SIAM J. MATH. DATA SCI.
Vol. 8, No. 3, pp. 738–765

An Extrapolated and Provably Convergent Algorithm for Nonlinear Matrix Decomposition with the ReLU Function*

Nicolas Gillis†, Margherita Porcelli‡, and Giovanni Seraghiti§

Abstract. ReLU matrix decomposition (RMD) is the following problem: given a sparse, nonnegative matrix X and a factorization rank r, identify a rank-r matrix Θ such that $X \approx \max(0, \Theta)$. RMD is a particular instance of nonlinear matrix decomposition that finds application in data compression, matrix completion with entries missing not at random, and manifold learning. The standard RMD model minimizes the least squares error, that is, $\|X - \max(0, \Theta)\|_F^2$. The corresponding optimization problem, least squares RMD, is nondifferentiable and highly nonconvex. This motivated Saul to propose an alternative model, dubbed Latent-RMD, where a latent variable Z is introduced and satisfies $\max(0, Z) = X$ while minimizing $\|Z - \Theta\|_F^2$ [*SIAM J. Math. Data Sci.*, 4 (2022), pp. 431–463]. Our first contribution is to show that the two formulations may yield different low-rank solutions Θ. We then consider a reparametrization of the Latent-RMD, called 3B-RMD, in which Θ is substituted by a low-rank product WH, where W has r columns and H has r rows. Our second contribution is to prove the convergence of a block coordinate descent approach applied to 3B-RMD.

Key words. low-rank matrix approximations, nonlinear matrix decomposition, rectified linear unit, block coordinate descent, extrapolation

MSC codes. 15A23, 65F55, 68Q25, 90C26, 65K0

DOI. 10.1137/25M1746859

1. Introduction.

Nonlinear matrix decomposition (NMD) can be defined as follows: given a data matrix $X \in \mathbb{R}^{m \times n}$ and a factorization rank r, NMD looks for a rank-r matrix $\Theta \in \mathbb{R}^{m \times n}$ such that $X \approx f(\Theta)$, where $f(\Theta)$ applies the scalar function f elementwise on Θ, that is, $[f(\Theta)]_{i,j} = f(\Theta_{i,j})$ for all i, j. The choice of f depends on the type of data at hand. For example, one might choose $f(x) = \text{sign}(x)$ for binary data in $\{-1, 1\}$ [12], or $f(x) = x^2$ for nonnegative data with a multiplicative structure, resulting in the Hadamard product decomposition which is efficient for the compression of dense images [10, 19, 36] and for the representation of probabilistic circuits [23]. For a more detailed description of the

*Received by the editors April 2, 2025; accepted for publication (in revised form) April 15, 2026; published electronically July 24, 2026.
https://doi.org/10.1137/25M1746859
Funding: The first and third authors acknowledge the support by the European Union (ERC consolidator, eLinoR, 101085607). The work of the second and third authors was partially supported by INdAM-GNCS through Progetti di Ricerca 2025 and 2026. The research of the second author was partially supported by the Italian Ministry of University and Research (MUR) through PRIN 2022, "MOLE: Manifold constrained Optimization and LEarning," code 2022ZK5ME7 MUR D.D. financing decree 20428 of November 6, 2024 (CUP B53C24006410006).

†University of Mons, 7000 Mons, Belgium (nicolas.gillis@umons.ac.be).
‡Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, 50134 Firenze, Italia, and ISTI–CNR, Pisa, Italia (margherita.porcelli@unifi.it).
§Corresponding author. University of Mons, 7000 Mons, Belgium, and Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, 50134 Firenze, Italia (giovanni.seraghiti@umons.ac.be).

choices for the function f, see [4]. In this work, we assume X to be nonnegative and sparse, opting for f to be the ReLU function, $f(\cdot) = \max(0, \cdot)$. This NMD, with $X \approx \max(0, \Theta)$, is referred to as the ReLU matrix decomposition (RMD). This model has a close connection with a two-layer neural network, using ReLU as the activation function in the output layer; see [27, 28] for more details. Intuitively, RMD aims at finding a matrix Θ that substitutes the zero entries of the target matrix X with negative values in order to decrease its rank. Note that some degree of sparsity on the matrix X is an essential condition for RMD to be meaningful in practice. In fact, if X has mostly positive entries, then Θ will also have mostly positive entries, and the ReLU nonlinear activation will be active only for a few entries, leading to a model that is similar to the truncated singular value decomposition (TSVD).

Minimizing the least squares error leads to an optimization problem referred to as LS-RMD:

$$\min_{\Theta} \|X - \max(0, \Theta)\|_F^2 \quad \text{such that} \quad \operatorname{rank}(\Theta) \leq r. \tag{1.1}$$

It is interesting to note that LS-RMD will always lead to a lower error than the TSVD. In fact, let X_r be an optimal rank-r approximation of $X \geq 0$ (e.g., obtained with the TSVD); then $\|X - X_r\|_F^2 \geq \|X - \max(X_r, 0)\|_F^2$ since negative entries in X_r, if any, are replaced by zeros which better approximate the corresponding nonnegative entries in X.

LS-RMD was introduced by Saul in [27, 28]. It is nondifferentiable and nonconvex, and hence difficult to tackle. Indeed, to the best of our knowledge, there exist only a few algorithms that address (1.1) directly [4, 5]. All other existing approaches rely on Saul's alternative formulation, dubbed Latent-RMD and defined as

$$\min_{Z,\Theta} \|Z - \Theta\|_F^2 \quad \text{such that} \quad \begin{cases} \operatorname{rank}(\Theta) \leq r, \\ \max(0, Z) = X. \end{cases} \tag{1.2}$$

The objective function in (1.2) is differentiable, while the subproblems in the variables Z and Θ can be solved up to global optimality; see section 4. However, for a general matrix X, the link between the solutions of (1.1) and (1.2) has not been explored in the literature. Our first contribution, in section 3, is to better understand the connection between the two.

If we reparametrize $\Theta = WH$ in (1.2), where $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$, as suggested in [29], we obtain the following equivalent problem:

$$\min_{Z,W,H} \|Z - WH\|_F^2 \quad \text{such that} \quad \max(0, Z) = X. \tag{1.3}$$

We denote this reformulation 3B-RMD, where 3B stands for three blocks of variables. The objective function in (1.3) is again jointly nonconvex, but it is convex with respect to each matrix variable, W, H, and Z, making it more friendly from an algorithmic point of view, as the rank constraint is not explicitly involved. Problems (1.2) and (1.3) can be tackled using the block coordinate descent (BCD) method, optimizing each matrix variable alternatively [29, 33].

Outline and contribution. Section 2 motivates the study of RMD by presenting several applications. In section 3, we clarify the relationship between LS-RMD and Latent-RMD. In particular, we provide a matrix X for which the infimum of the rank-1 Latent-RMD is

not attained, showing the ill-posedness of this model (Example 1). On the other hand, for the same matrix X, LS-RMD has a nonempty set of optimal rank-1 solutions (Example 2). This shows that, in general, the two models do not share the same set of optimal low-rank solutions, that is, the sets

$$\{\Theta^* \mid \Theta^* \text{ optimal for LS-RMD (1.1)}\} \quad \text{and} \quad \{\widehat{\Theta} \mid (\widehat{Z}, \widehat{\Theta}) \text{ optimal for Latent-RMD (1.2)}\}$$

do not necessarily coincide. However, we show that any feasible point $(\bar{Z}, \bar{\Theta})$ of the Latent-RMD model satisfies the relation

$$\|X - \max(0, \bar{\Theta})\|_F \leq 4\|\bar{Z} - \bar{\Theta}\|_F.$$

This guarantees that any approximate solution of Latent-RMD provides an approximate solution for LS-RMD whose residual is at most four times that of Latent-RMD.

In section 4, we first briefly review the state-of-the-art algorithms for solving Latent-RMD and 3B-RMD (section 4.1). We then prove the convergence of the BCD scheme for solving 3B-RMD (section 4.2), demonstrating that it falls within the framework of BCD with strictly quasi-convex subproblems [17]. In addition, we introduce a novel extrapolated version of BCD (eBCD; section 4.3), adapting the well-known LMaFit algorithm for matrix completion [35]. We prove the subsequence convergence of eBCD (Theorem 4.7). To the best of our knowledge, BCD and eBCD are the first algorithms for solving 3B-RMD with convergence guarantees.

In section 5, we provide extensive numerical experiments comparing our new approach, eBCD, with the state of the art. We test the algorithms on various applications: matrix completion with ReLU sampling, Euclidean distance matrix completion, compression of sparse data, and low-dimensional embedding. In particular, our experiments show that eBCD consistently accelerates BCD and performs favorably compared to the state of the art, while having stronger theoretical guarantees.

2. Applications of RMD.

In this section, we present some applications of RMD, illustrating how nonlinear models can be used in a variety of contexts.

2.1. Compression of sparse data.

The first application is the compression of sparse data. Let $X \in \mathbb{R}^{m \times n}$ be a sparse and nonnegative matrix with $\text{nnz}(X)$ nonzero entries, and let $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$ be such that $X \approx \max(0, WH)$. As long as $r(m+n) < \text{nnz}(X)$, storing the factors W and H requires less memory than storing X. RMD typically outperforms standard linear compression techniques, including the TSVD on sparse data, such as sparse images or dictionaries [5, 27, 28, 29, 33]. An interesting example is the identity matrix of any dimension, which can be exactly reconstructed using a rank-3 RMD [12, 28].

2.2. Matrix completion with ReLU sampling.

The class of matrix completion problems with entries missing not at random (MNR) is characterized by the probability of an entry being missing that depends on the matrix itself. For example, when completing a survey, people are likely to share nonsensitive information such as their name or surname, while they might be more hesitant to disclose their phone number. Therefore, we might expect the missing entries to be concentrated in some areas of the matrix more than others. Even though MNR matrix completion is still a relatively unexplored area, dropping the assumption that missing entries are independent of the matrix is a challenging problem that attracts more

and more researchers, as the increasing number of related works testifies [7, 8, 14, 22, 24]. Among them, Liu et al. [22] investigate the matrix completion problem when only positive values are observed, namely matrix completion with ReLU sampling. This instance of matrix completion falls within the MNR problems since the missing entries depend on the sign of the target matrix. Assume that X is a nonnegative and sparse matrix and that there exists a rank-r matrix Θ such that $X = \max(0, \Theta)$. In this case, finding an RMD is equivalent to recovering the negative values observing the positive ones; thus, it is equivalent to matrix completion with ReLU sampling. This implies that we can use the algorithms for RMD to address this matrix completion problem, as illustrated in [22].

Inspired by the connection to ReLU sampling matrix completion, we introduce an equivalent formulation of 3B-RMD that we will use throughout the paper. Let $\Omega = \{(i,j) \mid X_{ij} > 0\}$ be the set of positive entries of X, and let $P_\Omega(X)$ be equal to X in Ω and zero elsewhere. We rewrite 3B-RMD in (1.3) as

$$\min_{Z,W,H} \frac{1}{2}\|Z - WH\|_F^2 \quad \text{such that} \quad P_\Omega(Z) = P_\Omega(X), \quad P_{\Omega^C}(Z) \leq 0, \tag{2.1}$$

where Ω^C is the complement of Ω and a reformulation of the constraint $\max(0, Z) = X$ is introduced. To our knowledge, the problem (2.1), without the constraint $P_{\Omega^C}(Z) \leq 0$, was first addressed in [35] in the context of matrix completion. The novelties in our model are the additional information that the missing entries have negative signs, which translates into the constraints $P_{\Omega^C}(Z) \leq 0$, and the explicit dependence of the known index set Ω on the target matrix X. Since the formulations (1.3) and (2.1) are equivalent, we will refer to both as 3B-RMD.

2.3. Euclidean distance matrix completion. We provide one practical example of the MNR matrix completion problem that can be solved using RMD. Specifically, we address the recovery of a low-rank matrix for which only the smallest or the largest entries are observed. Let Θ be a low-rank matrix, and assume that we observe all the entries in Θ smaller than a known threshold d. We want to recover the missing entries, which are the ones larger than d. We can model this problem using a rank-1 modified RMD: $X = \max(0, dee^T - \Theta)$, where e denotes the vector of all ones of appropriate dimension. The matrix X is nonzero exactly in the entries where Θ is smaller than d. Of course, an analogous model can be derived when only the largest entries are observed. The 3B-RMD formulation in (2.1) can be easily adapted to include the rank-1 modification:

$$\min_{Z,W,H} \frac{1}{2}\|dee^T - WH - Z\|_F^2 \quad \text{such that} \quad P_\Omega(Z) = P_\Omega(X), \quad P_{\Omega^C}(Z) \leq 0. \tag{2.2}$$

The same idea applies also to the Latent-RMD and all the algorithms can be adjusted to take into account the additional rank-one term. This formulation is particularly interesting when the matrix Θ contains the squared distances between a collection of points $p_1, \ldots, p_n \in \mathbb{R}^d$. This problem is often referred to as Euclidean distance matrix completion [1, 13, 20, 31] and is particularly meaningful in sensor network localization problems where sensors can only communicate with nearby sensors. Let $P = [p_1, \ldots, p_n] \in \mathbb{R}^{d \times n}$; then the corresponding Euclidean distance matrix is $\Theta_{i,j} = \|p_i - p_j\|_2^2$ for all i, j, so that $\Theta = e\text{diag}(P^TP)^T + \text{diag}(P^TP)e^T + 2P^TP$

and it has at most rank $r = d + 2$ and thus the rank of the factorization in (2.2) is known in advance. We will show in section 5 that this slightly modified RMD allows us to solve Euclidean distance matrix completion in this context.

2.4. Manifold learning. Manifold learning aims at learning a norm-preserving and neighborhood-preserving mapping of high-dimensional inputs into a lower-dimensional space [21]. Given a collection of points $\{z_1, \dots, z_m\}$ in $\mathbb{R}^N$, Saul [27] looks for a faithful representation $\{y_1, \dots, y_m\}$ in $\mathbb{R}^r$ with $r < N$. Letting $\tau \in (0, 1)$, an embedding is τ-faithful if

$$\max(0, \langle y_i, y_j \rangle - \tau \|y_i\| \|y_j\|) = \max(0, \langle z_i, z_j \rangle - \tau \|z_i\| \|z_j\|). \tag{2.3}$$

Equation (2.3) implies that the embedding must preserve norms (take $i = j$ to obtain $(1 - \tau)\|y_i\|^2 = (1-\tau)\|z_i\|^2$) as well as the small angles between two points (z_i, z_j): when $\cos(z_i, z_j) > \tau$, we need $\cos(z_i, z_j) = \cos(y_i, y_j)$. Large angles have to remain large but do not need to be preserved, that is, $\cos(z_i, z_j) \leq \tau$ only requires $\cos(y_i, y_j) \leq \tau$. The condition (2.3) defines a similarity between points based on norm and angles, which differs substantially from the more common concept of pairwise distances. This makes it suitable, for example, in text analysis where the z_i's are sparse vectors of word counts.

This embedding strategy is dubbed threshold similarity matching (TSM). Saul proposes three main steps to compute such an embedding. First, for each couple of points (z_i, z_j), compute the right-hand side in (2.3) and construct the sparse, nonnegative similarity matrix

$$X_{ij} = \max(0, \langle z_i, z_j \rangle - \tau \|z_i\| \|z_j\|) \quad \forall i, j.$$

Second, compute an RMD, $X \approx \max(0, \Theta)$, and the matrix $\max(0, \Theta)$ represents the similarities between points in the lower-dimensional space. Finally, the embedded points are recovered from the low-rank approximation Θ; see [27] for more details.

3. LS-RMD versus Latent-RMD. In this section, we explore the link between LS-RMD in (1.1) and Latent-RMD in (1.2). In particular, we aim to determine if two optimal low-rank approximations provided by the two models coincide. First, when X contains negative values, Latent-RMD is infeasible since there exists no latent variable Z such that $X = \max(0, Z)$. In contrast, LS-RMD still admits feasible solutions. Of course, the RMD is more meaningful when X is nonnegative, which is the case we focus on in the following.

In the exact case, if there exists a low-rank matrix Θ such that $X = \max(0, \Theta)$, LS-RMD and Latent-RMD are equivalent: for any low-rank solution Θ^* of (1.1), there exists a latent matrix Z^* such that (Z^*, Θ^*) is a solution of (1.2) and vice versa. Indeed, in this case, there always exists a trivial latent solution $Z^* = \Theta^*$. We show in what follows that this is not always true if an exact decomposition does not exist.

We demonstrate first that there exists a specific matrix X such that the rank-1 Latent-RMD is ill-posed since the infimum of (1.2) is not attained (see Example 1). Moreover, by showing that for the same matrix X, LS-RMD admits a nonempty set of optimal solutions (Example 2), we establish that the two formulations do not necessarily yield the same optimal approximations. Note that we could not establish the general well-posedness of LS-RMD, which is a direction of future research.

Example 1. Let $0 < \epsilon < \frac{1}{\sqrt{2}}$ be a fixed parameter, $r = 1$, and

$$X = \begin{pmatrix} 1 & 0 \\ \epsilon & 1 \end{pmatrix}. \tag{3.1}$$

Then the infimum of Latent-RMD in (1.2) is not attained.

Proof. Let us prove the result by contradiction. Assume there exists a solution to (1.2), denoted $(\widehat{Z}, \widehat{\Theta})$. It is straightforward to notice that, independently of $\widehat{Z}$, the optimal $\widehat{\Theta}$ must be a best rank-1 approximation of $\widehat{Z}$. Let us denote the ith singular value of $\widehat{Z}$ as $\sigma_i(Z)$, and let $\widehat{Z}_1$ be a best rank-one approximation of $\widehat{Z}$. By the Eckart–Young theorem, $\|\widehat{Z} - \widehat{Z}_1\|_F^2 = \sigma_2(Z)^2$ since $\widehat{Z}$ is a 2-by-2 matrix. Hence $\widehat{Z}$ must belong to

$$\operatorname*{argmin}_Z \sigma_2(Z)^2 \quad \text{such that} \quad \max(0, Z) = X. \tag{3.2}$$

Equivalently, we want to find $b \leq 0$ such that the matrix $Z = \left(\begin{smallmatrix} 1 & b \\ \epsilon & 1 \end{smallmatrix}\right)$ has the lowest possible second singular value. Since Z is a 2×2 matrix, we can compute the eigenvalues of ZZ^T by solving $\det(ZZ^T - \sigma I) = 0$ and get an explicit expression for the second singular value of Z. Solving the equation allows us to reformulate (3.2) as

$$\operatorname*{argmin}_{b<0} \ell(b) := \frac{2 + b^2 + \epsilon^2 - \sqrt{(b^2 - \epsilon^2)^2 + 4(b + \epsilon)^2}}{2}. \tag{3.3}$$

One can verify that $\lim_{b \to -\infty} \ell(b) = \epsilon^2$, while $\ell(b) > \epsilon^2$ for every $\epsilon \leq 1/\sqrt{2}$ and $b < 0$. This means the infimum of $\ell(b)$ is not attained by any finite value of b, and hence $\widehat{Z}$ does not exist. ∎

We now demonstrate that rank-1 LS-RMD admits an optimal solution for X as in (3.1).

Example 2. Let X as in (3.1) and $r = 1$; then for every $0 < \epsilon < \frac{1}{\sqrt{2}}$, the infimum of LS-RMD is attained by all the matrices of the form

$$\Theta = \begin{pmatrix} 1 & -v \\ -\frac{1}{v} & 1 \end{pmatrix}, \quad v \in \mathbb{R}_+, \tag{3.4}$$

with optimal value equal to ϵ^2.

Proof. Let us denote the error function of LS-RMD as $F(\Theta) = \|X - \max(0, \Theta)\|_F^2$. Note that 2×2 rank-1 matrices have limited patterns that the signs of their entries must follow. For instance, if the matrix does not contain zeros, the number of positive and negative entries must be even; otherwise, the rank cannot be equal to one. We proceed by analyzing three different cases.

Case 1: The rank-1 matrix Θ contains one zero column or row. In this case, the residual $F(\Theta)$ cannot be less than 1 since the nonlinear approximation $\max(0, \Theta)$ has at least one zero in the diagonal.

Case 2: The rank-1 matrix Θ has no zero rows or columns, and it contains two negative entries. If there is a negative entry on the diagonal, then the error is larger than one. Otherwise, the positive entries on the diagonal can be set to 1 to minimize the error, which is exactly the matrix in (3.4), with error ϵ^2.

Case 3: The rank-1 matrix Θ contains only positive entries. If $\Theta \geq 0$, the nonlinear activation has no effect on Θ and the RMD corresponds to minimizing the function $\|X-\Theta\|_F^2$. Let $\widehat{\sigma}_2$ be the second singular value of X as in (3.1). If $\operatorname{rank}(\Theta) = 1$, then $\|X-\Theta\|_F^2 \geq \widehat{\sigma}_2^2$, and using (3.3), we have $\widehat{\sigma}_2^2 = \frac{2+\epsilon^2-\epsilon\sqrt{\epsilon^2+4}}{2}$. Simple calculations show that $\widehat{\sigma}_2^2$ is strictly larger than ϵ^2 for any $0 < \epsilon < \frac{1}{\sqrt{2}}$. ∎

As a consequence of Examples 1 and 2, we have the following corollary.

Corollary 3.1. *LS-RMD and Latent-RMD may not share the same set of optimal low-rank solutions.*

We established the nonequivalence between the low-rank solutions of LS-RMD and Latent-RMD. However, in practice, we are interested in computing accurate solutions of LS-RMD, even though they might not be optimal. Numerous numerical results show that the decompositions obtained by solving the Latent-RMD model achieve a small residual error also for LS-RMD [28, 29, 33]. The following theorem formalizes this observation, showing that any feasible point $(\bar{\Theta}, \bar{Z})$ of the latent model has a residual value in the original LS-RMD that is upper bounded by four times the residual of Latent-RMD.

Theorem 3.2. *Let $X \in \mathbb{R}^{n\times m}$ be a nonnegative matrix and denote as $(\bar{\Theta}, \bar{Z})$ a feasible point of the Latent-RMD model in (1.2); then*

$$\|X - \max(0, \bar{\Theta})\|_F^2 \leq 4\|\bar{Z} - \bar{\Theta}\|_F^2. \tag{3.5}$$

Proof. Using the matrix inequality $\|A+B\|_F^2 \leq 2\|A\|_F^2 + 2\|B\|_F^2$, we have

$$\begin{aligned} \|X - \max(0, \bar{\Theta})\|_F^2 &= \|X - \bar{Z} + \bar{Z} + \min(0, \bar{\Theta}) - \min(0, \bar{\Theta}) - \max(0, \bar{\Theta})\|_F^2 \\ &\leq 2\|X - \bar{Z} + \min(0, \bar{\Theta})\|_F^2 + 2\|\bar{Z} - \bar{\Theta}\|_F^2. \end{aligned} \tag{3.6}$$

Let us define $\Omega = \{(i,j) \mid X_{i,j} > 0\}$ and $\widehat{\Omega} = \{(i,j) \mid \bar{\Theta}_{i,j} > 0\}$, and Ω^C and $\widehat{\Omega}^C$ their respective complements. By assumption, $\bar{Z}$ is a feasible solution of (1.2); hence $\max(0, \bar{Z}) = X$. Therefore,

$$\begin{aligned} \|\bar{Z} - \bar{\Theta}\|_F^2 = &\sum_{(i,j)\in\Omega\cap\widehat{\Omega}} (X_{ij} - \bar{\Theta}_{ij})^2 + \sum_{(i,j)\in\Omega\cap\widehat{\Omega}^C} (X_{ij} - \bar{\Theta}_{ij})^2 \\ &+ \sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}} (\bar{Z}_{ij} - \bar{\Theta}_{ij})^2 + \sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}^C} (\bar{Z}_{ij} - \bar{\Theta}_{ij})^2. \end{aligned}$$

Similarly, $\bar{Z}_{i,j} \leq 0$ whenever $X_{i,j} = 0$; thus,

$$\begin{aligned} \|X - \bar{Z} + \min(0, \bar{\Theta})\|_F^2 = &\sum_{(i,j)\in\Omega\cap\widehat{\Omega}^C} \bar{\Theta}_{ij}^2 + \sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}} \bar{Z}_{ij}^2 + \sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}^C} (\bar{Z}_{ij} - \bar{\Theta}_{ij})^2 \\ \leq &\sum_{(i,j)\in\Omega\cap\widehat{\Omega}^C} (X_{ij} - \bar{\Theta}_{ij})^2 + \sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}} (\bar{Z}_{ij} - \bar{\Theta}_{ij})^2 + \sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}^C} (\bar{Z}_{ij} - \bar{\Theta}_{ij})^2 \\ \leq &\|\bar{Z} - \bar{\Theta}\|_F^2, \end{aligned} \tag{3.7}$$

where the first inequality holds because

$$\bar{\Theta}_{ij}^2 \leq (X_{ij} - \bar{\Theta}_{ij})^2 \text{ for } (i,j) \in \Omega \cap \widehat{\Omega}^C, \quad \text{and} \quad \bar{Z}_{ij}^2 \leq (\bar{Z}_{ij} - \bar{\Theta}_{ij})^2 \text{ for } (i,j) \in \Omega^C \cap \widehat{\Omega}.$$

Combining (3.6) and (3.7), we get (3.5). ∎

A straightforward consequence of Theorem 3.2 is that a solution of Latent-RMD with a small error provides an LS-RMD solution with a small error, even though the two models do not necessarily yield the same approximation. We specify that the same results hold if we reparametrize Θ as WH; thus, the same bound holds for 3B-RMD as well.

4. Algorithms for RMD.

This section is dedicated to the algorithms for computing RMDs and to the study of their convergence. In section 4.1, we briefly review the state-of-the-art algorithms for solving Latent-RMD and 3B-RMD. In section 4.2, we discuss the convergence of the BCD scheme. Finally, we present our new algorithm eBCD along with its convergence in section 4.3.

4.1. Previous works.

Let us start with the algorithm designed for Latent-RMD since we use them as baselines in the numerical comparison in section 5. To our knowledge, all existing algorithms are alternating optimization methods in which each block of variables is updated sequentially [17, 18, 25, 26, 32] and lack convergence guarantees.

The first approach is the so-called Naive algorithm, introduced by Saul in [28], which employs a basic alternating scheme over Z and Θ with closed-form solutions for both. Similar approaches have been studied in matrix completion; see, for example, [6, 9, 11, 16]. At each iteration, first Θ is fixed and then the optimal Z is computed solving the subproblem

$$\operatorname*{argmin}_{Z} \|Z-\Theta\|_F^2 \quad \text{such that} \quad \max(0,Z)=X. \tag{4.1}$$

The solution of (4.1) is given by the projection of Θ over the set $\{Z \,:\, \max(0,Z)=X\}$, that is,

$$Z_{ij}=\begin{cases} X_{ij} & \text{if } X_{ij}>0,\\ \min(0,\Theta_{ij}) & \text{if } X_{ij}=0. \end{cases}$$

Once Z is updated, Θ is updated solving $\operatorname{argmin}_{\Theta}\|Z-\Theta\|_F^2$ such that $\operatorname{rank}(\Theta)\le r$, whose optimal solution(s) can be obtained with the TSVD of Z. Despite its simplicity, this scheme is highly reliable in practice, with the most costly operation at each iteration being the computation of a TSVD.

An accelerated, heuristic version of the Naive scheme was proposed in [29]. Specifically, the so-called Aggressive Naive (A-Naive) algorithm employs an extrapolation step for both Z and Θ, while the extrapolation parameter changes adaptively at each iteration, depending on the value of the residual. A-Naive is an example where, despite its lack of convexity and theoretical guarantees, extrapolation can be considerably faster than the nonaccelerated counterpart [2, 35].

Another algorithm for solving Latent-RMD is the expectation-minimization (EM) algorithm [28]. This approach is based on a Gaussian latent variable model parametrized by the low-rank matrix Θ and a parameter σ. For each entry of the original matrix, a Gaussian latent variable $Z_{ij}=\mathcal{N}(\Theta_{ij},\sigma^2)$ is sampled. The method estimates Θ and σ by maximizing the likelihood of the data with respect to Θ and σ. The optimization is carried out using the EM scheme that consists of two main steps: the E-step which computes the posterior mean and variance of the Gaussian latent variable model, and the M-step which re-estimates the parameters of the model from the posterior mean and variance.

Algorithm 4.1. BCD: block coordinate descent for 3B-RMD.

Input: X, W^0, H^0, maxit.
Output: low-rank factors W^k and H^k.
1: Set $\Omega = \{(i,j) \mid X_{ij} > 0\}$
2: **for** $k = 0, 1, \ldots,$ maxit **do**
3: $\quad Z^{k+1} = P_\Omega(X) + P_{\Omega^C}(\min(0, W^k H^k))$
4: $\quad W^{k+1} = Z^{k+1}(H^k)^\dagger$
5: $\quad H^{k+1} = ((W^{k+1})^T)^\dagger Z^{k+1}$
6: **end for**

Finally, we present the BCD algorithm for solving the 3B-RMD model. In what follows, we use the formulation of 3B-RMD introduced in (2.1) rather than the equivalent one in (1.3). The BCD algorithm consists of an alternating optimization procedure over three blocks of variables (Z, W, H) and computes a global optimizer for each of the subproblems sequentially. Let $A^\dagger$ denote the Moore–Penrose inverse of A, and let (H^k, Z^k, W^k) be the current iteration. The $(k+1)$th step of BCD is as follows:

(4.2)
$$\begin{aligned} Z^{k+1} &= \operatorname*{argmin}_Z \{\|Z - W^k H^k\|_F^2 \quad \text{s.t.} \quad \max(0, Z) = X\} = P_\Omega(X) + P_{\Omega^C}(\min(0, W^k H^k)), \\ W^{k+1} &= \operatorname*{argmin}_W \|Z^{k+1} - W H^k\|_F^2 = Z^{k+1}(H^k)^\dagger, \\ H^{k+1} &= \operatorname*{argmin}_H \|Z^{k+1} - W^{k+1} H\|_F^2 = ((W^{k+1})^T)^\dagger Z^{k+1}, \end{aligned}$$

where P_Ω and P_Ω^C are defined as in (2.1). The solutions of the subproblems for $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$ are analogous. In particular, the first is separable by rows of W, leading to m least squares problems in r variables with data vector of dimension n, while the subproblem for H is separable by column and equivalent to n least squares problems of r variables with data vector of dimension m. Both these subproblems do not necessarily have unique solutions. In fact, uniqueness is guaranteed only if W^k and H^k remain full-rank throughout the iterations. Alternatively, uniqueness is ensured by adding a Tikhonov regularization term or a proximal term in each of the subproblems [3, 17, 18]. However, regularization might slow down the algorithm, and it is not necessary to prove the convergence of the BCD scheme in this context.

On the contrary, the subproblem in the latent variable Z admits a unique global minimum, which is the projection of the product $W^k H^k$ onto the feasible, convex set

$$\{Z \in \mathbb{R}^{m \times n} \ : \ P_\Omega(Z) = P_\Omega(X), \ P_{\Omega^C}(Z) \leq 0\}. \tag{4.3}$$

The overall BCD algorithm is reported in Algorithm 4.1 and has a computational cost of $O(r|\Omega^C| + (m+n)r^2)$.

Similarly to the Naive scheme for Latent-RMD, adding extrapolation might be beneficial also for BCD. In particular, the work [29] proposes to perform an extrapolation step on the latent variable Z and on the low-rank product WH, resulting in the so-called extrapolated

3B (e3B) algorithm. Moreover, Wang et al. [33, 34] added a Tikhonov regularization for both the variables W and H. Their algorithm is based on the e3B scheme but performs two additional extrapolation steps following the computation of W and H. Even though numerical experience shows that both accelerated algorithms outperform the original BCD, convergence still remains an open problem.

4.2. Convergence of BCD. In this section, we discuss the asymptotic convergence to a stationary point of the BCD algorithm (Algorithm 4.1), which, to the best of our knowledge, has not been studied before in the context of RMD. Indeed, our goal is to prove that BCD is convergent under the assumption that a limit point of the sequence generated by the scheme exists. We proceed by showing that the method falls into the more general framework of exact BCD, where one of the subproblems is strictly quasi-convex. Specifically, the general BCD scheme is studied by Grippo and Sciandrone [17] when the objective function is continuously differentiable over the Cartesian product of m closed, convex sets, one for each block of variables. We first observe that the 3B-RMD objective function is continuously differentiable and defined over three blocks of variables Z, W, and H. The latent variable Z is the only constrained variable, and it is easy to show that the constraint set in (4.3) is convex. Therefore, 3B-RMD is a particular instance of the more general problem considered in [17].

Theorem 4.1. *Any limit point of the sequence generated by BCD (Algorithm 4.1) is a stationary point of problem* (2.1).

Proof. The global convergence of the BCD scheme for a general function with m blocks of variables is analyzed under generalized convexity assumptions in [17, Proposition 5]. The key hypotheses are (1) the existence of a limit point of the sequence generated by the BCD algorithm, and (2) strict quasi-convexity of at least $m-2$ subproblems. Therefore, since 3B-RMD has three blocks of variables, we just need one subproblem to be strictly quasi-convex. The first subproblem in the variable Z is a convex, quadratic problem over a convex set, making it strictly quasi-convex, implying the convergence to a stationary point. ∎

The assumption of the existence of the limit point is an undesirable hypothesis. However, it is necessary because the level sets of the objective function are not bounded. Therefore, similarly to Latent-RMD, 3B-RMD might have instances in which the minimum is not attained (see Theorem 2). This means that, in pathological cases, the sequence generated by BCD may diverge simply because the infimum of the function is not attained by any finite value. Thus, the hypothesis regarding the existence of a limit point of the BCD sequence cannot be weakened. One possible way to overcome this limitation is by adding a regularization term to the objective function (2.1), which makes the level sets compact, as in [33]. However, this is not the purpose of this work, which focuses on the nonregularized formulation.

4.3. Extrapolated BCD. We want to accelerate the BCD algorithm while preserving convergence to a stationary point under mild assumptions. The eBCD method is an adaptation of the well-known LMaFit algorithm [35] for matrix completion. The standard matrix completion formulation in [35] is similar to the one we consider in (2.1), although it does not include the inequality constraint $P_{\Omega^C}(Z) \leq 0$, which models the crucial information that the missing entries are negative. Consequently, we derive a new optimization scheme, differing from the one in [35], as the update of Z presents an additional nonlinear term; see (4.4).

Moreover, we relax the hypothesis in [35] that requires $\{W^k\}$ and $\{H^k\}$ to remain full-rank for all iterations. We present two different versions of the eBCD algorithm that, starting from the same point, produce the same low-rank approximation W^kH^k, yet the factors W^k and H^k vary. In what follows, we omit the iteration index, k, to simplify the notation, as we focus on a single iteration of the eBCD algorithm. We assume we have a point (Z, W, H) from which we perform one iteration.

4.3.1. First version of eBCD. Let $\alpha \geq 1$; then the first version of the eBCD scheme uses the following update: given the iterate (Z, W, H), it computes

$$\begin{aligned} Z_\alpha &= \alpha Z + (1-\alpha)WH, \\ \widehat{W}(\alpha) &= Z_\alpha H^\dagger, \\ \widehat{H}(\alpha) &= (\widehat{W}(\alpha)^T)^\dagger Z_\alpha, \\ \widehat{Z}(\alpha) &= P_\Omega(X) + P_{\Omega^C}(\min(0, \widehat{W}(\alpha)\widehat{H}(\alpha))), \end{aligned} \tag{4.4}$$

where $(\widehat{Z}(\alpha), \widehat{W}(\alpha), \widehat{H}(\alpha))$ provides the next iterate. Similarly to BCD, the main computational cost for the scheme in (4.4) is solving the two least squares problems for computing $\widehat{W}(\alpha)$ and $\widehat{H}(\alpha)$. The optimization scheme in (4.4) differs from the one in (4.2) because of the new variable Z_α which is used in place of the original latent variable Z to update W and H. Moreover, if we assume the matrix H to be full-rank, we can derive an intuitive interpretation of the update of Z_α as an *indirect extrapolation step.* Since H is full-rank we have $H^\dagger = H^T(HH^T)^{-1}$ and thus

$$\widehat{W}(\alpha) = Z_\alpha H^T(HH^T)^{-1} = \alpha Z H^T(HH^T)^{-1} + (1-\alpha)WHH^T(HH^T)^{-1} = \alpha W_{\text{BCD}} + (1-\alpha)W,$$

where W_{BCD} is the update of the variable W that we would have had if we performed one step of the BCD scheme (4.2). Moreover, if we define $\beta = \alpha - 1$, then

$$\widehat{W}(\alpha) = \alpha W_{\text{BCD}} + (1-\alpha)W = (1+\beta)W_{\text{BCD}} - \beta W = W_{\text{BCD}} + \beta(W_{\text{BCD}} - W). \tag{4.5}$$

Thus, line (4.5) represents an extrapolation step for W_{BCD}, provided that $\alpha \in (1, 2)$, that means $\beta \in (0, 1)$. Unfortunately, such an intuitive interpretation does not apply to the update of the variable H. Note that the step in (4.5) is properly referred to as an extrapolation step if β is in the interval (0,1) or, equivalently, α is in the interval (1,2), but our scheme potentially allows for larger values that exceed the upper bound.

4.3.2. Improved version of eBCD. The second version of the eBCD scheme does not require solving any least squares problem and the main computational cost is associated with a single QR factorization of a matrix in $\mathbb{R}^{m\times r}$. We specify that this scheme can be adapted to design alternative versions of BCD and e3B as well. We state the following lemma from [35] that is used in what follows.

Lemma 4.2. *Let $\widehat{W}(\alpha)$ be the update of W for the scheme in (4.4); then*

$$\mathcal{R}(\widehat{W}(\alpha)) = \mathcal{R}(Z_\alpha H^T),$$

where $\mathcal{R}(A)$ denotes the range of matrix A.

Proof. Let $UDV^T = H$ be the economy form SVD of H; then $\widehat{W}(\alpha) = Z_\alpha H^\dagger = Z_\alpha V D^\dagger U^T$. Moreover, since $Z_\alpha H^T = Z_\alpha V D U^T$, we get $\mathcal{R}(\widehat{W}(\alpha)) = \mathcal{R}(Z_\alpha H^T)$. ∎

Lemma 4.2 shows that the range space of $\widehat{W}(\alpha)$ can be obtained from that of $Z_\alpha H^T$. Let $Q(\alpha)$ be a matrix whose columns form an orthonormal basis for $\mathcal{R}(Z_\alpha H^T)$. Then, using Lemma 4.2 and from the second and third lines in (4.4), we get

$$\widehat{W}(\alpha)\widehat{H}(\alpha) = \widehat{W}(\alpha)\widehat{W}(\alpha)^\dagger Z_\alpha = Q(\alpha)Q(\alpha)^T Z_\alpha. \tag{4.6}$$

This shows that the low-rank approximation produced by the eBCD is nothing other than the orthogonal projection of Z_α into $\mathcal{R}(Z_\alpha H^T)$. Justified by the relation in (4.6), we now state the improved version of eBCD. Given an iterate (Z, W, H), we compute

$$\begin{aligned} &Z_\alpha = \alpha Z + (1-\alpha)WH, \\ &W(\alpha) = Q(\alpha) \leftarrow \text{Orthonormal basis of } \mathcal{R}(Z_\alpha H^T), \\ &H(\alpha) = W(\alpha)^T Z_\alpha = Q(\alpha)^T Z_\alpha, \\ &Z(\alpha) = P_\Omega(X) + P_{\Omega^c}(\min(0, W(\alpha)H(\alpha))) \end{aligned} \tag{4.7}$$

to obtain the next iterate $(Z(\alpha), W(\alpha), H(\alpha))$. Equation (4.6) guarantees that the low-rank approximations computed by the scheme in (4.4) and in (4.7) are equal, that is, $\widehat{W}(\alpha)\widehat{H}(\alpha) = W(\alpha)H(\alpha)$. In practice, $Q(\alpha)$ can be computed from the economy QR decomposition of $Z_\alpha H^T$. Note that if $\operatorname{rank}(Z_\alpha H^T) = r_1 < r$, the first r_1 columns of the Q matrix from the economy QR decomposition of $Z_\alpha H^T$ are not, in general, an orthonormal basis of the range space of the matrix. To ensure this property, a modified version of the QR decomposition using column pivoting must be used instead. Specifically, there exists $Q(\alpha) \in \mathbb{R}^{m \times r_1}$, containing an orthonormal basis of $Z_\alpha H^T$, such that $Z_\alpha H^T \Pi = Q(\alpha)R(\alpha)$, where Π is a permutation matrix and $R(\alpha) \in \mathbb{R}^{r_1 \times r}$ [15]. Therefore, we potentially allow for a drop in the rank of the approximation whenever $Z_\alpha H^T$ is rank-deficient. However, we have never encountered this case in practical situations because $Z_\alpha H^T$ is likely to have rank at least r while the factors $Q(\alpha)$ and $H(\alpha)$ have rank exactly r.

We proceed by defining the residual of our algorithm, denoted as $S(\alpha)$,

$$S(\alpha) = Z(\alpha) - W(\alpha)H(\alpha) = P_\Omega(X - W(\alpha)H(\alpha)) - P_{\Omega^c}(\max(0, W(\alpha)H(\alpha))). \tag{4.8}$$

We highlight an interesting relation between the variable Z_α and the residual. Let S denote the residual matrix of the iterate (Z, W, H), defined as in (4.8) with Z, W, and H instead of $Z(\alpha)$, $W(\alpha)$, and $H(\alpha)$ respectively; then

$$\begin{aligned} Z_\alpha &= \alpha Z + (1-\alpha)WH = \alpha(P_\Omega(X) + P_{\Omega^c}(\min(0, WH)) - WH) + WH \\ &= \alpha(P_\Omega(X - WH) - P_{\Omega^c}(\max(0, WH))) + WH \\ &= WH + \alpha S. \end{aligned} \tag{4.9}$$

Equation (4.9) shows that Z_α is indeed the low-rank approximation WH at step k, to which we add a multiple of the residual S. This suggests a second interpretation of the update of Z_α as a residual corrected step of the low-rank approximation WH.

Restarting scheme of the extrapolation parameter. We need to determine a rule to select the extrapolation parameter α at each iteration. Extrapolating a variable means adding information from the past iterate; thus, the extrapolation parameter α affects the relevance of the past information in the current update. Of course, not all the values of α are acceptable to progress in minimizing the residual, and a good choice of the parameter is crucial. The updating scheme of the extrapolation parameter α is inspired by the one introduced in [35]. The main idea is to choose the parameter α such that the residual of the function is sufficiently decreased between two consecutive iterations.

At each step, we start from a candidate value for α, and evaluate the ratio $\delta(\alpha)$ between the norms of the residual $S(\alpha)$ at iteration $k+1$ and S at iteration k, defined as

$$\delta(\alpha) = \frac{\|S(\alpha)\|_F}{\|S\|_F}. \tag{4.10}$$

If $\delta(\alpha) < 1$, we declare our update successful, and the new iterate is set to $(W^{k+1}, H^{k+1}, Z^{k+1}) = (W(\alpha), H(\alpha), Z(\alpha))$. Furthermore, let $\bar{\delta}$ be a fixed threshold; if $\delta(\alpha) > \bar{\delta} \in (0,1)$, meaning we do not consider the decrease of the residual fast enough, we increase α by setting $\alpha \leftarrow \min(\alpha+\mu, \bar{\alpha})$, where $\mu = \max(\mu, 0.25(\alpha-1))$. If α reaches the upper bound $\bar{\alpha}$, we restart the parameter by setting $\alpha = 1$. This correction does not appear in [35], but we added it to guarantee the convergence of our algorithm. On the contrary, if $\delta(\alpha) \geq 1$, the step is declared unsuccessful, we do not accept the update, and we compute a standard BCD update instead by setting $\alpha = 1$. Consequently, either we find an $\alpha \in (1, \bar{\alpha})$ that allows a decrease of the residual, or we recover one iteration of the BCD, which is guaranteed to decrease the residual. Therefore, the new iterate of the algorithm produces a lower value in the residual with respect to the previous iteration.

The overall eBCD scheme is described in Algorithm 4.2, where we consider the updating scheme in (4.7). We choose the improved version of eBCD because (a) it exhibits faster performance in practice; (b) computing one QR factorization is, in general, more stable than solving two consecutive least squares problems; (c) one of the factors in the decomposition is orthogonal and this helps demonstrate crucial properties in the convergence analysis, such as boundedness of the sequence generated by the eBCD algorithm. Let us also mention that, in [35], convergence is proved for a scheme similar to the one in (4.4), instead of the scheme in (4.7) as done in this paper.

4.3.3. Convergence analysis. We present now the convergence analysis of Algorithm 4.2. The proofs of the lemmas can be found in the appendix. The main steps of the convergence analysis are summarized as follows.

Step 1 We first prove that there exists a range of values larger than 1 for α such that the scheme (4.7) decreases the residual, that is, $\|S\|_F^2 - \|S(\alpha)\|_F^2 > 0$. Specifically, we split this residual reduction into the contribution given by the update of each variable separately: Lemma 4.3 characterizes the residual reduction after the update of W and H, showing that it is nonnegative. Lemmas 4.4 and 4.5 describe the residual reduction after updating the entries of Z.

Step 2 We provide the KKT conditions for problem (2.1); see (4.17).

Algorithm 4.2. eBCD for 3B-RMD.

Input: X, Z^0,W^0, H^0, $\bar{\alpha}$ (default $=4$), μ (default $=0.3$), $\bar{\delta}$ (default $=0.8$), maxit

Set $\Omega = \{(i,j) \mid X_{ij} > 0\}$ and $\alpha_0 = 1$.
for $k = 0, \ldots,$ maxit **do**
  $Z_{\alpha_k} = \alpha_k Z^k + (1-\alpha_k) W^k H^k$
  Compute $Q(\alpha_k)$ orthonormal basis of $\mathcal{R}(Z_{\alpha_k}(H^k)^T)$
  $W(\alpha_k) = Q(\alpha_k)$
  $H(\alpha_k) = Q(\alpha_k)^T Z_{\alpha_k}$
  $Z(\alpha_k) = P_\Omega(X) + P_{\Omega^c}(\min(0, W(\alpha_k)H(\alpha_k)))$
  if $\delta(\alpha_k) \geq 1$ **then**
    Set α_k to 1
    $(W^{k+1}, H^{k+1}, Z^{k+1}) = (W^k, H^k, Z^k)$
  else
    $(W^{k+1}, H^{k+1}, Z^{k+1}) = (W(\alpha_k), H(\alpha_k), Z(\alpha_k))$
    if $\delta(\alpha_k) \geq \bar{\delta}$ **then**
      $\mu = \max(\mu, 0.25(\alpha_k - 1))$ and $\alpha_{k+1} = \min(\alpha_k + \mu, \bar{\alpha})$
      if $\alpha_{k+1} = \bar{\alpha}$ **then**
        $\alpha_{k+1} = 1$,
      end if
    else
      $\alpha_{k+1} = \alpha_k$.
    end if
  end if
end for

Step 3 We use the residual reduction that we characterized in Step 1 to prove the subsequence convergence of the eBCD algorithm in Theorem 4.7, showing that a subsequence of the eBCD iterates satisfies the KKT conditions at the limit. This is done by analyzing all possible situations that the restarting scheme arises.

Step 1. Let us denote $\sigma_{WH}(\alpha)$ the variation of the residual after the update of W and H and $\sigma_Z(\alpha)$ the variations after updating the entries of Z, that is,

$$\|S\|_F^2 - \|S(\alpha)\|_F^2 = \underbrace{\|S\|_F^2 - \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - Z_\alpha\|_F^2}_{=:\sigma_{WH}(\alpha)} + \underbrace{\frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - Z_\alpha\|_F^2 - \|S(\alpha)\|_F^2}_{=:\sigma_Z(\alpha)}.$$

Let U be an orthonormal basis of $\mathcal{R}(H^T)$. If H is full rank, we can define the orthogonal projections onto $\mathcal{R}(W(\alpha))$ and $\mathcal{R}(H^T)$ as

$$E(\alpha) = W(\alpha)W(\alpha)^T = Q(\alpha)Q(\alpha)^T, \qquad P = UU^T = H^T(HH^T)^{-1}H. \tag{4.11}$$

The following three lemmas characterize $\sigma_{WH}(\alpha)$ and $\sigma_Z(\alpha)$ (the proofs can be found in the appendix).

Lemma 4.3. *Given (Z, W, H) where H is full rank, let $(W(\alpha), H(\alpha))$ be computed by the scheme (4.7); then for any $\alpha \geq 1$,*

$$\sigma_{WH}(\alpha) = \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - WH\|_F^2 = \|SP\|_F^2 + \|E(\alpha)S(I-P)\|_F^2 \geq 0, \tag{4.12}$$

where $E(\alpha)$ and P are defined in (4.11).

Therefore, after the first two steps, the residual reduction is strictly positive unless $W(\alpha)H(\alpha) = WH$. We look at the residual reduction after updating the entries of Z in Ω^C (Lemma 4.4) and in Ω (Lemma 4.5). In the following lemmas, we assume that there exists an interval $[1, \widehat{\alpha}]$ such that $Z_\alpha H^T$ and ZH^T have the same rank. This hypothesis is needed to ensure the continuity of the residual reduction $\sigma_Z(\alpha)$ in a range of $\alpha > 1$; more details are given in the appendix.

Lemma 4.4. *Let $(W(\alpha), H(\alpha))$ be generated by the scheme in (4.7). Furthermore, assume there exists $\widehat{\alpha} > 1$ such that $\operatorname{rank}(Z_\alpha H^T) = \operatorname{rank}(ZH^T)$ for $\alpha \in [1, \widehat{\alpha}]$; then there exists $\alpha_* > 1$ such that for every $\alpha \in (1, \min(\alpha_*, \widehat{\alpha})]$,*

$$\frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_{\Omega^C}(S(\alpha))\|_F^2 \geq 0. \tag{4.13}$$

Moreover, if $P_{\Omega^C}(\min(0, W(1)H(1))) \neq P_{\Omega^C}((\min(0, WH))$, a strict inequality holds.

A similar result is proved in [35] for matrix completion. In their less restrictive setting, the residual reduction after updating the entries of Z in Ω^C is independent of α and nonnegative. However, due to the additional constraint $P_{\Omega^C}(Z) \leq 0$ in the 3B-RMD formulation, the residual reduction in (4.13) of the eBCD algorithm is only locally nonnegative in a range where α is larger than one.

Analogously to Lemma 4.4, we now state a similar result that characterizes the residual reduction when the entries of Z in Ω are updated.

Lemma 4.5. *Let $(W(\alpha), H(\alpha))$ be generated by the scheme in (4.7). Furthermore, assume there exists $\widehat{\alpha} > 1$ such that $\operatorname{rank}(Z_\alpha H^T) = \operatorname{rank}(ZH^T)$ for $\alpha \in [1, \widehat{\alpha}]$. Then*

$$\lim_{\alpha \to 1^+} \frac{1}{\alpha^2}\|P_\Omega(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_\Omega(S(\alpha))\|_F^2 = 0. \tag{4.14}$$

The following corollary gathers the findings from Lemmas 4.3, 4.4, and 4.5.

Corollary 4.6. *Let $(W(\alpha), H(\alpha), Z(\alpha))$ be generated by the scheme in (4.7). Furthermore, assume there exists $\widehat{\alpha} > 1$ such that $\operatorname{rank}(Z_\alpha H^T) = \operatorname{rank}(ZH^T)$ for $\alpha \in [1, \widehat{\alpha}]$. If $P_{\Omega^C}(\min(0, WH)) \neq P_{\Omega^C}(\min(0, W(1)H(1)))$, then there exists $\tilde{\alpha} > 1$ such that*

$$\|S\|_F^2 - \|S(\alpha)\|_F^2 = \sigma_{WH}(\alpha) + \sigma_Z(\alpha) > 0 \quad \forall \alpha \in [1, \min(\widehat{\alpha}, \tilde{\alpha})]. \tag{4.15}$$

Corollary 4.6 guarantees that there exist some values of the extrapolation parameter α strictly larger than 1 resulting in a positive residual reduction for eBCD (4.7).

Step 2. Before stating the actual convergence theorem, we need to introduce the optimality conditions or KKT conditions for problem (2.1). Let us define the Lagrangian function as

$$\mathcal{L}(W,H,Z,\Lambda,\Sigma)=\frac{1}{2}\|Z-WH\|_F^2+\langle\Lambda,P_\Omega(Z-X)\rangle+\langle\Sigma,P_{\Omega^C}(Z)\rangle, \tag{4.16}$$

where the dual variables are such that $\Lambda=P_\Omega(\Lambda)$ and $\Sigma=P_{\Omega^C}(\Sigma)$:

$$\begin{aligned}
\text{Stationarity condition: } & \nabla_W\mathcal{L}(Z,W,H)=-(Z-WH)H^T=0,\\
& \nabla_H\mathcal{L}(Z,W,H)=-W^T(Z-WH)=0,\\
& \nabla_Z\mathcal{L}(Z,W,H,\Lambda,\Sigma)=Z-WH+\Lambda+\Sigma=0,\\
\text{Primal feasibility: } & P_\Omega(Z-X)=0, P_{\Omega^C}(Z)\le 0,\\
\text{Dual feasibility: } & \Sigma\ge 0,\\
\text{Complementary slackness: } & \langle\Sigma,P_{\Omega^C}(Z)\rangle=0,
\end{aligned} \tag{4.17}$$

where the multiplier matrices Λ and Ω measure the residual in Ω and Ω^C, respectively, that is, $\Lambda=P_\Omega(WH-Z)$ and $\Sigma=P_{\Omega^C}(\max(0,WH))$. Our goal is to prove that, under some mild assumptions, a subsequence of the sequence generated by eBCD satisfies the KKT condition at the limit. Specifically, one can verify that the eBCD sequence satisfies the complementary slackness condition and the stationarity condition for Z at any point. Therefore, we just need to show that the optimality conditions for W and H are satisfied by a subsequence of the eBCD scheme at the limit, that is,

$$\lim_{k\to\infty}\nabla_W\mathcal{L}(Z^k,W^k,H^k)=0,\qquad \lim_{k\to\infty}\nabla_H\mathcal{L}(Z^k,W^k,H^k)=0. \tag{4.18}$$

Step 3. We are now ready to prove the convergence result for Algorithm 4.2.

Theorem 4.7. *Let $(W^{k+1},H^{k+1},Z^{k+1})$ be generated by eBCD (Algorithm 4.2), and assume that $\{P_{\Omega^C}(W^kH^k)\}$ is bounded for all k. Then there exists a bounded subsequence of $(W^{k+1},H^{k+1},Z^{k+1})$ whose limit point satisfies the KKT conditions (4.17).*

Proof. We first discuss the boundedness of the two sequences $\{W^{k+1}\}$ and $\{H^{k+1}\}$ produced by Algorithm 4.2. The boundedness of $\{P_{\Omega^C}(W^kH^k)\}$ implies that $\{Z^k\}$ and $\{W^kH^k\}$ are bounded. This also means that $\{Z_{\alpha_k}\}$ is a bounded sequence by construction. Moreover, the sequence $\{W^{k+1}\}$ contains orthogonal matrices whose columns form an orthonormal basis for $\mathcal{R}(Z_{\alpha_k}(H^k)^T)$, thus it is bounded. Finally, $H^{k+1}=(W^{k+1})^TZ_{\alpha_k}$ and since $\{W^{k+1}\}$ and $\{Z_{\alpha_k}\}$ are bounded, then also $\{H^{k+1}\}$ is bounded.

We observe that at every iteration H^{k+1} always has the same rank as W^{k+1}. This means that H^{k+1} is full rank for every k and Lemma 4.3 can be applied. In fact, it follows from Lemma 4.2 that $\mathcal{R}(Q^{k+1})=\mathcal{R}(Z_{\alpha_k}(H^k)^T)\subseteq\mathcal{R}(Z_{\alpha_k})$; that implies

$$\mathcal{R}(I)=\mathcal{R}((Q^{k+1})^TQ^{k+1})\subseteq\mathcal{R}((Q^{k+1})^TZ_{\alpha_k})=\mathcal{R}(H^{k+1}),$$

where I is the identity matrix of dimension equal to the number of columns of Q^{k+1} or, equivalently, to the rank of $Z_{\alpha_k}(H^k)^T$.

Next, we define the set of indices where the update of Z^k results in a positive residual reduction. Let $\mathcal{F} = \{k : \sigma_Z(\alpha_k) \geq 0\}$ and $\mathcal{F}^C$ its complement. Therefore, denoting $E^k = E(\alpha_k)$, since H^{k+1} is full rank, we can use Lemma 4.3, and summing over all $k \in \mathcal{F}$, we get

$$\|S^0\|_F^2 \geq \sum_{k\in\mathcal{F}} \sigma_{WH}(\alpha_k) = \sum_{k\in\mathcal{F}} \|S^k P^k\|_F^2 + \|E^k S^k (I - P^k)\|_F^2. \tag{4.19}$$

We analyze separately two different scenarios.

Case 1*:* Assume $|\mathcal{F}^C| < \infty$. It follows from (4.19) that

$$\lim_{k\to\infty, k\in\mathcal{F}} \|S^k P^k\|_F^2 = 0, \quad \lim_{k\to\infty, k\in\mathcal{F}} \|E^k S^k (I - P^k)\|_F^2 = 0, \tag{4.20}$$

which implies $\lim_{k\to\infty,k\in\mathcal{F}} \|E^k S^k\|_F^2 = 0$. Recall that $P_{\Omega^C}(Z^k) = P_{\Omega^C}(\min(0, W^k H^k))$ and $P_\Omega(X) = P_\Omega(Z^k)$. Using (4.11), we have $S^k P^k = (Z^k - W^k H^k)P^k = (Z^k - W^k H^k)U^k (U^k)^T$. Since U^k is an orthonormal basis for $\mathcal{R}((H^k)^T)$ and $\{H^k\}$ bounded,

$$\lim_{k\to\infty, k\in\mathcal{F}} (Z^k - W^k H^k)(H^k)^T = -\nabla_W \mathcal{L}(Z^k, W^k, H^k) = 0.$$

Moreover,

$$E^k S^k = E^k S^{k+1} + E^k (S^k - S^{k+1}) = W^{k+1}(W^{k+1})^T (Z^{k+1} - W^{k+1} H^{k+1}) + E^k (S^k - S^{k+1}). \tag{4.21}$$

Furthermore

$$\begin{aligned} \|S^k - S^{k+1}\|_F^2 &= \|P_\Omega(W^{k+1}H^{k+1} - W^k H^k)\|_F^2 + \|P_{\Omega^C}(\max(0, W^{k+1}H^{k+1}) \\ &\quad - \max(0, W^k H^k))\|_F^2 \\ &\leq \|W^{k+1}H^{k+1} - W^k H^k\|_F^2 + \|\max(0, W^{k+1}H^{k+1}) - \max(0, W^k H^k)\|_F^2 \\ &\leq 2\|W^{k+1}H^{k+1} - W^k H^k\|_F^2. \end{aligned} \tag{4.22}$$

Using (4.12) and the fact that $\alpha_k \leq \bar{\alpha}$ by construction, we have

$$\|W^{k+1}H^{k+1} - W^k H^k\|_F^2 \leq \bar{\alpha}^2 (\|S^k P^k\|_F^2 + \|E^k S^k (I - P^k)\|_F^2). \tag{4.23}$$

Combining (4.23) and (4.20) we get $\lim_{k\to\infty,k\in\mathcal{F}} \|W^{k+1}H^{k+1} - W^k H^k\|_F^2 = 0$; thus, from (4.22),

$$\lim_{k\to\infty, k\in\mathcal{F}} \|S^k - S^{k+1}\|_F^2 = 0. \tag{4.24}$$

Therefore, from (4.24) and (4.21), along with the boundedness of $\{W^k\}$, we conclude that

$$\lim_{k\to\infty, k\in\mathcal{F}} (W^{k+1})^T (Z^{k+1} - W^{k+1} H^{k+1}) = \lim_{k\to\infty, k\in\mathcal{F}} -\nabla_H \mathcal{L}(Z^{k+1}, W^{k+1}, H^{k+1}) = 0.$$

Case 2*:* $|\mathcal{F}^C| = \infty$. Let us analyze two subcases.

Case 2a*:* $|\{k \in \mathcal{F}^C \mid \delta(\alpha_k) > \bar{\delta}\}| < \infty$. For k sufficiently large, $\|S^{k+1}\|_F^2 \leq \bar{\delta}\|S^k\|_F^2$. Therefore, $\lim_{k\to\infty,k\in\mathcal{F}^C} \|S^k\|_F^2 = 0$, which means that the sequence converges to a global minimizer of the problem.

Case 2b*:* $|\{k \in \mathcal{F}^C \mid \delta(\alpha_k) > \bar{\delta}\}| = \infty$. eBCD sets the parameter α_k to 1 for an infinite number of iterations. Denoting $\mathcal{J}_1$ the set of indices in which $\alpha_k = 1$, a similar relation to (4.19) holds:

$$\|S^0\|_F^2 \geq \sum_{k \in \mathcal{J}_1} \sigma_{WH}(\alpha_k) = \sum_{k \in \mathcal{J}_1} \|S^k P^k\|_F^2 + \|E^k S^k (I - P^k)\|_F^2.$$

Therefore, one can replicate the proof for Case 1, and the subsequence whose indices are in $\mathcal{J}_1$ satisfies the optimality conditions at the limit. ∎

We emphasize that, similarly to BCD, we need a strong assumption, namely the boundedness of $\{P_{\Omega^C}(W^k H^k)\}$ which guarantees that the eBCD sequence remains bounded. This hypothesis serves the same purpose as the existence of the limit point in the BCD algorithm, since it allows us to exclude pathological cases where the infimum of the problem is not attained.

5. Numerical experiments. In this section, we present numerical experiments on synthetic and real-world data sets for different applications of RMD: matrix completion with ReLU sampling, the recovery of Euclidean distance matrices, low-dimensional embedding on text data, and the compression sparse dictionaries. Our main purpose is to show that eBCD behaves favorably compared to the state of the art, namely the Naive approach [28], EM [28], A-Naive [29], and e3B [29], while having convergence guarantees. In all experiments, we use random initialization: entries of W and H are sampled from a Gaussian distribution, `W = randn(m,r)` and `H = randn(r,n)` in MATLAB. We then scale them so that the initial point matches the magnitude of the original matrix, that is, we multiply W by $\sqrt{\|X\|_F}/\|W\|_F$ and H by $\sqrt{\|X\|_F}/\|H\|_F$. If not otherwise specified, we stop each algorithm when we reach a small relative residual, namely $\Gamma_k = \|Z^k - W^k H^k\|_F/\|X\|_F \leq 10^{-9}$, or a fixed time limit is reached. The time limit will be selected depending on the size of the data. For the algorithms solving Latent-RMD, the low-rank product $W^k H^k$ in Γ_k is substituted with the matrix Θ^k. The first stopping criteria can be achieved only if the target matrix admits an (almost) exact RMD. The algorithms are implemented in MATLAB R2021b on a 64-bit Samsung/Galaxy with 11th Gen Intel Core i5-1135G7 @ 2.40 GHz and 8 GB of RAM, under Windows 11 version 23H2. The code is available online from https://github.com/giovanniseraghiti/ReLU-NMD. The parameters of eBCD are the ones specified as default in Algorithm 4.2. For the parameters of the other methods, we use the setting from [29].

5.1. Matrix completion with ReLU sampling. In this subsection, we consider the same experiment used in [22] for symmetric matrix completion with ReLU sampling, adapted to the nonsymmetric case. Specifically, we randomly generate the entries W and H from a Gaussian distribution, as for our initializations, and we construct $\Theta^t = WH$, and set $X = \max(0, \Theta^t)$ in the noiseless case and $\max(0, \Theta^t + N)$ in the noisy case, where $N = \sigma * \tilde{N} * \|\Theta^t\|_F/\|\tilde{N}\|_F$, $\tilde{N} =$ `randn(m,n)`, and $\sigma \in (0,1)$. In our experiment, we set $m = n = 1000$, $r = 20$, and $\sigma = 10^{-2}$. Note that since W, H, and N are Gaussian, X is approximately 50% sparse. We present the loglog plot of the average residual Γ_k against the average CPU time over 20 different randomly generated matrices X in Figure 5.1. The algorithms are stopped when the objective functions in (1.2) and (1.3) achieve a relative residual Γ_k below 10^{-9} in the noiseless case, or below

Table 5.1

Average CPU time in seconds, standard deviation, and average iteration number to reach a tolerance on the relative residual Γ_k of 10^{-9} in the noiseless case, and of 10^{-2} in the noisy case. Best values are highlighted in bold.

		BCD	eBCD	e3B	Naive	A-Naive	EM
No Noise	time	7.92±0.16	**0.90±0.04**	1.95±0.04	22.0±0.78	6.37±0.33	26.7±0.70
	iter	304	121	**65**	308	84	296
With Noise	time	1.14±0.05	**0.19±0.01**	0.66±0.02	3.17±0.12	1.54±0.19	2.70±0.15
	iter	36	22	20	41	**19**	28

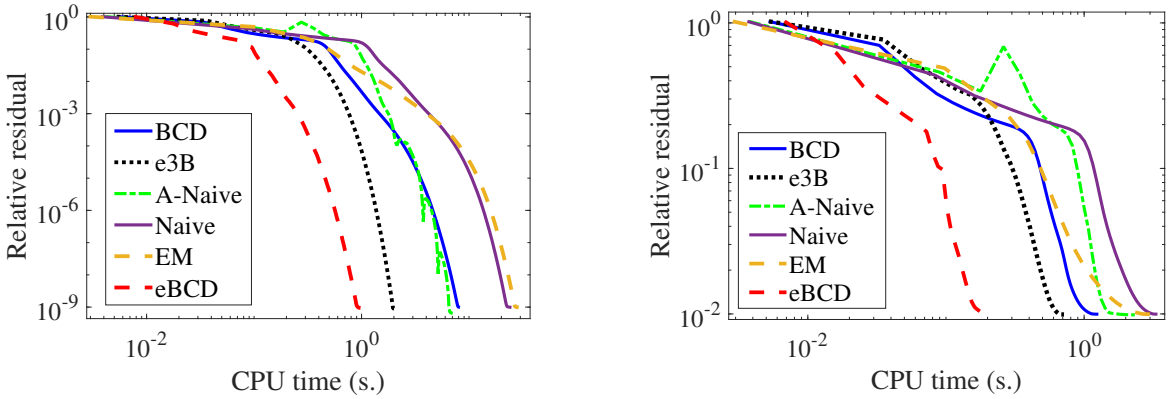

Figure 5.1. *Matrix completion with ReLU sampling: evolution of the average relative residual Γ_k w.r.t. CPU time for the noiseless case (left image) and the noisy case (right image).*

$\sigma = 10^{-2}$ in the noisy case. We also report the average CPU time and the standard deviation to reach these values in Table 5.1.

Figure 5.1 shows that all the algorithms reach the tolerance on the relative residual. Moreover, eBCD outperforms all the other methods, being considerably faster in both the noiseless and noisy cases, as Table 5.1 confirms. In fact, compared to the second best, eBCD is on average more than twice faster in the noiseless case, and more than three times faster in the noisy case (each time, the second best is e3B). Table 5.1 shows also that A-Naive and e3B need fewer iterations to converge than eBCD. However, their computational cost per iteration is significantly larger.

5.2. Euclidean distance matrix completion. We generate 200 points in two different ways in a three-dimensional space: (1) uniformly at random in $[0, 10]^3$, and (2) randomly clustered around six clusters (four with 30 points, and two with 40) where the centroid $(\bar{x}, \bar{y}, \bar{z})$ of the ith cluster is sampled uniformly at random in $[-10, 10]^3$, and we generate the points in MATLAB using $(\bar{x}, \bar{y}, \bar{z}) + \text{std} * \texttt{randn(1,3)}$, where std denotes the standard deviation that we fixed to std $= 3$. We compute the Euclidean distance matrix Θ^t containing the squared distance between all couples of points and we construct our target matrix $X = \max(0, dee^T - \Theta^t)$, where d represents the largest value in the matrix that we want to observe. The upper bound on the rank is fixed to $r = 5$, which is equal to the rank of Θ^t. In this experiment, we assume the threshold d to be known, but one might be interested in considering d as an unknown of the model. We aim to recover all the entries of Θ^t larger than d. In practice,

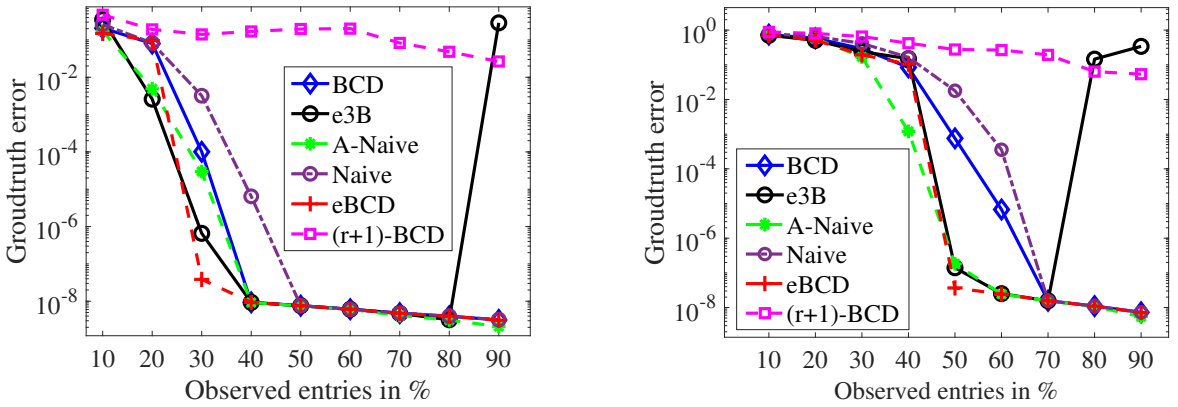

Figure 5.2. *Euclidean distance matrix completion: relative errors where the x-axis represents the percentage of known entries. The left figure is for the uniform distribution of the data points, the right figure for the clustered distribution.*

the threshold d is adapted to achieve the percentage of observed entries in the x-axis of the pictures on the right of Figure 5.2. Specifically, we display the average relative error of the low-rank approximation with the Euclidean distance matrix Θ^t for different percentages of observed entries, over 10 random experiments. Moreover, all algorithms have been adapted to include the rank-1 modification in (2.2). We excluded the EM algorithm from the analysis because adapting it to the rank-1 modified model appears less straightforward than for the other algorithms. We also consider as a baseline the rank-$(r+1)$ reconstruction given by the original BCD method solving the 3B-RMD of rank $r+1$, without considering any rank-1 modification ($(r+1)$-BCD). In this case, we compute a rank-$(r+1)$ decomposition of X and evaluate the relative error with $dee^T - \Theta^t$. We stop the algorithms when a tolerance of 10^{-9} on the relative residual Γ_k is reached or after 60 s.

The distribution of the points affects the recovery performance of the algorithms; see Figure 5.2. When the points are randomly distributed, 30% of the entries need to be observed for the eBCD to achieve an error below 10^{-7}. When the points are clustered, only 50% of the entries need to be observed for the eBCD to achieve an error below 10^{-7}.

Including the rank-1 modification in the model allows for a significantly more accurate reconstruction than solving a rank-$(r+1)$ 3B-RMD. In general, A-Naive, eBCD, and e3B provide the best reconstructions. However, e3B fails at least once in both examples when the percentage of observed entries is above 80%. This behavior might be due to the aggressive extrapolation procedure that e3B uses. We emphasize that neither the 3B-RMD nor the Latent-RMD formulation imposes any constraints to ensure that the solution is symmetric. However, if the number of observed entries is sufficiently large, the algorithms still recover the original symmetric solution.

5.3. Low-dimensional embedding. RMD can be employed to find a lower-dimensional representation of the similarity matrix of the input points; see section 2.4. Our focus is on text data, where each row of the data matrix corresponds to a document and each column to a word, and the entries (i,j) of the matrix are the number of occurrences of the jth word in the ith document. We consider two well-known and already prepocessed text-word data sets from [37]: *k1b* with 2340 documents and 21839 words, and *hitech* with 2301 documents

and 10080 words. The threshold parameter τ in the TSM framework is selected according to the statistical analysis proposed in [27], that is, $\tau = 0.17$ for *k1b* and $\tau = 0.08$ for *hitech*. The quality of the embedding is evaluated using the mean angular deviation (MAD) metric as suggested in [27]:

$$\Delta = \frac{1}{|\mathcal{P}(\tau)|} \sum_{(i,j)\in\mathcal{P}(\tau)} \left| \cos^{-1}\left(\frac{\langle z_i, z_j\rangle}{\|z_i\|\|z_j\|}\right) - \cos^{-1}\left(\frac{\langle y_i, y_j\rangle}{\|y_i\|\|y_j\|}\right)\right|,$$

where the z_i's are the higher-dimensional points and the y_i's the lower-dimensional embeddings, and $\mathcal{P}(\tau) = \{(i,j) \mid \langle z_i, z_j\rangle - \tau\|z_j\|\|z_i\| > 0\}$. Figure 5.3 displays the MAD and average run times per iteration for different ranks. We run all the algorithms for 60 s.

Even though no algorithm outperforms the others, Figure 5.3 shows that using 3B-RMD is to be preferred in this application. Indeed, BCD, eBCD, and e3B achieve the lowest MAD in both experiments. Moreover, these three algorithms have the lowest average iteration time, being more than one second faster per iteration than Naive, A-Naive, and EM. For this reason, BCD, eBCD, and e3B perform more iterations than Naive, A-Naive, and EM within 60 s, resulting in more accurate approximations.

5.4. Compression of sparse data. The last application is the compression of sparse data. We consider sparse grayscale images, that is, 10000 randomly selected images from MNIST and Fashion MNIST (fMNIST), both of size 784×10000, and on two well-known sparse images:

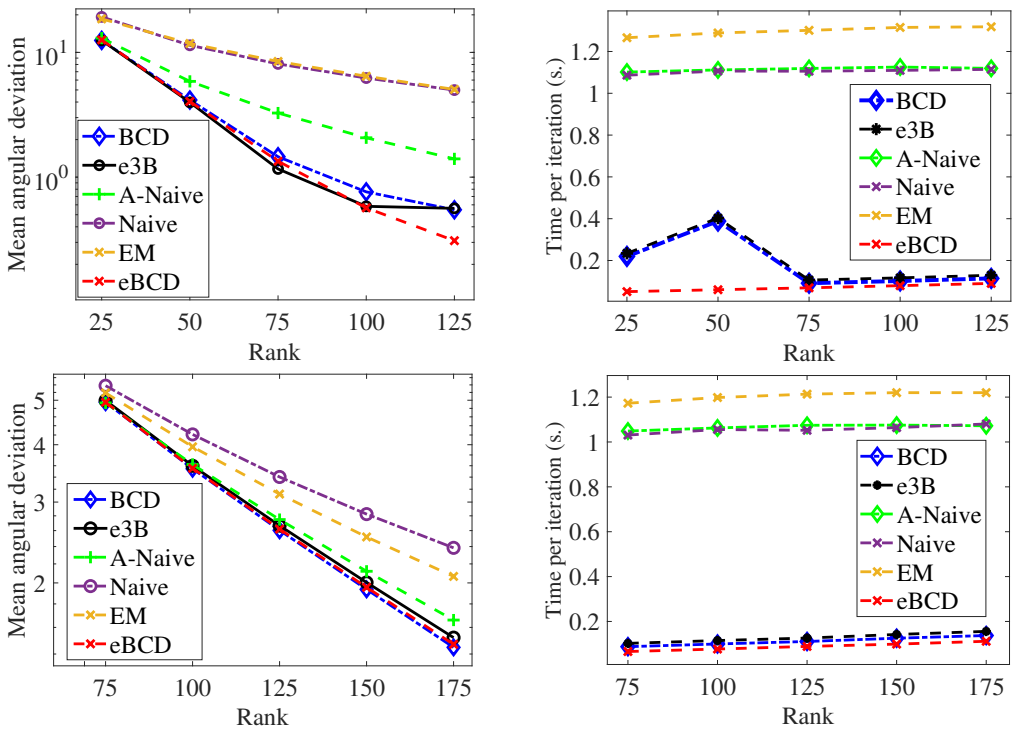

Figure 5.3. *MAD analysis and average iteration time for increasing values of the rank. The top two figures are for the k1b data set, and the bottom two for the hitech data set.*

Table 5.2

Compression of sparse matrices: we compare the average final relative error and the average iteration number, within the given time limit. The best values are highlighted in bold.

Data set		BCD	eBCD	e3B	Naive	A-Naive	EM	TSVD
MNIST (300 s.)	err	12.2%	11.6%	**11.5%**	12.3%	11.6%	12.9%	25.8%
$r = 70$	iter	1215	2159	1075	1088	899	413	/
fMNIST (300 s.)	err	9.6%	**9.1%**	9.2%	9.5%	**9.1%**	9.7%	14.0%
$r = 181$	iter	703	1498	637	914	777	513	/
Phantom (2.5 s.)	err	9.0%	**6.4%**	7.1	9.6%	7.8%	9.8%	19.2%
$r = 26$	iter	540	2898	508	374	340	278	/
Satellite (2.5 s.)	err	16.1%	**14.9%**	15.0%	17.0%	15.6%	18.2%	26.0%
$r = 12$	iter	1052	4385	983	374	352	252	/
Trec11 (10.2 s.)	err	31.3%	28.9%	**28.8%**	30.6%	**28.8%**	35.7%	57.3%
$r = 13$	iter	526	902	398	691	461	228	/
mycielskian (22.5 s.)	err	3.6%	**0.6%**	0.9%	8.0%	1.3%	9.0%	58.0%
$r = 14$	iter	516	1021	400	232	213	106	/
lock1074 (44.2 s.)	err	6.3%	**0.1%**	0.2%	15.6%	1.9%	29.2%	67.3%
$r = 12$	iter	576	1158	432	238	214	96	/
beaconfd (2.0 s.)	err	22.8 %	22.0%	**21.7**%	23.3%	**21.7**%	37.3%	39.12%
$r = 3$	iter	998	1514	680	525	419	206	/

Phantom (256×256) and Satellite (256×256). We also use sparse, structured matrices from various applications from [11]: Trec11 (235×1138), mycielskian (767×767), lock (1074×1074), and beaconfd (173×295). We choose the approximation rank r to ensure a desired level of compression. Specifically, when computing the RMD factors W and H of a sparse matrix $X \in \mathbb{R}^{m \times n}$ with $\text{nnz}(X)$ nonzero entries, the number of entries of WH is $r(n+m)$. Hence, the rank r is set to the closest integer such that $r \leq 0.5 \cdot \text{nnz}(X)/(n+m)$, which means a compression of 50%. Table 5.2 shows the relative error $\frac{\|X - \max(0, \bar{W}\bar{H})\|_F}{\|X\|_F}$, where $\bar{W}$ and $\bar{H}$ are the computed factors. Since we are considering real-world data of different dimensions, using a single time limit that is suitable for all the data sets is not meaningful. Therefore, we assign a time limit to all the algorithms (except the TSVD) scaled by the dimensions of each data set: we set a maximum time limit of $T = 300$ seconds for the largest data sets, namely MNIST and fMNIST with $m_{\max} = 784$ and $n_{\max} = 10000$. Then, we assign a scaled time limit T_{d} for a data set of dimension $m_{\text{d}} \times n_{\text{d}}$ using $T_{\text{d}} = T \frac{m_{\text{d}} \times n_{\text{d}}}{m_{\max} \times n_{\max}}$ seconds and display the average relative error over 10 different random initializations.

We observe that RMD outperforms the TSVD, as all the algorithms yield approximations that are significantly more accurate than the TSVD. Additionally, eBCD produces the most accurate solutions in five out of eight examples; moreover, eBCD outperforms BCD, achieving a smaller compression error for all the data sets considered. We also observe that the structure of the matrix has a crucial impact on the effectiveness of the compression using RMD. For example, for lock1074, the improvement of RMD is impressive: the relative error of eBCD is 0.1% while that of the TSVD is 67.3%. On the other hand, the improvement for fMNIST is less pronounced: from 14% relative error for the TSVD to 9.6% for RMD.

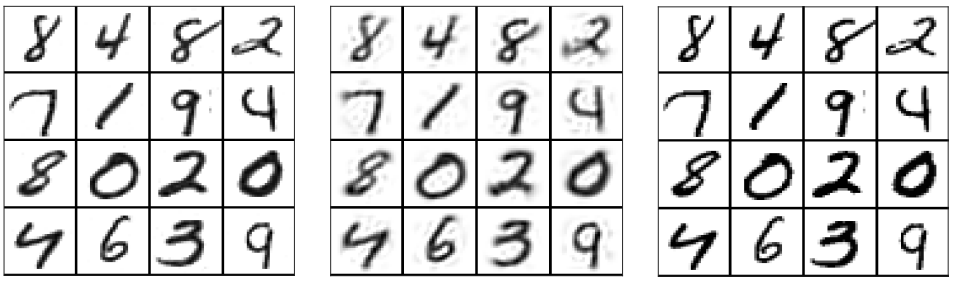

Figure 5.4. *Visual illustration: TSVD (relative error*$= 25.8\%$*) against RMD with eBCD (relative error*$= 11.6\%$*) on the MNIST set with* 50% *compression* ($r = 70$).

Figure 5.4 visually illustrates the difference between TSVD and RMD for 50% compression. One can visually appreciate that the eBCD approximation is consistently more accurate than the TSVD, as hinted by the errors reported in Table 5.2.

6. Conclusion. The RMD is a relatively recent matrix decomposition model (Saul [27]) that finds applications in the compression of sparse data, entry-dependent matrix completion, and manifold learning. Moreover, we showed, for the first time, how it can be used for the completion of Euclidean distance matrices when only the smallest (or largest) entries are observed. Our first main contribution is to establish the nonequivalence of the two main models used to compute RMDs, namely LS-RMD and Latent-RMD (Corollary 3.1). Furthermore, we derive an explicit connection between the optimal function values of the two, demonstrating that the solution of Latent-RMD is, in the worst case, four times larger than that of LS-RMD (Theorem 3.2). Our second main contribution is to prove the convergence of the BCD scheme for solving 3B-RMD by demonstrating that it falls into the general framework of exact BCD, where one of the subproblems is strictly quasi-convex (Theorem 4.1). Our third and most important contribution is a novel algorithm, eBCD, which uses extrapolation to accelerate BCD. Our new algorithm extends the well-known LMaFit algorithm from [35] for matrix completion. We proved the subsequence convergence of the eBCD-NMD scheme to a stationary point under mild assumptions (Theorem 4.7). Finally, we provided extensive numerical results for four applications on synthetic and real data sets, showing that eBCD performs favorably compared to the to the state of the art, while having convergence guarantees which other algorithms lack.

Future work includes investigating the well-posedness of the LS-RMD in (1.1), designing randomized algorithms to tackle large-scale problems, and exploring more advanced RMDs such as $X \approx W_1H_1 + \max(0, W_2H_2) - \max(0, W_3H_3)$ that could handle mixed-signed and nonsparse data.

Appendix A.

A.1. Proof of Lemma 4.3.

Proof. Let $(W(\alpha), H(\alpha))$ be generated by (4.7), and let $S = \mathcal{P}_\Omega(X - WH) - \mathcal{P}_{\Omega^C}(\max(0, WH))$ be the residual with respect to the kth iterate (W, H). Recall that $Z_\alpha = WH + \alpha S$, $P = H^\top(HH^\top)^{-1}H$, and $E(\alpha) = W(\alpha)W(\alpha)^T$. We denote $Q(\alpha)R(\alpha)\Pi = Z_\alpha H^T$,

the QR factorization of $Z_\alpha H^T$ with column pivoting, where Π is a permutation matrix. Note that Π is needed only if $Z_\alpha H^T$ is rank-deficient; otherwise, one can consider Π as the identity matrix. We have

(A.1)
$$W(\alpha)H(\alpha) - WH = \left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1} - W\right]H + W(\alpha)\left[H(\alpha) - R(\alpha)\Pi(HH^T)^{-1}H\right].$$

The first term in (A.1) can be written as

(A.2)
$$\begin{aligned}\left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1} - W\right]H &= W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H - WH\\ &= Z_\alpha H^T(HH^T)^{-1}H - W(HH^T)(HH^T)^{-1}H = Z_\alpha P - WHP\\ &= (Z_\alpha - WH)P = \alpha SP.\end{aligned}$$

Similarly, we write the second term of the summation in terms of the residual. Using (A.2),

$$\begin{aligned}H(\alpha) &= W(\alpha)^T Z_\alpha = W(\alpha)^T(WH + \alpha S)\\ &= W(\alpha)^T\left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H - (W(\alpha)R(\alpha)\Pi(HH^T)^{-1} - W)H + \alpha S\right]\\ &= W(\alpha)^T\left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H + \alpha S(I-P)\right].\end{aligned}$$

We multiply both sides by $W(\alpha)$ to obtain

$$\begin{aligned}W(\alpha)H(\alpha) &= W(\alpha)W(\alpha)^T\left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H + \alpha S(I-P)\right]\\ &= W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H + \alpha E(\alpha)S(I-P).\end{aligned}$$

Rearranging the terms, we have

$$W(\alpha)\left[H(\alpha) - R(\alpha)\Pi(HH^T)^{-1}H\right] = \alpha E(\alpha)S(I-P). \tag{A.3}$$

Finally, combining (A.1), (A.2), and (A.3) we obtain

(A.4)
$$\begin{aligned}W(\alpha)H(\alpha) - WH &= \left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1} - W\right]H + W(\alpha)\left[H(\alpha) - R(\alpha)\Pi(HH^T)^{-1}H\right]\\ &= \alpha SP + \alpha Q(\alpha)S(I-P)\\ &= \alpha(I - Q(\alpha))SP + \alpha Q(\alpha)S.\end{aligned}$$

From the properties of the orthogonal projection, (A.4) implies

$$\|W(\alpha)H(\alpha) - WH\|_F^2 = \alpha^2\|(I - Q(\alpha))SP\|_F^2 + \alpha^2\|Q(\alpha)S\|_F^2, \tag{A.5}$$

which proves the second inequality in (4.12).

Let us now obtain the first inequality. From (A.4) and the properties of orthogonal projection,

$$\begin{aligned}\langle \alpha S, W(\alpha)H(\alpha) - WH\rangle &= \alpha^2\langle S, (I - E(\alpha))SP\rangle + \alpha^2\langle S, E(\alpha)S\rangle\\ &= \alpha^2\|(I - E(\alpha))SP\|_F^2 + \alpha^2\|E(\alpha)S\|_F^2\\ &= \|W(\alpha)H(\alpha) - WH\|_F^2.\end{aligned} \tag{A.6}$$

From the observation (4.9) on Z_α, the definition of $\sigma_{WH}(\alpha)$, and (A.6), we get

$$\begin{aligned}\sigma_{WH}(\alpha) &= \|S\|_F^2 - \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - Z_\alpha\|_F^2 = \|S\|_F^2 - \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - WH - \alpha S\|_F^2 \\ &= \|S\|_F^2 - \frac{1}{\alpha^2}\left[\|W(\alpha)H(\alpha) - WH\|_F^2 + \alpha^2\|S\|_F^2 - 2\alpha\langle S, W(\alpha)H(\alpha) - WH\rangle\right] \\ &= \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - WH\|_F^2,\end{aligned}$$

which proves the first equality in (4.12). ∎

A.2. Proof of Lemma 4.4.

Proof. Let us first show the continuity of the function in (4.13) when $\alpha \to 1^+$. Recall that $Z_1 = Z$ and that $W(\alpha)$ is an orthogonal basis of $\mathcal{R}(Z_\alpha H^T)$. Therefore, the orthogonal projection operator into $\mathcal{R}(Z_\alpha H^T)$ can be written as $W(\alpha)W(\alpha)^T = (Z_\alpha H^T)(Z_\alpha H^T)^\dagger$. If $\text{rank}(Z_\alpha H^T)$=$\text{rank}(ZH^T)$ in some interval $[1, \widehat{\alpha}]$, then by the continuity of the Moore–Penrose inverse [30], we have

$$\lim_{\alpha\to 1^+} W(\alpha)W(\alpha)^T = \lim_{\alpha\to 1^+} (Z_\alpha H^T)(Z_\alpha H^T)^\dagger = (ZH^T)(ZH^T)^\dagger = W(1)W(1)^T. \tag{A.7}$$

From (A.7) and the continuity of the product,

$$\lim_{\alpha\to 1^+} W(\alpha)H(\alpha) = \lim_{\alpha\to 1^+} W(\alpha)W(\alpha)^T Z_\alpha = W(1)W(1)^T Z = W(1)H(1), \tag{A.8}$$

which leads to the continuity of (4.13) when $\alpha \to 1^+$. Let us recall that

$$Z_\alpha = WH + \alpha S,\ P_{\Omega^C}(S) = -P_{\Omega^C}(\max(0, WH)),\ P_{\Omega^C}(Z(\alpha)) = P_{\Omega^C}(\min(0, (W(\alpha)H(\alpha))).$$

We have

$$\begin{aligned}&\frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_{\Omega^C}(S(\alpha))\|_F^2 \\ &\quad= \frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_{\Omega^C}(W(\alpha)H(\alpha) - Z(\alpha))\|_F^2 \\ &\quad= \frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - WH - \alpha S)\|_F^2 - \|P_{\Omega^C}(W(\alpha)H(\alpha) - \min(0, (W(\alpha)H(\alpha)))\|_F^2 \\ &\quad= \frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - \min(0, WH) + (\alpha - 1)\max(0, WH))\|_F^2 \\ &\qquad- \|P_{\Omega^C}(W(\alpha)H(\alpha) - \min(0, (W(\alpha)H(\alpha)))\|_F^2.\end{aligned} \tag{A.9}$$

Since (4.13) is continuous for $\alpha \to 1^+$, we just need to show that it is larger than or equal to zero at $\alpha = 1$. Using (A.8) and (A.9),

$$\begin{aligned}
&\|P_{\Omega^C}(W(1)H(1) - \min(0, WH))\|_F^2 - \|P_{\Omega^C}(\max(0, (W(1)H(1)))\|_F^2 \\
&\quad = \|P_{\Omega^C}(W(1)H(1))\|_F^2 + \|P_{\Omega^C}(\min(0, WH))\|_F^2 - 2\langle P_{\Omega^C}(W(1)H(1)), P_{\Omega^C}(\min(0, WH))\rangle \\
&\qquad - \|P_{\Omega^C}(\max(0, (W(1)H(1)))\|_F^2 \\
&\quad = \|P_{\Omega^C}(\min(0, W(1)H(1)))\|_F^2 + \|P_{\Omega^C}(\min(0, WH))\|_F^2 \\
&\qquad - 2\langle P_{\Omega^C}(\min(0, W(1)H(1))), P_{\Omega^C}(\min(0, WH))\rangle \\
&\qquad - 2\langle P_{\Omega^C}(\max(0, W(1)H(1))), P_{\Omega^C}(\min(0, WH))\rangle \\
&\quad = \|P_{\Omega^C}(\min(0, W(1)H(1)) - (\min(0, WH))\|_F^2 \\
&\qquad - 2\langle P_{\Omega^C}(\max(0, W(1)H(1))), P_{\Omega^C}(\min(0, WH))\rangle \geq 0,
\end{aligned}$$

which concludes the proof. ∎

A.3. Proof of Lemma 4.5.

Proof. By an argument analogous to the one used in Lemma 4.5, we get the continuity of (4.14) as $\alpha \to 1^+$. Next, we show that (4.14) is zero at $\alpha = 1$. By the definition of the residual in (4.8),

$$\begin{aligned}
&\frac{1}{\alpha^2}\|P_\Omega(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_\Omega(S(\alpha))\|_F^2 \\
&\quad = \|P_\Omega(W(\alpha)H(\alpha) - X + X - WH - \alpha S)\|_F^2 - \|P_\Omega(S(\alpha))\|_F^2 \\
&\quad = \|-P_\Omega(S(\alpha)) + (1-\alpha)P_\Omega(S))\|_F^2 - \|P_\Omega(S(\alpha))\|_F^2,
\end{aligned}$$

which is equal to zero for $\alpha = 1$. ∎

Acknowledgments. We are grateful to the anonymous reviewers who carefully read the manuscript; their feedback helped us improve our paper.

REFERENCES

[1] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comput. Optim. Appl., 12 (1999), pp. 13–30, https://doi.org/10.1023/A:1008655427845.

[2] A. M. S. Ang and N. Gillis, *Accelerating nonnegative matrix factorization algorithms using extrapolation*, Neural Comput., 31 (2019), pp. 417–439, https://doi.org/10.1162/neco_a_01157.

[3] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality*, Math. Oper. Res., 35 (2010), pp. 438–457, https://doi.org/10.1287/moor.1100.0449.

[4] A. Awari, N. Gillis, and A. Vandaele, *Alternating Direction Method of Multipliers for Nonlinear Matrix Decompositions*, preprint, arXiv:2512.17473, 2025.

[5] A. Awari, H. Nguyen, S. Wertz, A. Vandaele, and N. Gillis, *Coordinate descent algorithm for nonlinear matrix decomposition with the ReLU function*, in Proceedings of the 32nd European Signal Processing Conference (EUSIPCO), 2024, pp. 2622–2626, https://doi.org/10.23919/EUSIPCO63174.2024.10715025.

[6] L. Balzano, R. Nowak, and B. Recht, *Online identification and tracking of subspaces from highly incomplete information*, in Proceedings of the Annual Allerton Conference on Communication, Control, and Computing, 2010, pp. 704–711.

[7] S. Bhattacharya and S. Chatterjee, *Matrix completion with data-dependent missingness probabilities*, IEEE Trans. Inform. Theory, 68 (2022), pp. 6762–6773, https://doi.org/10.1109/TIT.2022.3170244.
[8] Y. Bi and J. Lavaei, *On the absence of spurious local minima in nonlinear low-rank matrix recovery problems*, in Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 379–387.
[9] N. Boumal and P. A. Absil, *Low-rank matrix completion via preconditioned optimization on the Grassmann manifold*, Linear Algebra Appl., 475 (2015), pp. 200–239, https://doi.org/10.1016/j.laa.2015.02.027.
[10] M. Ciaperoni, A. Gionis, and H. Mannila, *The Hadamard decomposition problem*, Data Min. Knowl. Disc., 38 (2024), pp. 1–42, https://doi.org/10.1007/s10618-024-01033-y.
[11] W. Dai, E. Kerman, and O. Milenkovic, *A geometric approach to low-rank matrix completion*, IEEE Trans. Inform. Theory, 58 (2012), pp. 237–247, https://doi.org/10.1109/TIT.2011.2171521.
[12] P. Delsarte and Y. Kamp, *Low rank matrices with a given sign pattern*, SIAM J. Discrete Math., 2 (1989), pp. 51–63, https://doi.org/10.1137/0402006.
[13] H.-r. Fang and D. P.O'Leary, *Euclidean distance matrix completion problems*, Optim. Methods Softw., 27 (2012), pp. 695–717, https://doi.org/10.1080/10556788.2011.643888.
[14] R. S. Ganti, L. Balzano, and R. Willett, *Matrix completion under monotonic single index models*, Adv. Neural Inform. Process. Syst., 28 (2015), https://proceedings.neurips.cc/paper_files/paper/2015/hash/f197002b9a0853eca5e046d9ca4663d5-Abstract.html.
[15] G. H. Golub and C.F. Van Loan, *Matrix Computations*, JHU Press, Baltimore, 2013.
[16] F. Goyens, C. Cartis, and A. Eftekhari, *Nonlinear matrix recovery using optimization on the Grassmann manifold*, Appl. Comput. Harmon. Anal., 62 (2023), pp. 498–542, https://doi.org/10.1016/j.acha.2022.11.001.
[17] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett., 26 (2000), pp. 127–136, https://doi.org/10.1016/S0167-6377(99)00074-7.
[18] L. Grippo and M. Sciandrone, *Introduction to Methods for Nonlinear Optimization*, UNITEXT 152, Springer Nature, Cham, 2023.
[19] N. Hyeon-Woo, M. Ye-Bin, and T. H. Oh, *FedPara: Low-rank Hadamard Product for Communication-Efficient Federated Learning*, preprint, arXiv:2108.06098, 2021.
[20] N. Krislock and H. Wolkowicz, *Euclidean Distance Matrices and Applications*, Springer, Cham, 2012.
[21] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, Springer, Cham, 2007.
[22] H. Liu, P. Wang, L. Huang, Q. Qu, and L. Balzano, *Symmetric Matrix Completion with ReLU Sampling*, preprint, arXiv:2406.05822, 2024.
[23] L. Loconte, AM. Sladek, S. Mengel, M. Trapp, A. Solin, N. Gillis, and A. Vergari, *Subtractive mixture models via squaring: Representation and learning*, in Proceedings of the International Conference on Learning Representations, 2024.
[24] R. Naik, N. Trivedi, D. Tarzanagh, and L. Balzano, *Truncated matrix completion an empirical study*, in Proceedings of the 30th European Signal Processing Conference (EUSIPCO), IEEE, 2022, pp. 847–851.
[25] M. Patriksson, *Decomposition methods for differentiable optimization problems over Cartesian product sets*, Comput. Optim. Appl., 9 (1998), pp. 5–42, https://doi.org/10.1023/A:1018358602892.
[26] M. J. Powell, *On search directions for minimization algorithms*, Math. Program., 4 (1973), pp. 193–201, https://doi.org/10.1007/BF01584660.
[27] L. K. Saul, *A geometrical connection between sparse and low-rank matrices and its application to manifold learning*, Trans. Mach. Learn. Res., (2022), https://openreview.net/forum?id=p8gncJbMit.
[28] L. K. Saul, *A nonlinear matrix decomposition for mining the zeros of sparse data*, SIAM J. Math. Data Sci., 4 (2022), pp. 431–463, https://doi.org/10.1137/21M1405769.
[29] G. Seraghiti, A. Awari, A. Vandaele, M. Porcelli, and N. Gillis, *Accelerated algorithms for nonlinear matrix decomposition with the ReLU function*, in Proceedings of the International Workshop on Machine Learning for Signal Processing, 2023.
[30] G. W. Stewart, *On the continuity of the generalized inverse*, SIAM J. Appl. Math., 17 (1969), pp. 33–45, https://doi.org/10.1137/0117004.

[31] A. Tasissa and R. Lai, *Exact reconstruction of Euclidean distance geometry problem using low-rank matrix completion*, IEEE Trans. Inform. Theory, 65 (2018), pp. 3124–3144, https://doi.org/10.1109/TIT.2018.2881749.

[32] P. Tseng, *Decomposition algorithm for convex differentiable minimization*, J. Optim. Theory Appl., 70 (1991), pp. 109–135, https://doi.org/10.1007/BF00940507.

[33] Q. Wang, C. Cui, and D. Han, *A momentum accelerated algorithm for ReLU-based nonlinear matrix decomposition*, IEEE Signal Process. Lett., 31 (2024), pp. 2865–2869, https://doi.org/10.1109/LSP.2024.3475910.

[34] Q. Wang, Y. Qu, C. Cui, and D. Han, *An accelerated alternating partial Bregman algorithm for ReLU-based matrix decomposition*, J. Sci. Comput., 105 (2025), https://doi.org/10.1007/s10915-025-03067-w.

[35] Z. Wen, W. Yin, and Y. Zhang, *Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm*, Math. Program. Comput., 4 (2012), pp. 333–361, https://doi.org/10.1007/s12532-012-0044-1.

[36] S. Wertz, A. Vandaele, and N. Gillis, *Efficient algorithms for the hadamard decomposition*, in Proceedings of 2025 IEEE 35th International Workshop on Machine Learning for Signal Processing (MLSP), Istanbul, Turkiye, 2025, pp. 1–6, https://doi.org/10.1109/MLSP62443.2025.11204314.

[37] S. Zhong and J. Ghosh, *Generative model-based document clustering: A comparative study*, Knowl. Inf. Syst., 8 (2005), pp. 374–384, https://doi.org/10.1007/s10115-004-0194-1.