ satisfy Assumption 1. Assume $\kappa > 0$. Let μ satisfy $\mu \neq 0$ and $\epsilon \leq \mu$. Set $\boldsymbol{p} = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$ for the optimal solution $\boldsymbol{X}_{\mathrm{opt}}$ of problem $\mathsf{P}(\boldsymbol{A}, r)$. Then, anchors $T_1, \ldots, T_r$ with parameter μ satisfy*

$$\mathrm{score}(T_j, \boldsymbol{p}) \geq 1 - \frac{16\epsilon}{\kappa\mu(1 - \epsilon)}.$$

for every $j \in R$.

We can prove this in a similar way as Lemma 3; the proof is in Appendix B. From Lemma 4, we immediately obtain Corollary 1.

Corollary 1. *Let $\boldsymbol{A}$ satisfy Assumption 1. Set $\boldsymbol{p} = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$ for the optimal solution $\boldsymbol{X}_{\mathrm{opt}}$ of problem $\mathsf{P}(\boldsymbol{A}, r)$. Consider two cases as follows:*

- *Let ϵ satisfy $\epsilon < \frac{\kappa\omega}{578(r+1)}$. The value of μ is set as $\mu = \frac{17(r+1)\epsilon}{\kappa} + \xi$ by choosing an arbitrary real number ξ from the open interval $(0, \frac{\kappa}{35})$.*

- *Let ϵ satisfy $\epsilon < \frac{\kappa^2}{289(r+1)^2}$. The value of μ is set as $\mu = \sqrt{\epsilon} + \xi$ by choosing an arbitrary real number ξ from the open interval $(0, \frac{\kappa}{35})$.*

The following hold in both cases.

(a) $0 \leq \epsilon < 1$.

(b) $0 < \mu < \frac{\omega}{17}$.

(c) $\epsilon \leq \mu$.

(d) $\mathrm{score}(T_j, \boldsymbol{p}) > \frac{r}{r+1}$ *for every $j \in R$.*

We can easily check that the corollary holds. The proof is given in Appendix C. Part (a) just tells us that the bounds imposed on ϵ in the two cases do not violate Assumption 1(b). The role of ξ is to prevent the value of μ from being zero; hence, we are allowed to choose an arbitrary real number from the open interval $(0, \frac{\kappa}{35})$.

5.2.2 Structure of $\mathcal{F}$

We prove the observations about $\mathcal{F}$ that we made in Section 5.1.

Lemma 5. *Let $\mathcal{F}(\boldsymbol{q}) \neq \emptyset$ for some $\boldsymbol{q} \in \mathbb{R}_+^n$. Then, $\mathcal{G}(\boldsymbol{q}) \neq \emptyset$. Moreover, $\mathcal{F}(\boldsymbol{q})$ contains $\hat{S} = \arg\min_{S \in \mathcal{G}(\boldsymbol{q})} \mathrm{diam}(S)$.*

Proof. Since $\mathcal{F}_i(\boldsymbol{q}) \subset \mathcal{G}_i(\boldsymbol{q}) \subset \mathcal{G}(\boldsymbol{q})$ and $\mathcal{F}(\boldsymbol{q}) = \bigcup_{i \in N} \mathcal{F}_i(\boldsymbol{q})$, any element of $\mathcal{F}(\boldsymbol{q})$ belongs to $\mathcal{G}(\boldsymbol{q})$. Hence, $\mathcal{F}(\boldsymbol{q}) \subset \mathcal{G}(\boldsymbol{q})$ holds. Consequently, $\mathcal{F}(\boldsymbol{q}) \neq \emptyset$ implies $\mathcal{G}(\boldsymbol{q}) \neq \emptyset$.

Since $\hat{S}$ belongs to $\mathcal{G}(\boldsymbol{q}) = \bigcup_{i \in N} \mathcal{G}_i(\boldsymbol{q})$, we have $\hat{S} \in \Omega_{i_*}$ for some $i_* \in N$ and $\text{score}(\hat{S}, \boldsymbol{q}) > \frac{r}{r+1}$. From the relation $\mathcal{F}(\boldsymbol{q}) \subset \mathcal{G}(\boldsymbol{q})$, we have

$$\text{diam}(\hat{S}) = \min_{S \in \mathcal{G}(\boldsymbol{q})} \text{diam}(S) \leq \min_{S \in \mathcal{F}(\boldsymbol{q})} \text{diam}(S) \leq 3\mu.$$

Consequently, $\hat{S} \in \mathcal{F}_{i_*}(\boldsymbol{q})$, which implies $\hat{S} \in \mathcal{F}(\boldsymbol{q})$. □

Lemma 6. *Frame the hypotheses of Corollary 1. The following hold:*

(a) *Anchor T_j is not empty.*

(b) *All anchors $T_1, \ldots, T_r$ belong to $\mathcal{F}$.*

(c) *Anchor T_j belongs to the component $\bar{\mathcal{F}}_j$ of $\mathcal{F}$.*

Proof. Separability means that there is a map $\phi : R \to N$ such that $\boldsymbol{w}_j = \boldsymbol{v}_{\phi(j)}$ for each $j \in R$. We use the map ϕ in the proof of parts (a) and (b).

(a) From Corollary 1(c), we have

$$\|\boldsymbol{a}_{\phi(j)} - \boldsymbol{w}_j\|_1 = \|\boldsymbol{v}_{\phi(j)} + \boldsymbol{n}_{\phi(j)} - \boldsymbol{w}_j\|_1 = \|\boldsymbol{n}_{\phi(j)}\|_1 \leq \epsilon \leq \mu.$$

Hence, T_j contains $\phi(j)$, which means that T_j is not empty.

(b) We show that T_j belongs to $\mathcal{F}_{\phi(j)}$ for each $j \in R$. Since $\phi(j) \in T_j$, as shown in part (a), we have $T_j \in \Omega_{\phi(j)}$. By Corollary 1(d), the score of T_j by $\boldsymbol{p}$ satisfies $\text{score}(T_j, \boldsymbol{p}) > \frac{r}{r+1}$. The diameter of T_j in $\Omega_{\phi(j)}$ satisfies $\text{diam}(T_j) \leq 3\mu$, since any $u \in T_j$ satisfies

$$\|\boldsymbol{a}_u - \boldsymbol{a}_{\phi(j)}\|_1 = \|\boldsymbol{a}_u - \boldsymbol{v}_{\phi(j)} - \boldsymbol{n}_{\phi(j)}\|_1 = \|\boldsymbol{a}_u - \boldsymbol{w}_j - \boldsymbol{n}_{\phi(j)}\|_1 \leq \|\boldsymbol{a}_u - \boldsymbol{w}_j\|_1 + \|\boldsymbol{n}_{\phi(j)}\|_1 \leq 2\mu + \epsilon$$
$$\leq 3\mu.$$

The last inequality uses Corollary 1(c). Hence, $T_j \in \mathcal{F}_{\phi(j)}$ for each $j \in R$. In addition, the definition of $\mathcal{F}$ implies $\mathcal{F}_{\phi(j)} \subset \mathcal{F}$. Consequently, $T_1, \ldots, T_r$ belong to $\mathcal{F}$.

(c) We have already shown $T_j \in \mathcal{F}$ for each $j \in R$ in part (b). The definition of T_j implies that, for any $u \in T_j$, we have $\|\boldsymbol{a}_u - \boldsymbol{w}_j\|_1 \leq 2\mu \leq 8\mu$. Hence, T_j belongs to $\bar{\mathcal{F}}_j$. □

Parts (a) and (c) tell us that the components $\bar{\mathcal{F}}_1, \ldots, \bar{\mathcal{F}}_r$ of $\mathcal{F}$ are not empty. We will use this observation in the proof of Theorem 2.

Lemma 7. *Frame the hypotheses of Corollary 1. For any $S \in \mathcal{F}$, there is some $j \in R$ such that $S \cap T_j \neq \emptyset$.*

Proof. We start by showing that any two different anchors do not have a common element. Let $x, y \in R$ and $x \neq y$. No $u \in T_x$ belongs to T_y, since

$$\begin{aligned}
\|\boldsymbol{a}_u - \boldsymbol{w}_y\|_1 &= \|(\boldsymbol{w}_x - \boldsymbol{w}_y) + (\boldsymbol{a}_u - \boldsymbol{w}_x)\|_1 \\
&\geq \|\boldsymbol{w}_x - \boldsymbol{w}_y\|_1 - \|\boldsymbol{a}_u - \boldsymbol{w}_x\|_1 \\
&\geq \omega - 2\mu && \text{(by the definition of } \omega) \\
&> 15\mu && \text{(by Corollary 1(b)).}
\end{aligned}$$

16

Hence, $T_x \cap T_y = \emptyset$ holds for any different x and y in R. We will prove the lemma by contradiction. Assume that there is some $S \in \mathcal{F}$ such that $S \cap T_j = \emptyset$ for any $j \in R$. Since $T_x \cap T_y = \emptyset$ for $x, y \in R$ with $x \neq y$, any two different clusters among $S, T_1, \ldots, T_r$ do not have a common element. Hence, we have

$$\operatorname{score}(S, \boldsymbol{p}) + \sum_{j \in R} \operatorname{score}(T_j, \boldsymbol{p}) = \operatorname{score}(S \cup T_1 \cup \cdots \cup T_r, \boldsymbol{p}) \leq \operatorname{score}(N, \boldsymbol{p}) = r.$$

The last equality follows from the fact that $\operatorname{score}(N, \boldsymbol{p}) = \operatorname{tr}(\boldsymbol{X}_{\mathrm{opt}}) = r$ holds since the first constraint of problem P requires $\boldsymbol{X}_{\mathrm{opt}}$ to satisfy $\operatorname{tr}(\boldsymbol{X}_{\mathrm{opt}}) = r$. By Corollary 1(d), the score of T_j by $\boldsymbol{p}$ satisfies $\operatorname{score}(T_j, \boldsymbol{p}) > \frac{r}{r+1}$ for each $j \in R$. Therefore, we get $\operatorname{score}(S, \boldsymbol{p}) \leq \frac{r}{r+1}$ and reach a contradiction to $S \in \mathcal{F}$, which means $\operatorname{score}(S, \boldsymbol{p}) > \frac{r}{r+1}$. The assumption is false. That is, for any $S \in \mathcal{F}$, there is some $j \in R$ such that $S \cap T_j \neq \emptyset$. □

Lemma 8. *Frame the hypotheses of Corollary 1. The following hold:*

(a) *$\mathcal{F}$, i.e., the abbreviation of $\mathcal{F}(\boldsymbol{p})$, is represented as*

$$\mathcal{F} = \bigcup_{j \in R} \bar{\mathcal{F}}_j$$

by using the components $\bar{\mathcal{F}}_j$ of $\mathcal{F}$.

(b) *Let $x, y \in R$ and $x \neq y$. We have $S_x \cap S_y = \emptyset$ for any $(S_x, S_y) \in \bar{\mathcal{F}}_x \times \bar{\mathcal{F}}_y$.*

Proof. (a) First, we prove the inclusion "$\supset$". Let $S \in \cup_{j \in R} \bar{\mathcal{F}}_j$. Then, there is a $j_* \in R$ such that $S \in \bar{\mathcal{F}}_{j_*}$. The definition of $\bar{\mathcal{F}}_{j_*}$ implies $S \in \mathcal{F}$. Hence, the inclusion "$\supset$" holds.

Next, we prove the inclusion "$\subset$". Let $S \in \mathcal{F}$. Recall that $\mathcal{F}$ is defined by $\mathcal{F} = \cup_{i \in N} \mathcal{F}_i$. Hence, there is an $i_* \in N$ such that $S \in \mathcal{F}_{i_*}$. Lemma 7 ensures that there is a $j_* \in R$ such that $S \cap T_{j_*} \neq \emptyset$. Let $v \in S \cap T_{j_*}$. Then, for any $u \in S$,

$$
\begin{aligned}
\|\boldsymbol{a}_u - \boldsymbol{w}_{j_*}\|_1 = \|(\boldsymbol{a}_u - \boldsymbol{a}_v) + (\boldsymbol{a}_v - \boldsymbol{w}_{j_*})\|_1 &\leq \|\boldsymbol{a}_u - \boldsymbol{a}_v\|_1 + \|\boldsymbol{a}_v - \boldsymbol{w}_{j_*}\|_1 \\
&\leq \|\boldsymbol{a}_u - \boldsymbol{a}_v\|_1 + 2\mu && \text{(by } v \in T_{j_*}) \\
&= \|(\boldsymbol{a}_u - \boldsymbol{a}_{i_*}) + (\boldsymbol{a}_{i_*} - \boldsymbol{a}_v)\|_1 + 2\mu \\
&\leq \|\boldsymbol{a}_u - \boldsymbol{a}_{i_*}\|_1 + \|\boldsymbol{a}_{i_*} - \boldsymbol{a}_v\|_1 + 2\mu \\
&\leq 8\mu. && \text{(by } u, v \in S \text{ and } S \in \mathcal{F}_{i_*})
\end{aligned}
$$

Accordingly, we have $S \in \bar{\mathcal{F}}_{j_*}$ for $j_* \in R$, which implies $S \in \cup_{j \in R} \bar{\mathcal{F}}_j$. Hence, the inclusion "$\subset$" holds. Consequently, $\mathcal{F} = \bigcup_{j \in R} \bar{\mathcal{F}}_j$ as claimed.

(b) Let $x, y \in R$ and $x \neq y$. Let $S_x \in \bar{\mathcal{F}}_x$ and $S_y \in \bar{\mathcal{F}}_y$. We have, for any $u \in S_x$,

$$
\begin{aligned}
\|\boldsymbol{a}_u - \boldsymbol{w}_y\|_1 = \|(\boldsymbol{w}_x - \boldsymbol{w}_y) + (\boldsymbol{a}_u - \boldsymbol{w}_x)\|_1 &\\
&\geq \|\boldsymbol{w}_x - \boldsymbol{w}_y\|_1 - \|\boldsymbol{a}_u - \boldsymbol{w}_x\|_1 \\
&\geq \omega - 8\mu && \text{(by the definition of } \omega \text{ and } S_x \in \bar{\mathcal{F}}_x) \\
&> 9\mu && \text{(by Corollary 1(b)).}
\end{aligned}
$$

Hence, $u \notin S_y$. This means $S_x \cap S_y = \emptyset$. □

Here, we prove Lemma 9 for establishing Lemma 10. In Lemmas 9 and 10, we use the following notation: $\ell_1, \ldots, \ell_r$ denote the r integers in R; k is any positive integer satisfying $k < r$; and K is the set of consecutive integers from 1 to k.

Lemma 9. *Frame the hypotheses of Corollary 1. We have the relation*

$$\bigcup_{j \in K} \bar{\mathcal{F}}_{\ell_j} = \mathcal{F} \setminus \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j}.$$

Proof. Lemma 8(b) implies $\bar{\mathcal{F}}_x \cap \bar{\mathcal{F}}_y = \emptyset$ for $x, y \in R$ with $x \neq y$. We use this relation in the proof. To simplify the description, we denote $\mathcal{A} = \bigcup_{j \in K} \bar{\mathcal{F}}_{\ell_j}$ and $\mathcal{B} = \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j}$.

First, we prove the inclusion "$\subset$". Let $S \in \mathcal{A}$. Then, $S \in \bar{\mathcal{F}}_{\ell_{j_*}}$ for some $j_* \in K$. This implies $S \in \mathcal{F}$ by the definition of $\bar{\mathcal{F}}_{\ell_{j_*}}$. In addition, as shown above, we have $\bar{\mathcal{F}}_{\ell_{j_*}} \cap \bar{\mathcal{F}}_{\ell_j} = \emptyset$ for every $j \in R \setminus K$. Hence, $S \notin \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j}$. Consequently, the inclusion "$\subset$" holds.

Next, we prove the inclusion "$\supset$". It holds if $\mathcal{F} \setminus \mathcal{B} = \emptyset$. In what follows, we thus assume $\mathcal{F} \setminus \mathcal{B} \neq \emptyset$. We use contradiction. Since the assumption means that there exists $S \in \mathcal{F} \setminus \mathcal{B}$, we choose such S. Let us assume contradiction; $S \notin \mathcal{A}$. Then, $S \in \mathcal{F} \cap \mathcal{A}^c \cap \mathcal{B}^c$. Meanwhile,

$$\mathcal{F} \cap \mathcal{A}^c \cap \mathcal{B}^c = \mathcal{F} \cap (\mathcal{A} \cup \mathcal{B})^c = \mathcal{F} \cap \left(\bigcup_{j \in R} \bar{\mathcal{F}}_{\ell_j} \right)^c = \mathcal{F} \cap \mathcal{F}^c = \emptyset$$

holds by De Morgan's laws and Lemma 8(a). This contradicts the fact that S exists. Hence, the inclusion "$\supset$" holds. Consequently, $\bigcup_{j \in K} \bar{\mathcal{F}}_{\ell_j} = \mathcal{F} \setminus \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j}$ as claimed. $\square$

Lemma 10. *Frame the hypotheses of Corollary 1. Let $S_1, \ldots, S_k$ be clusters such that $S_j \in \bar{\mathcal{F}}_{\ell_j}$ for each $j \in K$. Suppose that we are given $S_1, \ldots, S_k$ and the point list $\boldsymbol{p}$. Construct a point list $\boldsymbol{q} \in \mathbb{R}^n_+$:*

$$\boldsymbol{q}(u) = \begin{cases} 0 & \text{if } u \in S_1 \cup \cdots \cup S_k, \\ \boldsymbol{p}(u) & \text{otherwise.} \end{cases}$$

Then, the following hold:

(a) *Let $S \in \mathcal{F}$. Then,*

$$S \in \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j} \Leftrightarrow \operatorname{score}(S, \boldsymbol{q}) > \frac{r}{r+1}.$$

(b) *$\mathcal{F}(\boldsymbol{q})$ is represented as*

$$\mathcal{F}(\boldsymbol{q}) = \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j}.$$

Proof. (a) First, we prove the direction "$\Rightarrow$". Let $S \in \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j}$. Then, S belongs to $\bar{\mathcal{F}}_{\ell_{j_*}}$ for some $j_* \in R \setminus K$. Meanwhile, S_j belongs to $\bar{\mathcal{F}}_{\ell_j}$ for $j \in K$. Lemma 8(b) then tells us that $S \cap S_j = \emptyset$ for every $j \in K$. Hence, from the construction of $\boldsymbol{q}$, we have $\boldsymbol{q}(u) = \boldsymbol{p}(u)$ for every $u \in S$. In addition, since $S \in \bar{\mathcal{F}}_{\ell_{j_*}}$ implies $S \in \mathcal{F}$ by the definition of $\bar{\mathcal{F}}_{\ell_{j_*}}$, it takes $\operatorname{score}(S, \boldsymbol{p}) > \frac{r}{r+1}$. Consequently, we obtain $\operatorname{score}(S, \boldsymbol{q}) = \operatorname{score}(S, \boldsymbol{p}) > \frac{r}{r+1}$.

18

Next, we prove the direction "⇐" by showing that the contrapositive is true. In light of Lemma 9, the contrapositive statement is

$$S \in \bigcup_{j \in K} \bar{\mathcal{F}}_{\ell_j} \Rightarrow \mathrm{score}(S, \boldsymbol{q}) \le \frac{r}{r+1}. \tag{13}$$

Let $S \in \bigcup_{j \in K} \bar{\mathcal{F}}_{\ell_j}$. Then, S belongs to $\bar{\mathcal{F}}_{\ell_{j_*}}$ for some $j_* \in K$. From the construction of $\boldsymbol{q}$, we have $\boldsymbol{q}(u) = 0$ for every $u \in S_{j_*}$. Hence,

$$\mathrm{score}(S, \boldsymbol{q}) = \sum_{u \in S} \boldsymbol{q}(u) = \sum_{u \in \bar{S}} \boldsymbol{q}(u). \tag{14}$$

for $\bar{S} = S \setminus S_{j_*}$. Let $S_{k+1}, \ldots, S_r$ be clusters such that $S_j \in \bar{\mathcal{F}}_{\ell_j}$ for each $j \in R \setminus K$. Since $S_j \in \bar{\mathcal{F}}_{\ell_j}$ for $j \in R$ and $\bar{S} = S \setminus S_{j_*}$ where $S, S_{j_*} \in \bar{\mathcal{F}}_{\ell_{j_*}}$, Lemma 8(b) tells us that the following statements hold:

$$\bar{S} \cap S_j = \emptyset \quad \text{for every } j \in R. \tag{15}$$
$$S_x \cap S_y = \emptyset \quad \text{for every different } x, y \in R. \tag{16}$$

Statement (15) implies that no element of $\bar{S}$ belongs to $S_1 \cup \cdots \cup S_k$. Hence,

$$\sum_{u \in \bar{S}} \boldsymbol{q}(u) = \sum_{u \in \bar{S}} \boldsymbol{p}(u) = \mathrm{score}(\bar{S}, \boldsymbol{p}). \tag{17}$$

It follows from equalities (14) and (17) that the relation $\mathrm{score}(S, \boldsymbol{q}) = \mathrm{score}(\bar{S}, \boldsymbol{p})$ holds. From statements (15) and (16), we have

$$\mathrm{score}(\bar{S}, \boldsymbol{p}) + \sum_{j \in R} \mathrm{score}(S_j, \boldsymbol{p}) = \mathrm{score}(\bar{S} \cup S_1 \cup \cdots \cup S_r, \boldsymbol{p}) \le \mathrm{score}(N, \boldsymbol{p}) = r.$$

Here, $\mathrm{score}(S_j, \boldsymbol{p}) > \frac{r}{r+1}$ since $S_j \in \bar{\mathcal{F}}_{\ell_j}$ implies $S_j \in \mathcal{F}$. Accordingly, the inequality above yields $\mathrm{score}(\bar{S}, \boldsymbol{p}) \le \frac{r}{r+1}$. Combining it with the relation $\mathrm{score}(S, \boldsymbol{q}) = \mathrm{score}(\bar{S}, \boldsymbol{p})$, we obtain $\mathrm{score}(S, \boldsymbol{q}) \le \frac{r}{r+1}$. Consequently, statement (13) holds.

(b) First, we prove the inclusion "⊃". Let $S \in \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j}$. Then, S belongs to $\bar{\mathcal{F}}_{\ell_{j_*}}$ for some $j_* \in R \setminus K$. It thus follows from part (a) that $\mathrm{score}(S, \boldsymbol{q}) > \frac{r}{r+1}$. In addition, $S \in \bar{\mathcal{F}}_{\ell_{j_*}}$ implies $S \in \mathcal{F}$ by the definition of $\bar{\mathcal{F}}_{\ell_{j_*}}$. From $\mathcal{F} = \cup_{i \in N} \mathcal{F}_i$, we have $S \in \mathcal{F}_{i_*}$ for some $i_* \in N$. Thus, $S \in \Omega_{i_*}$ and $\mathrm{diam}(S) \le 3\mu$ by the definition of $\mathcal{F}_{i_*}$. Consequently, we obtain $S \in \mathcal{F}_{i_*}(\boldsymbol{q})$, which implies $S \in \mathcal{F}(\boldsymbol{q})$ since $\mathcal{F}(\boldsymbol{q}) = \cup_{i \in N} \mathcal{F}_i(\boldsymbol{q})$.

Next, we prove the inclusion "⊂". Let $S \in \mathcal{F}(\boldsymbol{q})$. Then, S belongs to $\mathcal{F}_{i_*}(\boldsymbol{q})$ for some $i_* \in N$, since $\mathcal{F}(\boldsymbol{q}) = \cup_{i \in N} \mathcal{F}_i(\boldsymbol{q})$. It follows from the definition of $\mathcal{F}_{i_*}(\boldsymbol{q})$ that $S \in \Omega_{i_*}, \mathrm{diam}(S) \le 3\mu$, and $\mathrm{score}(S, \boldsymbol{q}) > \frac{r}{r+1}$. Since S satisfies $\mathrm{score}(S, \boldsymbol{q}) > \frac{r}{r+1}$, part (a) ensures that the inclusion "⊂" holds if $S \in \mathcal{F}$. Thus, the remainder of the proof is to show $S \in \mathcal{F}$. The construction of a point list $\boldsymbol{q}$ tells us that $\boldsymbol{p}(i) \ge \boldsymbol{q}(i)$ for every $i \in N$. Hence,

$$\mathrm{score}(S, \boldsymbol{p}) \ge \mathrm{score}(S, \boldsymbol{q}) > \frac{r}{r+1}.$$

holds. Consequently, we obtain $S \in \mathcal{F}_{i_*}(\boldsymbol{p})$, which implies $S \in \mathcal{F}$ since $\mathcal{F} = \mathcal{F}(\boldsymbol{p}) = \cup_{i \in N} \mathcal{F}_i(\boldsymbol{p})$.

□

5.2.3 Robustness to Noise

We are now ready to prove Theorem 2.

(Proof of Theorem 2). Let $S_1, \ldots, S_r$ be clusters generated by Algorithm 2. We claim that there is a permutation $\pi : R \to R$ such that $S_\ell \in \bar{\mathcal{F}}_{\pi(\ell)}$ for each $\ell \in R$. We use induction on ℓ. Set a parameter μ as $\mu = \lambda + \xi$ by choosing an arbitrary real number ξ from the open interval $(0, \frac{\kappa}{35})$. The value of λ is set according to the noise level described in the theorem:

- $\lambda = \frac{17(r+1)\epsilon}{\kappa}$ in the former case where $\epsilon < \frac{\kappa\omega}{578(r+1)}$.

- $\lambda = \sqrt{\epsilon}$ in the latter case where $\epsilon < \frac{\kappa^2}{289(r+1)^2}$.

Base case: Step 1 of the algorithm computes the optimal solution $\boldsymbol{X}_{\mathrm{opt}}$ of problem $\mathsf{P}(\boldsymbol{A}, r)$ and step 2 sets $\boldsymbol{p}_1 = \mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$. Thus, Lemmas 6 and 8 hold. Lemma 8(a) tells us that $\mathcal{F}(\boldsymbol{p}_1)$ is represented as $\mathcal{F}(\boldsymbol{p}_1) = \bigcup_{j \in R} \bar{\mathcal{F}}_j$. It follows from Lemmas 6(a) and 6(c) that the components $\bar{\mathcal{F}}_1 \ldots, \bar{\mathcal{F}}_r$ are not empty. This means that $\mathcal{F}(\boldsymbol{p}_1)$ is not empty. We can thus use Lemma 5, which tells us that $\mathcal{F}(\boldsymbol{p}_1)$ contains $S_1 = \arg\min_{S \in \mathcal{G}(\boldsymbol{p}_1)} \mathrm{diam}(S)$. Accordingly, there is a $j \in R$ such that $S_1 \in \bar{\mathcal{F}}_j$.

Induction step: Let $\ell_1, \ldots, \ell_r$ denote the r integers in R. Let k be any positive integer satisfying $k < r$, and K be the set of consecutive integers from 1 to k. Suppose that $S_j \in \bar{\mathcal{F}}_{\ell_j}$ holds for each $j \in K$. Lemma 10 holds; part (b) of the lemma tells us that $\mathcal{F}(\boldsymbol{p}_{k+1})$ is represented as $\mathcal{F}(\boldsymbol{p}_{k+1}) = \bigcup_{j \in R \setminus K} \bar{\mathcal{F}}_{\ell_j}$. As mentioned above, $\bar{\mathcal{F}}_{k+1} \ldots, \bar{\mathcal{F}}_r$ are not empty. Hence, $\mathcal{F}(\boldsymbol{p}_{k+1})$ is not empty. We can thus use Lemma 5, which tells us that $\mathcal{F}(\boldsymbol{p}_{k+1})$ contains $S_{k+1} = \arg\min_{S \in \mathcal{G}(\boldsymbol{p}_{k+1})} \mathrm{diam}(S)$. Accordingly, there is a $j \in R \setminus K$ such that $S_{k+1} \in \bar{\mathcal{F}}_{\ell_j}$. Consequently, there is a permutation $\pi : R \to R$ such that $S_\ell \in \bar{\mathcal{F}}_{\pi(\ell)}$ for each $\ell \in R$.

In light of the definition of $\bar{\mathcal{F}}_j$, this result implies that the output $\boldsymbol{W}_{\mathrm{out}} = \boldsymbol{A}(:, J)$ of the algorithm satisfies

$$\|\boldsymbol{W} - \boldsymbol{W}_{\mathrm{out}}\|_1 \le 8\mu = 8(\lambda + \xi)$$

by rearranging the columns of $\boldsymbol{W}$. Since the inequality holds for any small positive number ξ, it turns out that

$$\|\boldsymbol{W} - \boldsymbol{W}_{\mathrm{out}}\|_1 \le 8\lambda$$

holds. This gives the desired results. □

6 Experiments

We conducted experiments to see the practical performance of our algorithms. Gillis and Luce [13] observed in their experiments that the postprocessing of Gillis [9] does not always enhance the robustness of their refinement of Hottopixx. For that reason, they proposed to incorporate a hybrid postprocessing into their refinement. A detailed description was given in Algorithm 6 of [13]. They implemented it and showed its superiority to other algorithms. We incorporated the algorithmic framework of hybrid postprocessing into Algorithm 2, as described in Algorithm 3, and implemented it on MATLAB. The purpose of our experiments was to demonstrate its performance.

Algorithm 3 Refinement of Hottopixx with hybrid postprocessing

---

Input: $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ and a positive integer r.
Output: Set J of r elements from N.

1. Perform step 1 of Algorithm 2. Let J_1 be the index set corresponding to the r largest elements of $\mathrm{diag}(\boldsymbol{X}_{\mathrm{opt}})$.

2. Perform step 2 of Algorithm 2 where step 2-2 chooses

$$u = \arg \max_{u \in S_\ell} \boldsymbol{p}_\ell(u).$$

   Let $J_2 = J$ for the index set J obtained at the termination of step 2.

3. Compute

$$J = \arg \min_{J \in \{J_1, J_2\}} \mathrm{error}(J) \quad \text{where} \quad \mathrm{error}(J) = \min_{\boldsymbol{X} \geq \boldsymbol{0}} \|\boldsymbol{A} - \boldsymbol{A}(:, J)\boldsymbol{X}\|_F^2$$

   and return J.

We compared four algorithms as follows: RHHP (Algorithm 3), LP-rho1 (Algorithm 6 of [13]), Hottopixx (Algorithm 1 of [13]) and SPA (Algorithm 1 with $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2$ of [15]). SPA was originally proposed in [1] in the context of chemometrics, and is now considered a popular algorithm for solving separable NMF problems. For the implementation of LP-rho1, Hottopixx and SPA, we used the MATLAB functions `LPsepNMF_cplex`, `hottopixx_cplex` and `FastSepNMF` whose code is available at the website of the first author of [13]. For solving LP problems, the functions `hottopixx_cplex` and `LPsepNMF_cplex` employed CPLEX. Following them, we employed it in the implementation of RHHP.

We tested the algorithms on four synthetic datasets whose construction is the same as in [13, 10]. Each dataset contained noisy separable matrices $\boldsymbol{A} = \boldsymbol{WH} + \boldsymbol{N} \in \mathbb{R}^{30 \times 200}$ where the factorization rank is 10 and the set of basis indices is $\{1, \ldots, 10\}$. The components $\boldsymbol{W} \in \mathbb{R}_+^{30 \times 10}, \boldsymbol{H} \in \mathbb{R}_+^{10 \times 200}$ and $\boldsymbol{N} \in \mathbb{R}^{30 \times 200}$ were generated as follows.

- $\boldsymbol{W}$: Using the following procedures (A) and (B), two types of matrices were generated.

  (A) Normal: First, generate $\boldsymbol{W} \in \mathbb{R}^{30 \times 10}$ whose elements are drawn from a uniform distribution on the interval $[0, 1]$. Then, normalize the columns to have unit L_1 norm.

  (B) Ill-conditioned: First, generate $\boldsymbol{W} \in \mathbb{R}^{30 \times 10}$ as in the first step of procedure above. Second, compute the reduced SVD $\boldsymbol{W} = \boldsymbol{F\Sigma G}^\top$ where $\boldsymbol{\Sigma}$ is a diagonal matrix of size 10, $\boldsymbol{F} \in \mathbb{R}^{30 \times 10}$, and $\boldsymbol{G} \in \mathbb{R}^{10 \times 10}$. Third, choose a positive integer c and replace $\boldsymbol{W}$ by $\boldsymbol{FSG}^\top$ using a diagonal matrix $\boldsymbol{S}$ of size 10 whose ith diagonal element is $\alpha^{(i-1)}$ for $\alpha \in \mathbb{R}$ satisfying $\alpha^9 = 10^{-c}$. Finally, replace all negative elements by 0 and then normalize the columns to have unit L_1 norm.

- $\boldsymbol{H}$: It is formed as $\boldsymbol{H} = [\boldsymbol{I}, \bar{\boldsymbol{H}}]$ where the submatrix composed of 10 columns from the first one is an identity matrix of size 10, and the columns of the remaining submatrix of size 10×190 are from a Dirichlet distribution whose 10 parameters are uniformly from the interval $[0, 1]$.

Table 3: Average values of $\kappa, \omega, \sigma_{\max}/\sigma_{\min}$ and β over 50 matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ in datasets 1-4.

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Type of $\boldsymbol{W}$	Normal	Ill-conditioned with $c = 3$	Ill-conditioned with $c = 4$	Ill-conditioned with $c = 5$
κ	3.27×10^{-1}	3.67×10^{-2}	1.16×10^{-2}	3.43×10^{-3}
ω	4.70×10^{-1}	1.47×10^{-1}	8.62×10^{-2}	5.15×10^{-2}
$\sigma_{\max}/\sigma_{\min}$	1.09×10^1	3.07×10^2	3.38×10^3	5.36×10^4
β	8.03×10^{-1}	8.03×10^{-1}	8.03×10^{-1}	8.03×10^{-1}

Hence, $\boldsymbol{H}$ is nonnegative and every column has unit L_1 norm. Moreover, if one constructs $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H}$, our parameter choice of a Dirichlet distribution encourages the columns of $\boldsymbol{V}$ to lie around the boundary of the convex hull of the columns of $\boldsymbol{W}$.

- $\boldsymbol{N}$: First, choose a positive real number δ serving as a noise intensity, and generate $\boldsymbol{N} \in \mathbb{R}^{30 \times 200}$ whose elements are from a standard normal distribution. Then, normalize it such that the L_1 norm is equal to δ. Hence, the resulting matrix $\boldsymbol{N}$ satisfies $\|\boldsymbol{N}\|_1 = \delta$.

To generate noisy separable matrices in dataset 1, we chose 20 equally spaced points δ in log space between 10^{-2} and 1, and then constructed $\boldsymbol{N}$ satisfying $\|\boldsymbol{N}\|_1 = \delta$ for each δ. We used the matrices $\boldsymbol{W}$ generated by procedure (A). For those in datasets 2-4, we chose 20 equally spaced points δ in log space between 10^{-2} and 0.5, and then constructed $\boldsymbol{N}$ satisfying $\|\boldsymbol{N}\|_1 = \delta$. We used ill-conditioned matrices $\boldsymbol{W}$ generated by procedure (B) with the choice of c as follows: $c = 3$ for dataset 2, $c = 4$ for dataset 3, and $c = 5$ for dataset 4. In the construction of datasets 1-4, we generated 50 separable matrices $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H}$, and then formed noisy separable matrices $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N}$ by adding 20 matrices $\boldsymbol{N}$ to each $\boldsymbol{V}$; hence, each dataset contained 1,000 matrices in total. Table 3 displays the average values of $\kappa, \omega, \sigma_{\max}/\sigma_{\min}$ and β over 50 matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ in datasets 1-4. Here, $\sigma_{\max}/\sigma_{\min}$ is the ratio of the largest singular value of $\boldsymbol{W}$ divided by the smallest one. Recall that κ and ω are defined in terms of $\boldsymbol{W}$ and β in terms of the submatrix $\bar{\boldsymbol{H}}$ of $\boldsymbol{H}$.

The performance of the algorithm was evaluated by using the index recovery rate, defined by $|J \cup \{1, \ldots, 10\}|/10$ for an index set J output by it. LP-rho1 and Hottopixx required us to designate a noise level ϵ as input. For a matrix $\boldsymbol{A} = \boldsymbol{W}\boldsymbol{H} + \boldsymbol{N}$ in the datasets, we set $\epsilon = \|\boldsymbol{N}\|_1$, which is equal to δ, and then ran the algorithms. The experiments were conducted on Intel Xeon CPU E5-1620 with 64 GB memory running MATLAB.

Figure 2 and Table 4 summarize the experimental results: the figure displays the average of index recovery rates determined by the four algorithms; and the table lists the maximum values of δ for 100% and 80% recovery of basis indices by them. Regarding the index recovery rates of the algorithms, we can see the following:

- RHHP, LP-rho1, and SPA are better than Hottopixx for every dataset.

- For dataset 1, SPA is slightly better than RHHP and LP-rho1, since the maximum value of δ for 80% recovery determined by SPA exceeds those determined by RHHP and LP-rho1. RHHP is almost the same as LP-rho1.

Figure 2: Average of index recovery rates by four algorithms for datasets 1-4.

Table 4: Maximum values of δ for 100% and 80% recovery of basis indices. The symbol "-" in 100% recovery (resp. 80% recovery) means that the average of the index recovery rates at $\delta = 0.01$ is less than 1 (resp. 0.8). The bold-faced values indicate the maximum value in each column.

| | Dataset 1 | | Dataset 2 | | Dataset 3 | | Dataset 4 | |
	100%	80%	100%	80%	100%	80%	100%	80%
RHHP	**0.089**	0.298	**0.015**	**0.118**	-	**0.052**	-	**0.019**
LP-rho1	**0.089**	0.298	0.010	0.096	-	0.042	-	0.015
Hottopixx	-	0.043	-	0.010	-	-	-	-
SPA	**0.089**	**0.379**	-	0.052	-	0.015	-	-

23

- For datasets 2-4, RHHP and LP-rho1 are better than SPA. RHHP is slightly better than LP-rho1, since the maximum values of δ for 80% recovery determined by RHHP exceed those determined by LP-rho1.

The experimental results imply that, without taking a noise level as input, RHHP is as robust to noise as LP-rho1.

7 Concluding Remarks

We refined Hottopixx of Bittorf et al. [5] and the postprocessing of Gillis [9] and showed that our refinement has almost the same robustness to noise as the original one. To enable Hottopixx to run without prior knowledge of the noise level, we replaced the problem Q with P. This is a simple idea, and it is easy to see that Lemma 1 holds. From the lemma, we can immediately see that the refinement is similar in robustness to Hottopixx. However, it is not obvious how the postprocessing of Gillis can be refined so that the algorithm runs without prior knowledge of the noise level. We constructed a collection $\mathcal{F}$ of clusters containing anchors $T_1, \ldots, T_r$ and examined the structure of $\mathcal{F}$. On the basis of this examination, we developed a refinement of the postprocessing and analyzed its robustness to noise.

We close this paper with remarks on directions for future research. There is a computational issue in Algorithms 1 and 2. The bottleneck is in solving problem P. As shown in Section 4.1, this can be transformed into an equivalent LP problem P' with $O(n^2)$ variables and $O(n^2)$ constraints where n is the number of columns of the input matrix and we assume that it is greater than the number d of rows. Since the size of P' grows quadratically with n, solving P' is computationally challenging when n is large. We thus need to develop efficient algorithms. Bittorf et al. [5] and Gillis and Luce [14] used first-order methods and developed algorithms for solving their optimization models Q and R. The use of first-order methods would be promising for solving P' efficiently.

Regarding the bounds given in Theorems 1 and 2, it remains to investigate the tightness of them. Recently, Gillis [11] studied an ideal algorithm for solving separable NMF problems. Since the computational cost grows exponentially with the problem size, it is not realistic to apply the algorithm to large problems. They showed that it achieves the best possible bound on the error relative to the basis. There is a gap between our error bound shown for Algorithm 2 in Theorem 2 and the optimal one. It would be interesting to see whether we can reduce the gap.

Appendix A Proof of Lemma 2

(Proof of Lemma 2). We prove the first inequality. By Lemma 1,

$$
\begin{aligned}
2\epsilon \geq \theta = \|\boldsymbol{AX}_{\mathrm{opt}} - \boldsymbol{A}\|_1 &\geq \|\boldsymbol{AX}_{\mathrm{opt}}(:,i) - \boldsymbol{a}_i\|_1 \\
&\geq \|\boldsymbol{AX}_{\mathrm{opt}}(:,i)\|_1 - \|\boldsymbol{a}_i\|_1 \\
&\geq \|\boldsymbol{VX}_{\mathrm{opt}}(:,i) + \boldsymbol{NX}_{\mathrm{opt}}(:,i)\|_1 - \|\boldsymbol{a}_i\|_1 \\
&\geq \underbrace{\|\boldsymbol{VX}_{\mathrm{opt}}(:,i)\|_1}_{\text{(A)}} - \underbrace{\|\boldsymbol{NX}_{\mathrm{opt}}(:,i)\|_1}_{\text{(B)}} - \underbrace{\|\boldsymbol{a}_i\|_1}_{\text{(C)}}.
\end{aligned}
$$

The term (A) can be rewritten as

$$
\text{(A)} = \mathbf{1}^\top \boldsymbol{VX}_{\mathrm{opt}}(:,i) = \mathbf{1}^\top \boldsymbol{X}_{\mathrm{opt}}(:,i) = \|\boldsymbol{X}_{\mathrm{opt}}(:,i)\|_1
$$

since $\mathbf{1}^\top V = \mathbf{1}$ by Assumption 1(a) and $V, X_{\mathrm{opt}} \geq \mathbf{0}$. By using Assumptions 1(a) and 1(b), we bound the terms (B) and (C) as follows:

$$(B) \leq \|N\|_1 \|X_{\mathrm{opt}}(:, i)\|_1 \leq \epsilon \|X_{\mathrm{opt}}(:, i)\|_1,$$
$$(C) = \|v_i + n_i\|_1 \leq \|v_i\|_1 + \|n_i\|_1 \leq 1 + \epsilon.$$

Hence, we obtain $1 + 3\epsilon \geq (1 - \epsilon)\|X_{\mathrm{opt}}(:, i)\|_1$, which gives the first inequality of this lemma, since $0 \leq \epsilon < 1$ by Assumption 1(b).

Next, we prove the second inequality. By Lemma 1,

$$\begin{aligned}
2\epsilon \geq \theta = \|A - AX_{\mathrm{opt}}\|_1 &\geq \|a_i - AX_{\mathrm{opt}}(:, i)\|_1 \\
&= \|v_i + n_i - VX_{\mathrm{opt}}(:, i) - NX_{\mathrm{opt}}(:, i)\|_1 \\
&= \|v_i - VX_{\mathrm{opt}}(:, i) + n_i - NX_{\mathrm{opt}}(:, i)\|_1 \\
&\geq \|v_i - VX_{\mathrm{opt}}(:, i)\|_1 - \|n_i - NX_{\mathrm{opt}}(:, i)\|_1 \\
&\geq \|v_i - VX_{\mathrm{opt}}(:, i)\|_1 - \underbrace{(\|n_i\|_1 + \|NX_{\mathrm{opt}}(:, i)\|_1)}_{(A)}.
\end{aligned}$$

By using Assumption 1(b) and the first inequality of this lemma, we bound the term (A) as follows:

$$(A) \leq \|n_i\|_1 + \|N\|_1 \|X_{\mathrm{opt}}(:, i)\|_1 \leq \frac{2\epsilon(1 + \epsilon)}{1 - \epsilon}.$$

We then obtain the second inequality of this lemma.

$\square$

Appendix B   Proof of Lemmas 3 and 4

We use the following lemma to prove Lemmas 3 and 4.

Lemma 11. *Let A satisfy Assumption 1. Let $\phi : R \to N$ be a map such that $w_j = v_{\phi(j)}$ for each $j \in R$. Then, for $j \in R$ and $i = \phi(j) \in N$, we have*

$$\|(1 - \eta + \tilde{\epsilon})w_j - W(:, R \setminus \{j\})z\|_1 \leq 2\tilde{\epsilon} \quad and \quad 1 - \eta + \tilde{\epsilon} \geq 0$$

by letting

$$\eta = H(j, :)X_{\mathrm{opt}}(:, i), \quad z = H(R \setminus \{j\}, :)X_{\mathrm{opt}}(:, i) \quad and \quad \tilde{\epsilon} = \frac{4\epsilon}{1 - \epsilon}$$

where X_{opt} is the optimal solution of problem $\mathsf{P}(A, r)$.

Proof. Let $j \in R$ and $i = \phi(j) \in N$. Lemma 2 tells us that

$$\begin{aligned}
\tilde{\epsilon} \geq \|v_i - VX_{\mathrm{opt}}(:, i)\|_1 &= \|v_i - WHX_{\mathrm{opt}}(:, i)\|_1 \\
&= \|v_{\phi(j)} - WHX_{\mathrm{opt}}(:, i)\|_1 \\
&= \|w_j - \underbrace{WHX_{\mathrm{opt}}(:, i)}_{(A)}\|_1.
\end{aligned}$$

Since

$$\boldsymbol{W}\boldsymbol{H} = \boldsymbol{w}_1\boldsymbol{H}(1,:) + \cdots + \boldsymbol{w}_r\boldsymbol{H}(r,:) = \boldsymbol{w}_j\boldsymbol{H}(j,:) + \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{H}(R\setminus\{j\},:)$$

the term (A) is rewritten as

$$(\mathrm{A}) = \eta \cdot \boldsymbol{w}_j + \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z}.$$

by letting

$$\eta = \boldsymbol{H}(j,:)\boldsymbol{X}_{\mathrm{opt}}(:,i) \in \mathbb{R} \quad \text{and} \quad \boldsymbol{z} = \boldsymbol{H}(R\setminus\{j\},:)\boldsymbol{X}_{\mathrm{opt}}(:,i) \in \mathbb{R}^{r-1}.$$

Accordingly,

$$
\begin{aligned}
\tilde{\epsilon} &\geq \|(1-\eta)\boldsymbol{w}_j - \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z}\|_1 \\
&= \|(1-\eta+\tilde{\epsilon})\boldsymbol{w}_j - \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z} - \tilde{\epsilon}\boldsymbol{w}_j\|_1 \\
&\geq \|(1-\eta+\tilde{\epsilon})\boldsymbol{w}_j - \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z}\|_1 - \tilde{\epsilon}\|\boldsymbol{w}_j\|_1 \\
&= \|(1-\eta+\tilde{\epsilon})\boldsymbol{w}_j - \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z}\|_1 - \tilde{\epsilon} \qquad \text{(by Assumption 1(a)).}
\end{aligned}
$$

Note that $\|\tilde{\epsilon}\boldsymbol{w}_j\|_1 = \tilde{\epsilon}\|\boldsymbol{w}_j\|_1 \geq 0$ holds since $\tilde{\epsilon} = \frac{4\epsilon}{1-\epsilon} \geq 0$ by Assumption 1(b). We thus obtain $\|(1-\eta+\tilde{\epsilon})\boldsymbol{w}_j - \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z}\|_1 \leq 2\tilde{\epsilon}$ for η and $\boldsymbol{z}$ defined above. Moreover, considering that all elements of $\boldsymbol{H}$ are less than or equal to 1 since Assumption 1(a) holds and $\boldsymbol{H} \geq \boldsymbol{0}$, we have

$$
\begin{aligned}
\eta = \boldsymbol{H}(j,:)\boldsymbol{X}_{\mathrm{opt}}(:,i) &\leq \boldsymbol{1}^\top \boldsymbol{X}_{\mathrm{opt}}(:,i) \\
&= \|\boldsymbol{X}_{\mathrm{opt}}(:,i)\|_1 \qquad\qquad \text{(by } \boldsymbol{X}_{\mathrm{opt}} \geq \boldsymbol{0}\text{)} \\
&\leq 1 + \tilde{\epsilon} \qquad\qquad\qquad\quad \text{(by Lemma 2).}
\end{aligned}
$$

This gives $1 - \eta + \tilde{\epsilon} \geq 0$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

We are now able to prove Lemma 3.

(Proof of Lemma 3). Since we put Assumption 1 on $\boldsymbol{A}$, it can be written as $\boldsymbol{A} = \boldsymbol{V} + \boldsymbol{N} \in \mathbb{R}^{d\times n}$ for $\boldsymbol{V} \in \mathbb{R}_+^{d\times n}$ and $\boldsymbol{N} \in \mathbb{R}^{d\times n}$. Since $\boldsymbol{V}$ is r-separable of the form $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{H} = \boldsymbol{W}[\boldsymbol{I}, \bar{\boldsymbol{H}}]\boldsymbol{\Pi}$ shown in (1), there is a map $\phi : R \to N$ such that $\boldsymbol{w}_j = \boldsymbol{v}_{\phi(j)}$ for each $j \in R$. Hence, the basis index I of $\boldsymbol{V}$ is given as $I = \{\phi(1), \ldots, \phi(j)\}$. Let $i = \phi(j)$ for $j \in R$. Lemma 11 tells us that

$$\|(1-\eta+\tilde{\epsilon})\boldsymbol{w}_j - \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z}\|_1 \leq 2\tilde{\epsilon} \quad \text{and} \quad 1-\eta+\tilde{\epsilon} \geq 0$$

hold for

$$\eta = \boldsymbol{H}(j,:)\boldsymbol{X}_{\mathrm{opt}}(:,i), \quad \boldsymbol{z} = \boldsymbol{H}(R\setminus\{j\},:)\boldsymbol{X}_{\mathrm{opt}}(:,i) \quad \text{and} \quad \tilde{\epsilon} = \frac{4\epsilon}{1-\epsilon}.$$

First, consider the case where $1 - \eta + \tilde{\epsilon} > 0$. We find that

$$
\begin{aligned}
2\tilde{\epsilon} &\geq \|(1-\eta+\tilde{\epsilon})\boldsymbol{w}_j - \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z}\|_1 \\
&= (1-\eta+\tilde{\epsilon})\|\boldsymbol{w}_j - \boldsymbol{W}(:,R\setminus\{j\})\boldsymbol{z}'\|_1 \qquad \text{(by letting } \boldsymbol{z}' = \boldsymbol{z}/(1-\eta+\tilde{\epsilon})\text{)} \\
&\geq (1-\eta+\tilde{\epsilon})\kappa \qquad\qquad\qquad\qquad\qquad\quad \text{(by the definition of } \kappa\text{).}
\end{aligned}
$$

Note that $\boldsymbol{z}' \geq 0$ since $\boldsymbol{z} = \boldsymbol{H}(R \setminus \{j\}, :)\boldsymbol{X}_{\text{opt}}(:, i) \geq 0$ and $1 - \eta + \tilde{\epsilon} > 0$. Accordingly, we obtain a lower bound on η,

$$\eta \geq 1 + \frac{(\kappa - 2)\tilde{\epsilon}}{\kappa}. \tag{18}$$

We can upper bound η using $\boldsymbol{p}(i)$. Since $\boldsymbol{w}_j = \boldsymbol{v}_{\phi(j)}$ and $i = \phi(j)$, we have $\boldsymbol{H}(:, i) = \boldsymbol{e}_j$, and thus $\boldsymbol{H}(j, i) = 1$. In light of this, we rewrite η as

$$\eta = \boldsymbol{H}(j, :)\boldsymbol{X}_{\text{opt}}(:, i) = \boldsymbol{X}_{\text{opt}}(i, i) + \boldsymbol{H}(j, N \setminus \{i\})\boldsymbol{X}_{\text{opt}}(N \setminus \{i\}, i)$$
$$= \boldsymbol{p}(i) + \underbrace{\boldsymbol{H}(j, N \setminus \{i\})\boldsymbol{X}_{\text{opt}}(N \setminus \{i\}, i)}_{(A)}$$

and bound the term (A) as follows:

$$\begin{aligned} (A) &\leq \beta \cdot \boldsymbol{1}^\top \boldsymbol{X}_{\text{opt}}(N \setminus \{i\}, i) && \text{(by the definition of } \beta\text{)} \\ &= \beta(\|\boldsymbol{X}_{\text{opt}}(:, i)\|_1 - \boldsymbol{X}_{\text{opt}}(i, i)) && \text{(by } \boldsymbol{X}_{\text{opt}} \geq \boldsymbol{0}\text{)} \\ &\leq \beta(1 + \tilde{\epsilon} - \boldsymbol{p}(i)) && \text{(by Lemma 2).} \end{aligned}$$

We thus obtain an upper bound on η,

$$\eta \leq (1 - \beta)\boldsymbol{p}(i) + \beta(1 + \tilde{\epsilon}). \tag{19}$$

The bounds (18) and (19) yield

$$1 + \frac{(\kappa - 2)\tilde{\epsilon}}{\kappa} \leq (1 - \beta)\boldsymbol{p}(i) + \beta(1 + \tilde{\epsilon}) \Leftrightarrow \boldsymbol{p}(i) \geq 1 + \tilde{\epsilon} - \frac{2\tilde{\epsilon}}{\kappa(1 - \beta)}.$$

Assumption 1(b) implies $\tilde{\epsilon} = 4\epsilon/(1-\epsilon) \geq 0$. Recall that $i = \phi(j)$ for $j \in R$ and $I = \{\phi(1), \ldots, \phi(r)\}$. Hence, from the inequality above, we obtain $\boldsymbol{p}(i) \geq 1 - \frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)}$ for every $i \in I$.

Next, consider the case where $1 - \eta + \tilde{\epsilon} = 0$. By inequality (19), we have

$$1 + \tilde{\epsilon} = \eta \leq (1 - \beta)\boldsymbol{p}(i) + \beta(1 + \tilde{\epsilon}),$$

which gives $\boldsymbol{p}(i) \geq 1 + \tilde{\epsilon}$. Here, $\tilde{\epsilon} \geq 0$ and $\frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)} \geq 0$ by Assumption 1(b), $\kappa > 0$ and $\beta < 1$. We thus obtain $\boldsymbol{p}(i) \geq 1 - \frac{8\epsilon}{\kappa(1-\beta)(1-\epsilon)}$ for every $i \in I$. $\square$

Remark 2. *In the proof above, to find a lower bound on η, we have used the observation that $1 - \eta + \tilde{\epsilon}$ is positive or zero, which is not taken into account in the proof of Lemma 2.2 of [9].*

Let us move on to prove Lemma 4. To do so, we prove the following lemma.

Lemma 12. *Let $\boldsymbol{A}$ satisfy Assumption 1. Let T_j be an anchor with parameter μ satisfying $\epsilon \leq \mu$. Then, for $j \in R$, we have*

$$\max_{u \in T_j^c} \boldsymbol{H}(j, u) < 1 - \frac{\mu}{2}.$$

Proof. For any $u \in T_j^c$,

$$\begin{aligned}
2\mu < \|\boldsymbol{w}_j - \boldsymbol{a}_u\|_1 &= \|\boldsymbol{w}_j - \boldsymbol{W}\boldsymbol{H}(:,u) - \boldsymbol{n}_u\|_1 \\
&\leq \|\boldsymbol{w}_j - \boldsymbol{W}\boldsymbol{H}(:,u)\|_1 + \|\boldsymbol{n}_u\|_1 \\
&\leq \|\boldsymbol{w}_j - \boldsymbol{W}\boldsymbol{H}(:,u)\|_1 + \epsilon \\
&\leq \|\boldsymbol{w}_j - \boldsymbol{W}\boldsymbol{H}(:,u)\|_1 + \mu.
\end{aligned}$$

Hence, $\|\boldsymbol{w}_j - \boldsymbol{W}\boldsymbol{H}(:,u)\|_1 > \mu$ holds. Furthermore,

$$\begin{aligned}
\mu &< \|\boldsymbol{w}_j - \boldsymbol{W}\boldsymbol{H}(:,u)\|_1 \\
&= \|\boldsymbol{w}_j - \boldsymbol{w}_j \boldsymbol{H}(j,u) - \boldsymbol{W}(:,R \setminus \{j\})\boldsymbol{H}(R \setminus \{j\},u)\|_1 \\
&\leq (1 - \boldsymbol{H}(j,u))\|\boldsymbol{w}_j\|_1 + \|\boldsymbol{W}(:,R \setminus \{j\})\|_1 \|\boldsymbol{H}(R \setminus \{j\},u)\|_1 \\
&= 1 - \boldsymbol{H}(j,u) + \|\boldsymbol{H}(R \setminus \{j\},u)\|_1 & \text{(by Assumption 1(a))} \\
&= 1 - 2\boldsymbol{H}(j,u) + \|\boldsymbol{H}(:,u)\|_1 & \text{(by } \boldsymbol{H} \geq \boldsymbol{0}\text{)} \\
&= 2 - 2\boldsymbol{H}(j,u) & \text{(by Assumption 1(a))}.
\end{aligned}$$

Note that $\|(1-\boldsymbol{H}(j,u))\boldsymbol{w}_j\|_1 = (1-\boldsymbol{H}(j,u))\|\boldsymbol{w}_j\|_1$ holds since $1-\boldsymbol{H}(j,u) \geq 0$ by Assumption 1(a) and $\boldsymbol{H} \geq \boldsymbol{0}$. It follows from the inequality above that $\max_{u \in T_j^c} \boldsymbol{H}(j,u) < 1 - \frac{\mu}{2}$ holds. $\square$

In light of this, we can prove Lemma 4 in a similar way as Lemma 3. The proof is almost the same, except the evaluation of the upper bound on η.

(Proof of Lemma 4). We use Lemma 11. Let $\phi : R \to N$ be a map such that $\boldsymbol{w}_j = \boldsymbol{v}_{\phi(j)}$ for each $j \in R$. The lemma tells us that, for $j \in R$ and $i = \phi(j) \in N$, we have

$$\|(1 - \eta + \tilde{\epsilon})\boldsymbol{w}_j - \boldsymbol{W}(:,R \setminus \{j\})\boldsymbol{z}\|_1 \leq 2\tilde{\epsilon} \quad \text{and} \quad 1 - \eta + \tilde{\epsilon} \geq 0$$

where η, $\boldsymbol{z}$ and $\tilde{\epsilon}$ are as shown in the lemma.

First, consider the case where $1 - \eta + \tilde{\epsilon} > 0$. As in the proof of Lemma 3, we have $2\tilde{\epsilon} \geq \|(1 - \eta + \tilde{\epsilon})\boldsymbol{w}_j - \boldsymbol{W}(:,R \setminus \{j\})\boldsymbol{z}\|_1 \geq (1 - \eta + \tilde{\epsilon})\kappa$, which gives a lower bound on η, as shown in (18). We can upper bound η using $\text{score}(T_j, \boldsymbol{p})$. Write η as

$$\eta = \boldsymbol{H}(j,:)\boldsymbol{X}_{\text{opt}}(:,i) = \underbrace{\boldsymbol{H}(j,T_j)\boldsymbol{X}_{\text{opt}}(T_j,i)}_{(A)} + \underbrace{\boldsymbol{H}(j,T_j^c)\boldsymbol{X}_{\text{opt}}(T_j^c,i)}_{(B)}.$$

The term (A) is bounded as follows:

$$\begin{aligned}
(A) &\leq \boldsymbol{1}^\top \boldsymbol{X}_{\text{opt}}(T_j,i) & \text{(by Assumption 1(a) and } \boldsymbol{H} \geq \boldsymbol{0}\text{)} \\
&= \|\boldsymbol{X}_{\text{opt}}(T_j,i)\|_1 & \text{(by } \boldsymbol{X}_{\text{opt}} \geq \boldsymbol{0}\text{)}.
\end{aligned}$$

The term (B) is bounded as follows:

$$\begin{aligned}
(B) &< \left(1 - \frac{\mu}{2}\right) \boldsymbol{1}^\top \boldsymbol{X}_{\text{opt}}(T_j^c,i) & \text{(by Lemma 12)} \\
&= \left(1 - \frac{\mu}{2}\right) \|\boldsymbol{X}_{\text{opt}}(T_j^c,i)\|_1 & \text{(by } \boldsymbol{X}_{\text{opt}} \geq \boldsymbol{0}\text{)} \\
&= \left(1 - \frac{\mu}{2}\right) (\|\boldsymbol{X}_{\text{opt}}(:,i)\|_1 - \|\boldsymbol{X}_{\text{opt}}(T_j,i)\|_1) \\
&\leq \left(1 - \frac{\mu}{2}\right) (1 + \tilde{\epsilon} - \|\boldsymbol{X}_{\text{opt}}(T_j,i)\|_1) & \text{(by Lemma 2)}.
\end{aligned}$$

We then find that

$$\eta < \|\boldsymbol{X}_{\text{opt}}(T_j, i)\|_1 + \left(1 - \frac{\mu}{2}\right)(1 + \tilde{\epsilon} - \|\boldsymbol{X}_{\text{opt}}(T_j, i)\|_1) = \frac{\mu}{2}\|\boldsymbol{X}_{\text{opt}}(T_j, i)\|_1 + \left(1 - \frac{\mu}{2}\right)(1 + \tilde{\epsilon}).$$

Here,

$$\|\boldsymbol{X}_{\text{opt}}(T_j, i)\|_1 = \sum_{u \in T_j} \boldsymbol{X}_{\text{opt}}(u, i) \leq \sum_{u \in T_j} \boldsymbol{X}_{\text{opt}}(u, u) = \text{score}(T_j, \boldsymbol{p}).$$

since $\boldsymbol{X}_{\text{opt}}(u, i) \leq \boldsymbol{X}_{\text{opt}}(u, u)$ and $\boldsymbol{X}_{\text{opt}} \geq \boldsymbol{0}$ by the third and fourth constraints of problem P. Accordingly, we obtain

$$\eta < \frac{\mu}{2}\text{score}(T_j, \boldsymbol{p}) + \left(1 - \frac{\mu}{2}\right)(1 + \tilde{\epsilon}). \tag{20}$$

The bounds (18) and (20) yield

$$1 + \frac{(\kappa - 2)\tilde{\epsilon}}{\kappa} \leq \frac{\mu}{2}\text{score}(T_j, \boldsymbol{p}) + \left(1 - \frac{\mu}{2}\right)(1 + \tilde{\epsilon}) \Leftrightarrow \text{score}(T_j, \boldsymbol{p}) \geq 1 + \tilde{\epsilon} - \frac{4\tilde{\epsilon}}{\kappa\mu}.$$

Here, $\tilde{\epsilon} \geq 0$. We thus obtain $\text{score}(T_j, \boldsymbol{p}) \geq 1 - \frac{16\epsilon}{\kappa\mu(1-\epsilon)}$ for every $j \in R$.

Next, consider the case where $1 - \eta + \tilde{\epsilon} = 0$. By inequality (20), we have

$$1 + \tilde{\epsilon} = \eta < \frac{\mu}{2}\text{score}(T_j, \boldsymbol{p}) + \left(1 - \frac{\mu}{2}\right)(1 + \tilde{\epsilon}),$$

which gives $\text{score}(T_j, \boldsymbol{p}) > 1 + \tilde{\epsilon}$. Here, $\tilde{\epsilon} \geq 0$ and $\frac{16\epsilon}{\kappa\mu(1-\epsilon)} \geq 0$. We thus obtain $\text{score}(T_j, \boldsymbol{p}) \geq 1 - \frac{16\epsilon}{\kappa\mu(1-\epsilon)}$ for every $j \in R$.

$\square$

Appendix C   Proof of Corollary 1

(Proof of Corollary 1). Since Assumption 1(a) holds, we have the bounds $0 \leq \kappa \leq 1$ and $0 \leq \omega \leq 2$ shown in (3) and (4). Also, since Assumption 1(b) holds, we have $\epsilon \geq 0$. Hence, the bounds imposed on ϵ in the two cases imply $\kappa > 0$. Accordingly, κ and ω satisfy $0 < \kappa \leq 1$ and $0 \leq \omega \leq 2$. In addition, they satisfy the relation $\kappa \leq \omega$ shown in (2).

  $\boxed{\text{Former case}}$ (a) We only have to prove $\epsilon < 1$ since $\epsilon \geq 0$ by Assumption 1(b). The bounds $\kappa \leq 1$ and $\omega \leq 2$ imply

$$\epsilon < \frac{\omega\kappa}{578(r+1)} \leq \frac{1}{578}. \tag{21}$$

Hence, ϵ satisfies $\epsilon < 1$. (b) Since $r, \kappa, \xi > 0$ and $\epsilon \geq 0$, we have

$$\mu = \frac{17(r+1)\epsilon}{\kappa} + \xi > 0.$$

By using the bound on ϵ, we can put a bound on μ:

$$\mu = \frac{17(r+1)\epsilon}{\kappa} + \xi < \frac{\omega}{34} + \xi.$$

Since $\xi < \kappa/35$ and $\kappa \leq \omega$, we have

$$\mu < \frac{\omega}{34} + \frac{\kappa}{35} \leq \frac{\omega}{17}.$$

Hence, $0 < \mu < \omega/17$ holds. (c) Since $\xi > 0$ and $\kappa \leq 1$, we have

$$\mu = \frac{17(r+1)\epsilon}{\kappa} + \xi > \frac{17(r+1)\epsilon}{\kappa} \geq 34\epsilon.$$

Thus, $\epsilon \leq \mu$ holds. (d) The corollary satisfies the hypotheses of Lemma 4. This is because Assumption 1(b) is not violated by the bound on ϵ that we put in part (a); $\kappa > 0$ holds, as explained at the beginning of the proof; and $\mu \neq 0$ and $\epsilon \leq \mu$ hold, as shown in parts (b) and (c). Accordingly,

$$\mathrm{score}(T_j, \boldsymbol{p}) \geq 1 - \frac{16\epsilon}{\kappa\mu(1-\epsilon)}$$

holds for every $j \in R$. If $\epsilon = 0$, then, $\mathrm{score}(T_j, \boldsymbol{p}) \geq 1 > r/(r+1)$. We thus assume $\epsilon > 0$. The bound on ϵ in (21) implies $\epsilon < 1/17$ and we have

$$\epsilon < \frac{1}{17} \Leftrightarrow \frac{16}{1-\epsilon} < 17.$$

Write the value of μ as $\mu = \lambda + \xi$ by letting $\lambda = 17(r+1)\epsilon/\kappa$. We find that

$$\mathrm{score}(T_j, \boldsymbol{p}) \geq 1 - \frac{16\epsilon}{\kappa\mu(1-\epsilon)} > 1 - \frac{17\epsilon}{\kappa\mu} > 1 - \frac{17\epsilon}{\kappa\lambda} = \frac{r}{r+1}$$

where the third inequality uses $\mu = \lambda + \xi$ and $\lambda, \xi > 0$, and the equality uses $\lambda = 17(r+1)\epsilon/\kappa$.

$\boxed{\text{Latter case}}$ (a) As mentioned in the former case, we only have to prove $\epsilon < 1$. From the bound $\kappa \leq 1$ and $289 = 17^2$, we obtain a bound on ϵ,

$$\epsilon < \frac{\kappa^2}{17^2(r+1)^2} \leq \left(\frac{1}{34}\right)^2. \tag{22}$$

Hence, ϵ satisfies $\epsilon < 1$. (b) Since $\xi > 0$ and $\epsilon \geq 0$, we have $\mu = \sqrt{\epsilon} + \xi > 0$. Here, ξ satisfies $\xi < \kappa/35 < \kappa/34$, and we have $\kappa \leq \omega$. Hence, using the bound on ϵ, we obtain a bound on μ,

$$\mu = \sqrt{\epsilon} + \xi < \frac{\kappa}{17(r+1)} + \xi < \frac{\kappa}{17} \leq \frac{\omega}{17}.$$

Hence, $0 < \mu < \omega/17$ holds. (c) Two functions $f_1(x) = x$ and $f_2(x) = \sqrt{x}$ satisfy $f_1(x) \leq f_2(x)$ for $0 \leq x \leq 1$. Since ξ satisfies $\xi > 0$ and ϵ satisfies $0 \leq \epsilon < 1$ as shown in part (a), we have $\epsilon \leq \mu = \sqrt{\epsilon} + \xi$. (d) The bound on ϵ in (22) implies $\epsilon < 1/17$. We can thus prove this part in the same way as part (d) of the former case.

$\square$

Acknowledgments

References

[1] U. M. C. Araújo, B. T. C. Saldanha, R. K. H. Galvão, T. Yoneyama, H. C. Chame, and V. Visani. The successive projections algorithm for variable selection in spectroscopic multi-component analysis. *Chemometrics and Intelligent Laboratory Systems*, 57(2):65–73, 2001.

[2] S. Arora, R. Ge, Y. Halpern, D. Mimno, and A. Moitra. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

[3] S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization – Provably. In *Proceedings of the 44th symposium on Theory of Computing (STOC)*, pages 145–162, 2012.

[4] S. Arora, R. Ge, and A. Moitra. Learning topic models – Going beyond SVD. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–10, 2012.

[5] V. Bittorf, B. Recht, C. Re, and J. A. Tropp. Factoring nonnegative matrices with linear programs. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1223–1231, 2012.

[6] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Proceedings of Advances in Neural Information Processing Systems 16 (NIPS)*, pages 1141–1148, 2003.

[7] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Processing Magazine*, 36(2):59–80, 2019.

[8] N. Gillis. Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13:3349–3386, 2012.

[9] N. Gillis. Robustness analysis of Hottopixx, a linear programming model for factoring nonnegative matrices. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1189–1212, 2013.

[10] N. Gillis. Successive nonnegative projection algorithm for robust nonnegative blind source separation. *SIAM Journal on Imaging Sciences*, 7(2):1420–1450, 2014.

[11] N. Gillis. Separable simplex-structured matrix factorization: Robustness of combinatorial approaches. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5521–5525, 2019.

[12] N. Gillis. *Nonnegative Matrix Factorization*. SIAM, 2020.

[13] N. Gillis and R. Luce. Robust near-separable nonnegative matrix factorization using linear optimization. *Journal of Machine Learning Research*, 15:1249–1280, 2014.

[14] N. Gillis and R. Luce. A fast gradient method for nonnegative sparse regression with self-dictionary. *IEEE Transactions on Image Processing*, 27(1):24–37, 2018.

[15] N. Gillis and S. A. Vavasis. Fast and robust recursive algorithms for separable nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):698–714, 2014.

[16] W.-K. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C.-Y. Chi. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Processing Magazine*, 31(2):67–81, 2014.

[17] T. Mizutani. Ellipsoidal rounding for nonnegative matrix factorization under noisy separability. *Journal of Machine Learning Research*, 15:1011–1039, 2014.

[18] J. M. P. Nascimento and J. M. B. Dias. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, 2005.

[19] S. A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal of Optimization*, 20(3):1364–1377, 2009.