



Robust PCA via Regularized REAPER with a Matrix-Free Proximal Algorithm

Robert Beinert¹ · Gabriele Steidl¹

Received: 12 May 2020 / Accepted: 16 January 2021 / Published online: 15 February 2021
© The Author(s) 2021

Abstract

Principal component analysis (PCA) is known to be sensitive to outliers, so that various robust PCA variants were proposed in the literature. A recent model, called REAPER, aims to find the principal components by solving a convex optimization problem. Usually the number of principal components must be determined in advance and the minimization is performed over symmetric positive semi-definite matrices having the size of the data, although the number of principal components is substantially smaller. This prohibits its use if the dimension of the data is large which is often the case in image processing. In this paper, we propose a regularized version of REAPER which enforces the sparsity of the number of principal components by penalizing the nuclear norm of the corresponding orthogonal projector. If only an upper bound on the number of principal components is available, our approach can be combined with the L -curve method to reconstruct the appropriate subspace. Our second contribution is a matrix-free algorithm to find a minimizer of the regularized REAPER which is also suited for high-dimensional data. The algorithm couples a primal-dual minimization approach with a thick-restarted Lanczos process. This appears to be the first efficient convex variational method for robust PCA that can handle high-dimensional data. As a side result, we discuss the topic of the bias in robust PCA. Numerical examples demonstrate the performance of our algorithm.

Keywords Robust PCA · Regularized REAPER · Matrix-free PCA · PCA offset · Thick-restarted Lanczos algorithm

Mathematics Subject Classification 58C05 · 62H25 · 65K10

1 Introduction

Principal component analysis (PCA) [37] realizes the dimensionality reduction in data by projecting them onto those affine subspace which minimizes the sum of the squared Euclidean distances between the data points and their projections. Unfortunately, PCA is very sensitive to outliers, so that various robust approaches were developed in robust statistics [17,28,46] and nonlinear optimization. In this paper, we focus on the second one.

One possibility to make PCA robust consists in removing outliers before computing the principal components which has the serious drawback that outliers are difficult to identify and other data points are often falsely labeled as outliers. Another approach assigns different weights to data points based on their estimated relevance, to get a weighted PCA [20] or repeatedly estimate the model parameters from a random subset of data points until a satisfactory result indicated by the number of data points within a certain error threshold is obtained [11]. In a similar vein, least trimmed squares PCA models [38,41] aim to exclude outliers from the squared error function, but in a deterministic way. In [44], a dual principal component pursuit is used for this purpose. The variational model in [4] decomposes the data matrix into a low rank and a sparse part. Related approaches such as [7,32,51] separate the low-rank component from the column sparse one using different norms in the variational model. Another group of robust PCA replaces the squared L_2 norm in the PCA model by the L_1 norm [18]. Unfortunately, this norm is not rotationally invariant, i.e., when rotating the centered

R. Beinert was supported by the Austrian Science Fund (FWF) Grant P28858, and G. Steidl by German Research Foundation (DFG) Grant STE 571/16-1.

✉ Robert Beinert
beinert@math.tu-berlin.de
Gabriele Steidl
steidl@math.tu-berlin.de

¹ Institut für Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany

data points, the minimizing subspace is not rotated in the same way. Replacing the squared Euclidean norm in the PCA model by just the Euclidean one leads to a non-convex robust PCA model with minimization over the Stiefel or Grassmannian manifold, see, e.g., [9,26,31,34]. Instead of the previous model which minimizes over the sparse number of directions spanning the low-dimensional subspace, it is also possible to minimize over the orthogonal projectors onto the desired subspace. This has the advantage that the minimization can be performed over symmetric positive semi-definite matrices, e.g., using methods from semi-definite programming, and the disadvantage that the dimension of the projectors is as large as the data now. This prohibits this approach for many applications in particular in image processing. The projector PCA model is still non-convex, and a convex relaxation, called REAPER, was recently proposed by Lerman et al. [27]. An extensive comparison of the benefits and drawbacks of the different approaches in the rich literature of robust PCA as well as of the related numerical algorithms can, for instance, be found in [26].

In this paper, we build up on the advantages of the convex REAPER model, but modify it in two important directions: (i) by penalizing the nuclear norm of the approximated projectors, our model does only require an upper bound on the dimension of the desired subspace. Having the same effect as the sparsity promotion of the 1-norm, the nuclear norm—the 1-norm of the eigenvalues—promotes low-rank matrices or, equivalently, sparse eigenvalue decompositions; (ii) by combining primal-dual minimization techniques with a thick-restarted Lanczos process, we are able to handle high-dimensional data. We call our new convex model rREAPER. We provide all computation steps leading to an efficient and a provable convergent algorithm to the minimum of the objective function. A performance analysis following the lines of [27] is given. The choice of the offset in robust PCA is an interesting problem which is not fully discussed in the literature so far. Usually, the geometric median is used. We do not provide a full solution of this issue, but show that under some assumptions the affine hyperplane in \mathbb{R}^d having the smallest Euclidean distance to $n > d$ given data points goes through $d + 1$ of these points. We underline our theoretical findings by numerical examples.

The outline of this paper is as follows: preliminaries from linear algebra and convex analysis are given in Sect. 2. In Sect. 3, we introduce our regularized REAPER model. The basic primal-dual algorithm for its minimization is discussed in Sect. 4. The algorithm is formulated with respect to the full projection matrix. The matrix-free version of the algorithm is given in Sect. 5. It is based on the thick-restarted Lanczos algorithm and is suited for high-dimensional data. In Sect. 6, we examine the performance analysis of rREAPER along the lines of [27]. Some results on the offset in robust PCA are proved in Sect. 7. The very good performance of

rREAPER in particular for high-dimensional data is demonstrated in Sect. 8. Further, the relation between the dimension of the wanted subspace and the regularization parameter is addressed via the L -curve method. Section 9 finishes the paper with conclusions and directions of future research.

2 Notation and Preliminaries

Throughout this paper, we will use the following notation and basic facts from linear algebra and convex analysis which can be found in detail in various monographs and overview papers as [1,3,6,13,40].

Linear algebra By $\|\cdot\|_2$ we denote the Euclidean vector norm and by $\|\cdot\|_1$ the norm which sums up the absolute vector components. Recall that for any $\mathbf{x} \in \mathbb{R}^n$,

$$\frac{1}{\sqrt{n}}\|\mathbf{x}\|_1 \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1. \tag{1}$$

Let $\mathbf{1}_n$ resp. $\mathbf{0}_n$ be the vectors having n entries 1, resp., 0. Analogously, we write $\mathbf{1}_{n,d}$ and $\mathbf{0}_{n,d}$ for the all-one and all-zero matrix in $\mathbb{R}^{n,d}$. Further, \mathbf{I}_n is the $n \times n$ identity matrix. Let $\text{tr } \mathbf{A}$ denote the *trace* of the quadratic matrix $\mathbf{A} \in \mathbb{R}^{n,n}$, i.e., the sum of its eigenvalues. On $\mathbb{R}^{n,d}$ the Hilbert–Schmidt inner product is defined by

$$\langle \mathbf{X}, \mathbf{Y} \rangle := \text{tr}(\mathbf{X}^T \mathbf{Y}) = \text{tr}(\mathbf{Y} \mathbf{X}^T), \quad \mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n,d},$$

and the corresponding so-called *Frobenius norm* by

$$\|\mathbf{X}\|_F^2 = \langle \mathbf{X}, \mathbf{X} \rangle.$$

Let $\mathcal{S}(n) \subset \mathbb{R}^{n,n}$ denote the linear subspace of symmetric matrices. For two symmetric matrices $\mathbf{A}, \mathbf{B} \in \mathcal{S}(n)$, we write $\mathbf{A} \preceq \mathbf{B}$ if $\mathbf{B} - \mathbf{A}$ is positive semi-definite. Every $\mathbf{A} \in \mathcal{S}(n)$ has a spectral decomposition

$$\mathbf{A} = \mathbf{U} \text{diag}(\boldsymbol{\lambda}_A) \mathbf{U}^T,$$

where $\boldsymbol{\lambda}_A \in \mathbb{R}^n$ denotes the vector containing the eigenvalues of \mathbf{A} in descending order $\lambda_1 \geq \dots \geq \lambda_n$ and \mathbf{U} is the orthogonal matrix having the corresponding orthogonal eigenvectors as columns. The nuclear norm (trace norm) of $\mathbf{A} \in \mathcal{S}(n)$ is given by

$$\|\mathbf{A}\|_{\text{tr}} := \sum_{j=1}^n |\lambda_j|.$$

The trace and Frobenius norm correspond to the Schatten 1-norm and 2-norm, respectively, where the Schatten p -norm with $1 \leq p \leq \infty$ of a symmetric matrix \mathbf{A} is defined by $\|\mathbf{A}\|_{S_p} := \|\boldsymbol{\lambda}_A\|_p$. Recall that $\boldsymbol{\Pi} \in \mathbb{R}^{n,n}$ is an orthogonal

projector if $\mathbf{\Pi} \in \mathcal{S}(n)$ and $\mathbf{\Pi}^2 = \mathbf{\Pi}$. This is equivalent to the statement that $\mathbf{\Pi} \in \mathcal{S}(n)$ and has only eigenvalues in $\{0, 1\}$. The *nuclear norm* is the unique norm such that

$$\text{rank}(\mathbf{\Pi}) = \|\mathbf{\Pi}\|_{\text{tr}}$$

for every orthogonal projector $\mathbf{\Pi}$.

For a given norm $\|\cdot\|$ on \mathbb{R}^n , the *dual norm* is defined by

$$\|\mathbf{x}\|_* := \max_{\|\mathbf{y}\| \leq 1} \langle \mathbf{x}, \mathbf{y} \rangle.$$

In particular, for a matrix $\mathbf{X} = (\mathbf{x}_1 | \dots | \mathbf{x}_N) \in \mathbb{R}^{n,N}$ we will be interested in the norm

$$\|\mathbf{X}\|_{2,1} := \sum_{k=1}^N \|\mathbf{x}_k\|_2,$$

which can be considered as norm on \mathbb{R}^{nN} by arranging the columns of the matrix into a vector. Its dual norm is given by

$$\|\mathbf{X}\|_{2,1,*} = \|\mathbf{X}\|_{2,\infty} := \max_{k=1,\dots,N} \|\mathbf{x}_k\|_2.$$

Convex analysis Let $\Gamma_0(\mathbb{R}^n)$ denote the space of proper, lower semi-continuous, convex functions mapping from \mathbb{R}^n into the extended real numbers $(-\infty, \infty]$. The *indicator function* $\iota_{\mathcal{C}}$ of $\mathcal{C} \subseteq \mathbb{R}^n$ is defined by

$$\iota_{\mathcal{C}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{C}, \\ +\infty & \text{otherwise.} \end{cases}$$

We have $\iota_{\mathcal{C}} \in \Gamma_0(\mathbb{R}^n)$ if and only if \mathcal{C} is non-empty, convex and closed.

For $f \in \Gamma_0(\mathbb{R}^n)$, the *proximal mapping* is defined by

$$\text{prox}_f(\mathbf{x}) := \underset{\mathbf{y} \in \mathbb{R}^n}{\text{argmin}} \left\{ f(\mathbf{y}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \right\}.$$

Indeed, the minimizer exists and is unique [40, Thm. 31.5]. If $\mathcal{C} \subset \mathbb{R}^n$ is a non-empty, closed, convex set, then the proximal mapping of a multiple of $\iota_{\mathcal{C}}$ is just the orthogonal projection onto \mathcal{C} , i.e.,

$$\text{prox}_{\sigma \iota_{\mathcal{C}}}(\mathbf{x}) = \text{proj}_{\mathcal{C}}(\mathbf{x}), \quad \sigma > 0.$$

In particular, the orthogonal projection onto the halfspace $\mathcal{H}(\mathbf{a}, \beta) := \{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{a}, \mathbf{x} \rangle \leq \beta\}$ with $\mathbf{a} \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$ can be computed by

$$\text{proj}_{\mathcal{H}(\mathbf{a}, \beta)}(\mathbf{x}) = \mathbf{x} - \frac{(\langle \mathbf{a}, \mathbf{x} \rangle - \beta)_+}{\|\mathbf{a}\|_2^2} \mathbf{a},$$

where $(y)_+ := \max\{0, y\}$. Further, the orthogonal projection onto the hypercube $\mathcal{Q} := [0, 1]^n$ is given by

$$\text{proj}_{\mathcal{Q}}(\mathbf{x}) = \left(\max \{ \min\{x_j, 1\}, 0 \} \right)_{j=1}^n. \tag{2}$$

The *Fenchel dual* of $f \in \Gamma_0(\mathbb{R}^n)$ is the function $f^* \in \Gamma_0(\mathbb{R}^n)$ defined by

$$f^*(\mathbf{p}) := \max_{\mathbf{x} \in \mathbb{R}^n} \langle \mathbf{p}, \mathbf{x} \rangle - f(\mathbf{x}).$$

The dual function of a norm is just the indicator function of the unit ball with respect to its dual norm. In particular, we have for $\|\cdot\|_{2,1} : \mathbb{R}^{n,N} \rightarrow \mathbb{R}$ that

$$\|\mathbf{X}\|_{2,1}^* = \iota_{\mathcal{B}_{2,\infty}}(\mathbf{X}), \tag{3}$$

where $\mathcal{B}_{2,\infty} := \{\mathbf{X} \in \mathbb{R}^{n,N} : \|\mathbf{x}_k\|_2 \leq 1 \text{ for all } k = 1, \dots, N\}$.

3 Regularized REAPER

Given N data points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^n$, the classical PCA finds a d -dimensional affine subspace $\{\mathbf{A}\mathbf{t} + \mathbf{b} : \mathbf{t} \in \mathbb{R}^d\}$, $1 \leq d \ll n$, by minimizing

$$\sum_{k=1}^N \min_{\mathbf{t} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{t} + \mathbf{b} - \mathbf{x}_k\|_2^2 = \sum_{k=1}^N \|(\mathbf{A}\mathbf{A}^T - \mathbf{I}_n)(\mathbf{b} - \mathbf{x}_k)\|_2^2 \tag{4}$$

subject to $\mathbf{A}^T \mathbf{A} = \mathbf{I}_d$

over $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n,d}$. It is not hard to check that the affine subspace goes through the *offset* (bias)

$$\bar{\mathbf{b}} := \frac{1}{N}(\mathbf{x}_1 + \dots + \mathbf{x}_N). \tag{5}$$

Therefore, we can reduce our attention to data points $\mathbf{x}_k - \bar{\mathbf{b}}$, $k = 1, \dots, N$, which we denote by \mathbf{x}_k again, and minimize over the linear d -dimensional subspaces through the origin, i.e.,

$$\min_{\mathbf{A} \in \mathbb{R}^{n,d}} \sum_{k=1}^N \|(\mathbf{A}\mathbf{A}^T - \mathbf{I}_n)\mathbf{x}_k\|_2^2 \quad \text{subject to } \mathbf{A}^T \mathbf{A} = \mathbf{I}_d,$$

where $\mathbf{X} := (\mathbf{x}_1 | \dots | \mathbf{x}_N) \in \mathbb{R}^{n,N}$.

Unfortunately, the solution of this minimization problem is sensitive to outliers. Therefore, several robust PCA variants were proposed in the literature. A straightforward approach consists in just skipping the square in the Euclidean norm leading to

$$\min_{A \in \mathbb{R}^{n,d}} \sum_{k=1}^N \|(\mathbf{A}\mathbf{A}^T - \mathbf{I}_n)\mathbf{x}_k\|_2 = \|\mathbf{A}\mathbf{A}^T\mathbf{X} - \mathbf{X}\|_{2,1} \tag{6}$$

subject to $\mathbf{A}^T\mathbf{A} = \mathbf{I}_d$.

This is a non-convex model which requires the minimization over matrices \mathbf{A} in the so-called Stiefel manifold,

$$\text{St}(n, d) := \{\mathbf{A} \in \mathbb{R}^{n,d} : \mathbf{A}^T\mathbf{A} = \mathbf{I}_d\},$$

see [9,26,34,35].

Another approach is based on the observation that $\mathbf{\Pi} := \mathbf{A}\mathbf{A}^T$ is the orthogonal projector onto the linear subspace spanned by the columns of \mathbf{A} . Since the linear subspace is d -dimensional, exactly d eigenvalues of $\mathbf{\Pi}$ have to be one. Thus, problem (6) can be reformulated as

$$\min_{\mathbf{\Pi} \in \mathcal{S}(n)} \|\mathbf{\Pi}\mathbf{X} - \mathbf{X}\|_{2,1} \tag{7}$$

subject to $\lambda_{\mathbf{\Pi}} \in \{0, 1\}^n, \text{tr}(\mathbf{\Pi}) = d$.

Having computed $\mathbf{\Pi}$, we can determine \mathbf{A} by spectral decomposition. Unfortunately, (7) is still a non-convex model which is moreover NP hard to solve. Therefore, Lerman et al. [27] suggested to replace it by a convex relaxation, called REAPER,

$$\min_{\mathbf{P} \in \mathcal{S}(n)} \|\mathbf{P}\mathbf{X} - \mathbf{X}\|_{2,1} \quad \text{subject to} \quad \mathbf{0}_{n,n} \leq \mathbf{P} \leq \mathbf{I}_n, \text{tr}(\mathbf{P}) = d.$$

In order to deal with the non-differentiability of the objective function, Lerman et al. [27] iteratively solve a series of positive semi-definite programs. In contrast to models minimizing directly over $\mathbf{A} \in \mathbb{R}^{n,d}$, algorithms for minimizing REAPER or rREAPER seem to require the handling of a large matrix $\mathbf{P} \in \mathcal{S}(n)$ or, more precisely, the handling of its spectral decomposition which makes the method not practicable for high-dimensional data.

The above model requires the exact knowledge of the dimension d of the linear subspace the data will be reduced to. In this paper, we suggest to replace the strict trace constraint by a relaxed variant $\text{tr}(\mathbf{\Pi}) \leq d$ and to add the nuclear norm of $\mathbf{\Pi}$ as a regularizer which enforces the sparsity of the rank of $\mathbf{\Pi}$:

$$\min_{\mathbf{\Pi} \in \mathcal{S}(n)} \|\mathbf{\Pi}\mathbf{X} - \mathbf{X}\|_{2,1} + \alpha \|\mathbf{\Pi}\|_{\text{tr}} \tag{8}$$

subject to $\lambda_{\mathbf{\Pi}} \in \{0, 1\}^n, \text{tr}(\mathbf{\Pi}) \leq d$.

Here $\alpha > 0$ is an appropriately fixed regularization parameter.

Since (8) is again hard so solve, we use a relaxation for the eigenvalues and call the new model regularized REAPER

(rREAPER):

$$\min_{\mathbf{P} \in \mathcal{S}(n)} \|\mathbf{P}\mathbf{X} - \mathbf{X}\|_{2,1} + \alpha \|\mathbf{P}\|_{\text{tr}} \tag{9}$$

subject to $\mathbf{0}_{n,n} \leq \mathbf{P} \leq \mathbf{I}_n, \text{tr}(\mathbf{P}) \leq d$.

Finally, we project the solution of rREAPER to the set of ortho-projectors with rank not larger than d :

$$\mathcal{O}_d := \{\mathbf{\Pi} \in \mathcal{S}(n) : \lambda_{\mathbf{\Pi}} \in \mathcal{E}_d\},$$

where

$$\mathcal{E}_d := \{\lambda \in \mathbb{R}^n : \lambda \in \{0, 1\}^n, \langle \lambda, \mathbf{1}_n \rangle \leq d\}.$$

In the following, we will present a primal-dual approach to solve (9) which uses only the sparse spectral decomposition of \mathbf{P} , but not the matrix itself within the computation steps.

4 Primal-Dual Algorithm

REAPER is a convex optimization problem; so we may choose from various convex solvers. Before choosing a specific method, we study the structure of rREAPER in more details. For this purpose, we define the forward operator

$$\mathcal{X} : \mathcal{S}(n) \rightarrow \mathbb{R}^{n,N} : \mathbf{P} \mapsto \mathbf{P}\mathbf{X}$$

and rearrange (9) as

$$\min_{\mathbf{P} \in \mathcal{S}(n)} \|\mathcal{X}(\mathbf{P}) - \mathbf{X}\|_{2,1} + \alpha \mathcal{R}(\mathbf{P}), \tag{10}$$

where the regularizer $\mathcal{R} : \mathcal{S}(n) \rightarrow [0, +\infty]$ is defined by

$$\mathcal{R}(\mathbf{P}) := \|\mathbf{P}\|_{\text{tr}} + \iota_{\mathcal{C}}(\mathbf{P}), \tag{11}$$

$$\mathcal{C} := \{\mathbf{P} \in \mathcal{S}(n) : \mathbf{0}_{n,n} \leq \mathbf{P} \leq \mathbf{I}_n, \text{tr}(\mathbf{P}) \leq d\}.$$

Since \mathcal{C} is compact and convex, and since the norms $\|\cdot\|_{2,1}$ and $\|\cdot\|_{\text{tr}}$ are continuous, rREAPER has a global minimizer. This minimizer is in general not unique. Concerning the adjoint operator $\mathcal{X}^* : \mathbb{R}^{n,N} \rightarrow \mathcal{S}(n)$, we observe

$$\begin{aligned} \langle \mathcal{X}(\mathbf{P}), \mathbf{Y} \rangle &= \frac{1}{2}(\langle \mathbf{P}\mathbf{X}, \mathbf{Y} \rangle + \langle \mathbf{P}^T\mathbf{X}, \mathbf{Y} \rangle) \\ &= \frac{1}{2}(\text{tr}(\mathbf{Y}^T\mathbf{P}\mathbf{X}) + \text{tr}(\mathbf{X}^T\mathbf{P}\mathbf{Y})) \\ &= \frac{1}{2} \left\langle \mathbf{P}, \frac{1}{2}(\mathbf{X}\mathbf{Y}^T + \mathbf{Y}\mathbf{X}^T) \right\rangle. \end{aligned}$$

for all $\mathbf{P} \in \mathcal{S}(n)$ and $\mathbf{Y} \in \mathbb{R}^{n,N}$, where we exploit the symmetry of \mathbf{P} by $\mathbf{P} = 1/2(\mathbf{P} + \mathbf{P}^T)$. Thus, the adjoint is just

$$\mathcal{X}^*(\mathbf{Y}) = \frac{1}{2}(\mathbf{X}\mathbf{Y}^T + \mathbf{Y}\mathbf{X}^T).$$

The operator norm of \mathcal{X} is given by the spectral norm of $\mathbf{X} \in \mathbb{R}^{n,N}$, i.e.,

$$\|\mathcal{X}\| = \|\mathbf{X}\|_2.$$

In more detail, for $\mathbf{P} = (\mathbf{p}_1 | \dots | \mathbf{p}_n) \in \mathcal{S}(n)$, we obtain

$$\begin{aligned} \|\mathcal{X}\| &= \max_{\substack{\mathbf{P} \in \mathcal{S}(n) \\ \|\mathbf{P}\|_F \leq 1}} \|\mathbf{P}\mathbf{X}\|_F = \max_{\substack{\mathbf{P} \in \mathcal{S}(n) \\ \|\mathbf{P}\|_F \leq 1}} \left(\sum_{j=k}^n \|\mathbf{X}^T \mathbf{p}_k\|_2^2 \right)^{\frac{1}{2}} \\ &\leq \max_{\substack{\mathbf{P} \in \mathcal{S}(n) \\ \|\mathbf{P}\|_F \leq 1}} \left(\|\mathbf{X}\|_2^2 \sum_{k=1}^n \|\mathbf{p}_k\|_2^2 \right)^{\frac{1}{2}} \leq \|\mathbf{X}\|_2. \end{aligned}$$

Here the inequality becomes sharp for

$$\mathbf{P} = \mathbf{U} \operatorname{diag}((1, 0, \dots, 0)^T) \mathbf{U}^T,$$

where \mathbf{U} arises from the singular value decomposition $\mathbf{X} = \mathbf{U} \operatorname{diag}(\boldsymbol{\sigma}_X) \mathbf{V}^T$ with descending ordered singular values $\sigma_1 \geq \dots \geq \sigma_{\min\{n,N\}}$.

More generally, rREAPER is an optimization problem of the form

$$\text{minimize } F(\mathcal{A}(\mathbf{P})) + G(\mathbf{P}), \tag{12}$$

whose objective consists of two convex, lower semi-continuous functions $F = \|\cdot - \mathbf{X}\|_{2,1}$ and $G = \alpha\mathcal{R}$ and a linear mapping $\mathcal{A} = \mathcal{X}$. Since both—data fidelity and regularizer of rREAPER—are non-differentiable, gradient decent methods as well as the forward-backward splitting or the fast iterative shrinkage-thresholding algorithm (FISTA) cannot be applied. In order to exploit the structure, various saddle-point methods that do not require differentiability can be used. Examples are the alternating directions method of multipliers (ADMM) which is related to the Douglas–Rachford splitting or primal-dual algorithms, see for instance [6] and references therein. Since the iteration variables with respect to \mathbf{P} in rREAPER computed via ADMM usually have no sparse spectral decomposition and cannot be handled for high-dimensional instances, we prefer to choose the primal-dual method of Chambolle and Pock [6] with extrapolation of the primal variable to compute the minimizer of rREAPER (10).

For the general minimization problem (12), the primal-dual method consists in the iteration

$$\begin{aligned} \mathbf{Y}^{(r+1)} &:= \operatorname{prox}_{\sigma F^*}(\mathbf{Y}^{(r)} + \sigma \mathcal{A}(\bar{\mathbf{P}}^{(r)})) \\ \mathbf{P}^{(r+1)} &:= \operatorname{prox}_{\tau G}(\mathbf{P}^{(r)} - \tau \mathcal{A}^*(\mathbf{Y}^{(r+1)})) \\ \bar{\mathbf{P}}^{(r+1)} &:= (1 + \theta) \mathbf{P}^{(r+1)} - \theta \mathbf{P}^{(r)} \end{aligned}$$

with fixed parameters $\tau, \sigma > 0$ and $\theta \in [0, 1]$. The extrapolation step consisting in the calculation of $\bar{\mathbf{P}}^{(r+1)}$ is here required to ensure the convergence whenever $\tau\sigma < 1/\|\mathcal{A}\|^2$ and $\theta = 1$, where $\|\mathcal{A}\|$ denotes the operator norm of \mathcal{A} . If the functions F and G possesses more regularity like strong convexity, the parameter θ may be varied to prove an acceleration of the convergence and to guarantee certain convergence rates [5,6]. For rREAPER, the above iteration leads us to the following numerical method.

Algorithm 1 (Primal-Dual Algorithm)

INPUT: $\mathbf{X} \in \mathbb{R}^{n,N}$ $d \in \mathbb{N}$, and $\sigma, \tau > 0$ with $\sigma\tau < 1/\|\mathbf{X}\|_2^2$, and $\theta \in (0, 1]$.

INITIALIZATION: $\mathbf{P}^{(0)} = \bar{\mathbf{P}}^{(0)} = \mathbf{0}_{n,n}$, $\mathbf{Y}^{(0)} := \mathbf{0}_{n,N}$.

ITERATION:

$$\begin{aligned} \mathbf{Y}^{(r+1)} &:= \operatorname{prox}_{\sigma\|\cdot - \mathbf{X}\|_{2,1}^*}(\mathbf{Y}^{(r)} + \sigma \mathcal{X}(\bar{\mathbf{P}}^{(r)})), \\ \mathbf{P}^{(r+1)} &:= \operatorname{prox}_{\tau\alpha\mathcal{R}}(\mathbf{P}^{(r)} - \tau \mathcal{X}^*(\mathbf{Y}^{(r+1)})), \\ \bar{\mathbf{P}}^{(r+1)} &:= (1 + \theta) \mathbf{P}^{(r+1)} - \theta \mathbf{P}^{(r)} \end{aligned}$$

More generally, Chambolle and Pock [6] have proven that the sequence $\{\mathbf{P}^{(r)}\}_{r \in \mathbb{N}}$ converges to a minimizer $\hat{\mathbf{P}}$ of (10) and the sequence $\{\mathbf{Y}^{(r)}\}_{r \in \mathbb{N}}$ to a minimizer of the dual problem

$$\min_{\mathbf{Y} \in \mathbb{R}^{n,N}} \|\cdot - \mathbf{X}\|_{2,1}^*(\mathbf{Y}) + (\alpha\mathcal{R})^*(-\mathcal{X}^*(\mathbf{Y}))$$

if the Lagrangian

$$L(\mathbf{P}, \mathbf{Y}) := -\|\cdot - \mathbf{X}\|_{2,1}^*(\mathbf{Y}) + \alpha\mathcal{R}(\mathbf{P}) + \langle \mathcal{X}(\mathbf{P}), \mathbf{Y} \rangle \tag{13}$$

has a saddle-point which is, however, clear for rREAPER.

The algorithm requires the computation of the proximal mapping of the dual data fidelity and of the regularizer which we consider next.

Proposition 1 (Proximal mapping of the dual data fidelity)

For $\mathbf{x} \in \mathbb{R}^{n,N}$ and $\sigma > 0$, we have

$$\operatorname{prox}_{\sigma\|\cdot - \mathbf{X}\|_{2,1}^*} = \operatorname{proj}_{\mathfrak{B}_{2,\infty}}(\cdot - \sigma \mathbf{X}).$$

Proof Using (3) and, since $(f(\cdot - x_0))^* = f^* + \langle \cdot, x_0 \rangle$, we obtain

$$\begin{aligned} \operatorname{prox}_{\sigma\|\cdot - \mathbf{X}\|_{2,1}^*}(\mathbf{Y}) &= \operatorname{argmin}_{\mathbf{Z} \in \mathbb{R}^{n,N}} \left\{ \frac{1}{2} \|\mathbf{Z} - \mathbf{Y}\|_F^2 + \iota_{\mathfrak{B}_{2,\infty}}(\mathbf{Z}) + \sigma \langle \mathbf{Z}, \mathbf{X} \rangle \right\} \\ &= \operatorname{argmin}_{\mathbf{Z} \in \mathbb{R}^{n,N}} \left\{ \frac{1}{2} \|\mathbf{Z} - (\mathbf{Y} - \sigma \mathbf{X})\|_F^2 + \iota_{\mathfrak{B}_{2,\infty}}(\mathbf{Z}) \right\} \\ &= \operatorname{proj}_{\mathfrak{B}_{2,\infty}}(\mathbf{Y} - \sigma \mathbf{X}). \end{aligned}$$

For the maximal dimension d of the target subspace, we henceforth use the half-space

$$\mathcal{H} := \mathcal{H}(\mathbf{1}_n, d) = \{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{x}, \mathbf{1}_n \rangle \leq d\}.$$

in order to bound the trace of the primal iteration variable $\mathbf{P}^{(r)}$. Then the proximal mapping of the regularizer is given in the following proposition.

Proposition 2 (Proximal mapping of the regularizer) For $\mathbf{P} \in \mathcal{S}(n)$ with spectral decomposition $\mathbf{P} = \mathbf{U} \text{diag}(\boldsymbol{\lambda}_P) \mathbf{U}^T$ and \mathcal{R} in (11) it holds

$$\text{prox}_{\tau\alpha\mathcal{R}}(\mathbf{P}) = \mathbf{U} \text{diag}(\text{proj}_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda}_P - \tau\alpha\mathbf{1}_n)) \mathbf{U}^T.$$

Proof A symmetric matrix \mathbf{P} is in \mathcal{C} if and only if $\boldsymbol{\lambda}_P \in \mathcal{Q} \cap \mathcal{H}$. Hence, the regularizer can be written as

$$\mathcal{R}(\mathbf{P}) = \langle \boldsymbol{\lambda}_P, \mathbf{1}_n \rangle + \iota_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda}_P).$$

and

$$\begin{aligned} \text{prox}_{\tau\alpha\mathcal{R}}(\mathbf{P}) &= \underset{\mathbf{S} \in \mathcal{S}(n)}{\text{argmin}} \left\{ \frac{1}{2} \|\mathbf{S} - \mathbf{P}\|_F^2 \right. \\ &\quad \left. + \tau\alpha \langle \boldsymbol{\lambda}_S, \mathbf{1}_n \rangle + \iota_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda}_S) \right\}. \end{aligned} \tag{14}$$

By the theorem of Hoffmann and Wielandt [16, Thm. 6.3.5], we know that

$$\|\mathbf{S} - \mathbf{P}\|_F^2 \geq \|\boldsymbol{\lambda}_S - \boldsymbol{\lambda}_P\|_2^2$$

with equality if and only if \mathbf{S} possesses the same eigenspaces as \mathbf{P} . Therefore, the minimizer in (14) has to be of the form $\mathbf{S} = \mathbf{U} \text{diag}(\boldsymbol{\lambda}_S) \mathbf{U}^T$, where the columns of \mathbf{U} are the eigenvectors of \mathbf{P} . Incorporating this observation in (14), we determine the eigenvalues $\boldsymbol{\lambda}_S$ by solving the minimization problem

$$\begin{aligned} \boldsymbol{\lambda}_S &= \underset{\boldsymbol{\lambda}_S \in \mathbb{R}^n}{\text{argmin}} \left\{ \frac{1}{2} \|\boldsymbol{\lambda}_S - \boldsymbol{\lambda}_P\|_2^2 \right. \\ &\quad \left. + \tau\alpha \langle \boldsymbol{\lambda}_S, \mathbf{1}_n \rangle + \iota_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda}_S) \right\} \\ &= \underset{\boldsymbol{\lambda}_S}{\text{argmin}} \left\{ \frac{1}{2} \|\boldsymbol{\lambda}_S + \tau\alpha\mathbf{1}_n - \boldsymbol{\lambda}_P\|_2^2 + \iota_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda}_S) \right\} \\ &= \text{proj}_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda}_P - \tau\alpha\mathbf{1}_n). \end{aligned}$$

□

Alternatively to the proof, we could argue with the so-called spectral function related to \mathcal{R} which is invariant under permutations, see, e.g., [1].

By Proposition 2, the proximal mapping of the regularizer requires the projection onto the truncated hypercube. The following proposition can be found in [1, Ex. 6.32].

Proposition 3 (Projection onto the truncated hypercube)

For any $\boldsymbol{\lambda} \in \mathbb{R}^n$ and any $d \in (0, n]$, the projection to the truncated hypercube is given by

$$\text{proj}_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda}) = \begin{cases} \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda}) & \text{if } \langle \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda}), \mathbf{1}_n \rangle \leq d, \\ \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda} - \hat{t}\mathbf{1}_n) & \text{otherwise,} \end{cases}$$

where \hat{t} is the smallest root of the function

$$\varphi(t) := \langle \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda} - t\mathbf{1}_n), \mathbf{1}_n \rangle - d, \quad t \geq 0. \tag{15}$$

Due to the projection to the hypercube, see (2), only the positive components of $\boldsymbol{\lambda}$ influence its projection onto $\mathcal{Q} \cap \mathcal{H}$. More precisely, we have

$$\text{proj}_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda}) = \text{proj}_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda})_+,$$

where the function $(\cdot)_+$ is employed componentwise.

To formulate a projection algorithm, in particular, to compute the zero of φ , we study the properties of φ and derive an explicit representation. Within a different setting, the reader may also have a look at [30].

Lemma 1 (Properties of φ) For fixed $\boldsymbol{\lambda} \in \mathbb{R}^n$ with $\langle \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda}), \mathbf{1}_n \rangle > d$, the function $\varphi : [0, \infty) \rightarrow \mathbb{R}$ defined in (15) has the following properties:

- (i) φ is Lipschitz continuous.
- (ii) φ is monotone decreasing and piecewise linear. More precisely, we can construct a sequence $0 = s_0 < s_1 < s_2 < \dots < s_M$ with $M \leq 2n$ such that

$$\varphi(t) = \varphi(s_l) - k_l(t - s_l),$$

for $t \in [s_l, s_{l+1})$, $l = 0, \dots, M - 1$, where

$$k_l := |\{j \in \{1, \dots, n\} : (\boldsymbol{\lambda} - s_l\mathbf{1}_n)_j \in (0, 1]\}|.$$

Moreover, $\varphi(t) = -d$ for $t \geq s_M$.

- (iii) The smallest root \hat{t} of φ is given by

$$\hat{t} = s_m + \frac{1}{k_m} \varphi(s_m),$$

where m is the unique index such that $\varphi(s_m) > 0$ and $\varphi(s_{m+1}) \leq 0$.

Proof (i) Using the definition of φ , the Cauchy–Schwarz inequality, and the non-expansiveness of the projection, we get

$$\begin{aligned} |\varphi(t) - \varphi(s)| &= |\langle \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda} - t\mathbf{1}_n), \mathbf{1}_n \rangle - \langle \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda} - s\mathbf{1}_n), \mathbf{1}_n \rangle| \end{aligned}$$

$$\begin{aligned} &\leq \sqrt{n} \|\text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda} - t\mathbf{1}_n) - \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda} - s\mathbf{1}_n)\|_2 \\ &\leq \sqrt{n} \|(s - t)\mathbf{1}_n\|_2 = n |s - t|. \end{aligned}$$

- (ii) Starting with $s_0 = 0$, we construct s_l with $l = 1, \dots, M$ iteratively as follows: given s_l , we set $\boldsymbol{\mu} := \boldsymbol{\lambda} - s_l\mathbf{1}_n$ and choose

$$s_{l+1} := s_l + h_l, \quad h_l := \min\{s_{\text{leave}}, s_{\text{enter}}\},$$

where

$$s_{\text{leave}} := \min_j \{\mu_j : \mu_j \in (0, 1]\},$$

$$s_{\text{enter}} := \min_j \{\mu_j - 1 : \mu_j > 1\}.$$

Here we use the convention $\min \emptyset = \infty$. If both sets in the definition of s_{leave} and s_{enter} are empty, we stop the construction since all components of $\boldsymbol{\mu}$ are non-positive implying $\text{proj}_{\mathcal{Q}}(\boldsymbol{\mu}) = \mathbf{0}_n$ and thus $\varphi(s_M) = -d$ as well as $\varphi(t) = -d$ for $t \geq s_M$, where M is the index of the last computed point s_l .

Considering the projection to the hypercube \mathcal{Q} in (2), we see that the index set $\{j \in \{1, \dots, n\} : (\boldsymbol{\lambda} - t\mathbf{1}_n)_j \in (0, 1]\}$ does not change for $t \in [s_l, s_{l+1})$ and that a change appears exactly in s_{l+1} , where at least one component enters or leaves the interval $(0, 1]$. Hence, we have

$$\varphi(t) = \varphi(s_l) - k_l(t - s_l), \quad t \in [s_l, s_{l+1}).$$

Let s_M be the first value in this procedure, where all components of $\boldsymbol{\mu}$ become non-positive. Since each component in $\boldsymbol{\lambda} - t\mathbf{1}_n$ can at most one times enter or leave the interval $(0, 1]$, we know that $M < 2n$. Further, $k_l \geq 0$ shows that φ is monotone decreasing.

- (iii) By definition of φ and by the assumption $\langle \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda}), \mathbf{1}_n \rangle > d$, we have $\varphi(0) > 0$. On the other side, $\varphi(s_m) = -d$ was shown above. Consequently, the smallest zero \hat{t} of φ has to lie in the interval $[s_m, s_{m+1}]$ with $\varphi(s_m) > 0$ and $\varphi(s_{m+1}) \leq 0$ and can be computed by solving

$$\varphi(\hat{t}) = \varphi(s_m) - k_m(\hat{t} - s_m) = 0.$$

This results in $\hat{t} = s_m + \frac{1}{k_m}\varphi(s_m)$ and finishes the proof. \square

Following Proposition 3 and the previous proof, we obtain the following algorithm for the projection onto $\mathcal{Q} \cap \mathcal{H}$.

Algorithm 2 (Projection onto truncated hypercube)

INPUT: $\boldsymbol{\lambda} \in \mathbb{R}^n, d \in \mathbb{N}$.

- (i) Compute $\boldsymbol{\mu} := \text{proj}_{\mathcal{Q}}(\boldsymbol{\lambda})$ by (2).
If $\langle \boldsymbol{\mu}, \mathbf{1}_n \rangle \leq d$, then return $\hat{\boldsymbol{\lambda}} = \boldsymbol{\mu}$;
otherwise set $s := 0, \varphi := +\infty$ and $\boldsymbol{\mu} = \boldsymbol{\lambda}$.

- (ii) Repeat until $\varphi \leq 0$:

- (a) $s_{\text{old}} := s$,
- (b) $s_{\text{leave}} := \min_j \{\mu_j : \mu_j \in (0, 1]\}$,
- (c) $s_{\text{enter}} := \min_j \{\mu_j - 1 : \mu_j > 1\}$,
- (d) $s := s + \min\{s_{\text{leave}}, s_{\text{enter}}\}$,
- (e) $\boldsymbol{\mu} = \boldsymbol{\lambda} - s\mathbf{1}_n$,
- (f) $\varphi = \langle \text{proj}_{\mathcal{Q}}(\boldsymbol{\mu}), \mathbf{1}_n \rangle - d$,

- (iii) Compute

- (a) $k := |\{j \in \{1, \dots, n\} : (\boldsymbol{\lambda} - s_{\text{old}}\mathbf{1}_n)_j \in (0, 1]\}|$,
- (b) $\hat{t} = s_{\text{old}} + \frac{1}{k}\varphi(s_{\text{old}})$.

OUTPUT: $\hat{\boldsymbol{\lambda}} := \text{proj}_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda})$.

Based on the derived proximal mappings, the primal-dual Algorithm 2 to solve rREAPER (10) can be specified in matrix form as follows.

Algorithm 3 (Primal-dual rREAPER)

INPUT: $\mathbf{X} \in \mathbb{R}^{n,N}, d \in \mathbb{N}, \alpha > 0$, and $\sigma, \tau > 0$ with $\sigma\tau < 1/\|\mathbf{X}\|_2^2$, and $\theta \in [0, 1)$.

INITIATION: $\mathbf{P}^{(0)} = \bar{\mathbf{P}}^{(0)} := \mathbf{0}_{n,n}, \mathbf{Y}^{(0)} := \mathbf{0}_{n,N}$.

ITERATION:

- (i) DUAL UPDATE:

$$\mathbf{Y}^{(r+1)} := \text{proj}_{\mathfrak{B}_{2,\infty}}(\mathbf{Y}^{(r)} + \sigma(\mathcal{X}(\bar{\mathbf{P}}^{(r)}) - \mathbf{X})).$$

- (ii) PRIMAL UPDATE:

- (a) $\mathbf{U} \text{diag}(\boldsymbol{\lambda}) \mathbf{U}^T := \mathbf{P}^{(r)} - \tau \mathcal{X}^*(\mathbf{Y}^{(r+1)})$,
- (b) $\hat{\boldsymbol{\lambda}} := \text{proj}_{\mathcal{Q} \cap \mathcal{H}}(\boldsymbol{\lambda} - \tau\alpha\mathbf{1}_n)$ (Algorithm 2),
- (c) $\mathbf{P}^{(r+1)} := \mathbf{U} \text{diag}(\hat{\boldsymbol{\lambda}}) \mathbf{U}^*$.

- (iii) EXTRAPOLATION: $\bar{\mathbf{P}}^{(r+1)} := (1 + \theta) \mathbf{P}^{(r+1)} - \theta \mathbf{P}^{(r)}$.

OUTPUT: $\hat{\mathbf{P}}$ (Solution of rREAPER (10)).

5 Matrix-Free Realization

Solving rREAPER with the primal-dual Algorithm 3 is possible if the dimension of the surrounding space \mathbb{R}^n is moderate which is often not the case in image processing tasks. While the dual variable $\mathbf{Y} \in \mathbb{R}^{n,N}$ matches the dimension of the data, the primal variable \mathbf{P} is in $S(n)$ instead of $\mathbb{R}^{n,d}, d \ll n$. How can the primal-dual iteration be realized in the case $n \gg d$ though the primal variable cannot be hold in memory and the required eigenvalue decomposition cannot be computed in a reasonable amount of time?

Here the nuclear norm in rREAPER that promotes low-rank matrices comes to our aid. Our main idea to derive a practical implementation of the primal-dual iteration is thus based on

The stopping criteria of the thick-restarted Lanczos process is here deduced from the fact that the chosen Ritz pairs fulfill the equation

$$P \check{u}_k = \check{\lambda}_k \check{u}_k + \check{y}_{k_{\max},k} \check{r}_{k_{\max}},$$

where $\check{r}_{k_{\max}}$ is the last residuum vector of the previous iteration [50]. Consequently, the absolute error of the chosen Ritz pairs is given by

$$\|P \check{u}_k - \check{\lambda}_k \check{u}_k\|_2 = |\check{y}_{k_{\max},k}| \|\check{r}_{k_{\max}}\|_2 = \check{y}_{k_{\max}} |\check{y}_{k_{\max},k}|.$$

Usually, the absolute value of the leading Ritz value is a good approximation of the required spectral norm $\|P\|$ to estimate the current relative error.

Remark 3 (Power method) Besides the Krylov subspace methods like the discussed Lanczos process with thick-restarting, the required eigenpairs may be calculated using the power method. Here the main idea is to start with an orthogonal matrix $U^{(0)} := [u_1^{(0)} | \dots | u_{\ell_{\text{fix}}}^{(0)}] \in \mathbb{R}^{n \times \ell_{\text{fix}}}$ representing an ℓ_{fix} -dimensional subspace $\mathcal{U}^{(0)}$ and to successively compute $\mathcal{U}^{(r)} := P(\mathcal{U}^{(r-1)})$ again represented by an orthogonal matrix $U^{(r)}$. The convergence of the so-called orthogonal iteration may be improved by constructing $U^{(r)}$ using the eigenvalue decomposition of P restricted to $\mathcal{U}^{(r)}$. This approach goes back to [43] and consists in the iteration

$$E^{(r)} R^{(r)} = P U^{(r-1)}, \tag{18}$$

$$E^{(r)T} P E^{(r)} = Y^{(r)} \Lambda^{(r)} Y^{(r)T} \tag{19}$$

$$U^{(r)} := E^{(r)} Y^{(r)}, \tag{20}$$

where the first step consists in the QR decomposition of $P U^{(r-1)}$ into an orthogonal matrix $E^{(r)}$ and an upper triangular matrix $R^{(r)}$ and the second in the eigenvalue decomposition of $E^{(r)T} P E^{(r)}$.

If none of the columns of $U^{(0)}$ is orthogonal to the ℓ_{fix} leading eigenvectors of P , and if the ℓ_{fix} leading eigenvalues are well-defined—both in absolute value, then the columns of $U^{(r)}$ and the eigenvalues in $\Lambda^{(r)}$ converge to the leading eigenpairs of P , see [13,43].

Although there is no convergence guarantee for the restarted Lanczos process, it usually enjoys much faster convergence than the orthogonal iteration (18)–(20) for high-dimensional problems. Moreover, the chosen restarting technique can be easily adapted to compute the proximal mapping of the regularizer as discussed in the next section.

5.2 Matrix-Free Primal Update

The thick-restarted Lanczos method allows us to compute the leading absolute eigenvalues and their corresponding eigenvectors in a matrix-free manner using only the action of the

considered matrix. In our primal-dual method for rREAPER, we need the action of $P^{(r)} - \tau \mathcal{L}^*(Y^{(r+1)})$. Incorporating the low-rank representation (16), we see that this can be rewritten as

$$e \in \mathbb{R}^n \mapsto \left\{ \sum_{k=1}^{d_r} \lambda_k^{(r)} \langle e, u_k^{(r)} \rangle u_k^{(r)} \right\} - \frac{\tau}{2} \left\{ Y^{(r+1)} [X^T e] + X [(Y^{(r+1)})^T e] \right\}.$$

For the evaluation of the primal proximal mapping, we first compute the eigenvalue decomposition of $P^{(r)} - \tau \mathcal{L}^*(Y^{(r+1)})$, next shift the eigenvalues, and finally project them to the truncated hypercube $\mathcal{D} \cap \mathcal{H}$, see Algorithm 3. Since the projection onto $\mathcal{D} \cap \mathcal{H}$ is independent of negative eigenvalues, see note after Proposition 3, it is thus sufficient to compute only the eigenpairs with eigenvalue larger than $\alpha\tau$.

For the numerical implementation, we compute the relevant eigenpairs with the thick-restarted Lanczos method. In the course of this, we are confronted with the issue that we actually do not know how many eigenpairs has to be computed. To reduce the overhead of Algorithm 4 as much as possible, the parameters ℓ_{fix} and k_{\max} can be easily adapted between the restarts. Further, the computation of strongly negative eigenvalues can be avoided by an eigenvalue shift, i.e., actually compute the eigenpairs of $P^{(r)} - \tau \mathcal{L}^*(Y^{(r+1)}) + \nu I$ with $\mu \geq 0$, where the required action has the form

$$e \in \mathbb{R}^n \mapsto \left\{ \sum_{k=1}^{d_r} \lambda_k^{(r)} \langle e, u_k^{(r)} \rangle u_k^{(r)} \right\} - \frac{\tau}{2} \left\{ Y^{(r+1)} [X^T e] + X [(Y^{(r+1)})^T e] \right\} + \nu e. \tag{21}$$

Essentially, we may thus implement the primal proximation in the following manner.

Algorithm 5 (Matrix-free primal proximation)

INPUT: $P^{(r)} \in \mathcal{S}(n)$, $Y^{(r+1)} \in \mathbb{R}^{n,N}$, $d > 0$, $\tau > 0$, $\alpha > 0$.

- (i) THICK-RESTARTED LANCZOS METHOD:
 - Setting $\nu := 0$, $\ell_{\text{fix}} := \text{rank}(P^{(r)})$, $k_{\max} := \min\{2\ell_{\text{fix}}, n\}$, run Algorithm 4 with action (21). Between restarts, check convergence and update parameters:
 - (a) If $\gamma_{k_{\max}} |y_{k_{\max},k}| \leq \delta \|P^{(r)} - \tau \mathcal{L}^*(Y^{(r+1)}) + \nu I\|$ for $k = 1, \dots, m + 1$, and if $\lambda_1 \geq \dots \geq \lambda_m \geq \alpha\tau + \nu > \lambda_{m+1}$, then return $U := [u_1 | \dots | u_m]$ and $\Lambda := \text{diag}(\lambda_1 - \nu, \dots, \lambda_m - \nu)$.
 - (b) If $\lambda_{\ell_{\text{fix}}} > \alpha\tau + \nu$, then increase ℓ_{fix} , k_{\max} so that $\ell_{\text{fix}} < k_{\max} \leq n$.

- (c) Set $\xi := \max\{[\lambda_1]_-, \dots, [\lambda_{k_{\max}}]_-\}$ and $\nu := \nu + \xi$.
Restart with $(\lambda_k + \xi, \mathbf{u}_k), k = 1, \dots, \ell_{\text{fix}}$.

- (ii) PROJECTION ONTO $\mathcal{Q} \cap \mathcal{H}$:
Run Algorithm 2 on

$$\lambda := (\lambda_1 - \alpha\tau, \dots, \lambda_m - \alpha\tau, 0, \dots, 0)^T \in \mathbb{R}^n$$

to get $\hat{\lambda} := \text{proj}_{\mathcal{Q} \cap \mathcal{H}}(\lambda)$.

- (iii) NEW LOW-RANK REPRESENTATION:
Determine $d_{r+1} := \max\{k : \hat{\lambda}_k > 0\}$ and return

$$\mathbf{P}^{(r+1)} := \sum_{k=1}^{d_{r+1}} \hat{\lambda}_k \mathbf{u}_k \mathbf{u}_k^T.$$

OUTPUT: $\mathbf{P}^{(r+1)} := \sum_{k=1}^{d_{r+1}} \lambda_k^{(r+1)} \mathbf{u}_k^{(r+1)} (\mathbf{u}_k^{(r+1)})^T$.

Remark 4 If the matrix $\mathbf{P}^{(r)} - \tau \mathcal{X}^*(\mathbf{Y}^{(r+1)})$ does not possess any eigenvalues greater than $\alpha\tau$, then the Lanczos process stops in step (i.a) with $m = 0$. Since the projection to the truncated hypercube is then the zero vector again, the new iteration $\mathbf{P}^{(r+1)}$ can be represented by an empty low-rank representation, i.e., $d_{r+1} = 0$.

5.3 Matrix-Free Dual Update

Compared with the primal update, the derivation of the matrix-free dual update is more straightforward. First, the matrix

$$\mathbf{Z} := \mathbf{Y}^{(r)} + \sigma [\mathcal{X}((1 + \theta) \mathbf{P}^{(r)} - \theta \mathbf{P}^{(r-1)}) - \mathbf{X}]$$

is computed, where the over-relaxation $\bar{\mathbf{P}}^{(r)} := (1 + \theta) \mathbf{P}^{(r)} - \theta \mathbf{P}^{(r-1)}$ is already plugged in. The low-rank representations of $\mathbf{P}^{(r)}$ and $\mathbf{P}^{(r-1)}$ similar to (16) can efficiently be incorporated by calculating the matrix $\mathbf{Z} := [\mathbf{z}_1 | \dots | \mathbf{z}_N]$ column by column. This way of handling the forward operator \mathcal{X} nicely matches with the projection of the columns \mathbf{z}_k to the Euclidean unit ball in the second step. Writing the matrix $\mathbf{Y}^{(r)} := [\mathbf{y}_1^{(r)} | \dots | \mathbf{y}_N^{(r)}]$ column by column too, we obtain the following numerical method.

Algorithm 6 (Matrix-free dual proximation)

INPUT: $\mathbf{Y}^{(r)} \in \mathbb{R}^{n,N}, \mathbf{P}^{(r)} \in S(n), \mathbf{P}^{(r-1)} \in S(n), \sigma > 0, \theta \in (0, 1]$.

- (i) For $k = 1, \dots, N$, compute

$$\begin{aligned} \mathbf{z}_k := & \mathbf{y}_k^{(r)} + \sigma (1 + \theta) \left\{ \sum_{\ell=1}^{d_r} \lambda_\ell^{(r)} \langle \mathbf{x}_k, \mathbf{u}_\ell^{(r)} \rangle \mathbf{u}_\ell^{(r)} \right\} \\ & - \sigma \theta \left\{ \sum_{\ell=1}^{d_{r-1}} \lambda_\ell^{(r-1)} \langle \mathbf{x}_k, \mathbf{u}_\ell^{(r-1)} \rangle \mathbf{u}_\ell^{(r-1)} \right\} - \sigma \mathbf{x}_k. \end{aligned}$$

- (ii) For $k = 1, \dots, N$, compute $\mathbf{z}_k := \mathbf{z}_k / (1 + \|\mathbf{z}_k\|_2 - 1)_+$.
- (iii) Return $\mathbf{Y}^{(r+1)} := [\mathbf{z}_1 | \dots | \mathbf{z}_N]$.

OUTPUT: $\mathbf{Y}^{(r+1)}$

5.4 Matrix-Free Projection onto the Orthoprojectors

With the matrix-free implementations of the primal and dual proximal mappings, we are already able to solve rREAPER (9) numerically. Before summarizing the compound algorithm, we briefly discuss the last needed component to tackle the robust PCA problem (8). The final step is to project the solution $\hat{\mathbf{P}}$ of rREAPER onto the set of orthoprojectors with rank not larger than d :

$$\mathcal{O}_d := \{\mathbf{\Pi} \in \mathcal{S}(n) : \lambda_{\mathbf{\Pi}} \in \mathcal{E}_d\},$$

where

$$\mathcal{E}_d := \{\lambda \in \mathbb{R}^n : \lambda \in \{0, 1\}^n, \langle \lambda, \mathbf{1}_n \rangle \leq d\}.$$

We may calculate the projection explicitly in the following manner.

Proposition 4 (Projection onto the orthoprojectors) For $\mathbf{P} \in S(n)$ with eigenvalue decomposition $\mathbf{P} = \mathbf{U} \text{diag}(\lambda_{\mathbf{P}}) \mathbf{U}^T$, and for every $1 \leq p \leq \infty$, the projection onto \mathcal{O}_d with respect to the Schatten p -norm is given by

$$\text{proj}_{\mathcal{O}_d}(\mathbf{P}) = \mathbf{U} \text{diag}(\text{proj}_{\mathcal{E}_d}(\lambda)) \mathbf{U}^T.$$

Proof The key ingredient to prove this statement is the theorem of Lidskii–Mirsky–Wielandt, see for instance [29]. Using this theorem to estimate the Schatten p -Norm, we obtain

$$\min_{\mathbf{\Pi} \in \mathcal{O}_d} \|\mathbf{P} - \mathbf{\Pi}\|_{S_p} \geq \min_{\lambda_{\mathbf{\Pi}} \in \mathcal{E}_d} \|\lambda_{\mathbf{P}} - \lambda_{\mathbf{\Pi}}\|_p, \tag{22}$$

where we have equality if $\mathbf{\Pi}$ has the same eigenvectors as \mathbf{P} . Recall that the eigenvalues in $\lambda_{\mathbf{P}}$ appear in descending order. The right-hand side of (22) thus becomes minimal if we choose the eigenvalues of $\mathbf{\Pi}$ for $k = 1, \dots, d$ as

$$\hat{\lambda}_{\mathbf{\Pi},k} := \begin{cases} 1 & \text{if } \lambda_{\mathbf{P},k} \geq \frac{1}{2}, \\ 0 & \text{if } \lambda_{\mathbf{P},k} < \frac{1}{2}, \end{cases}$$

and set $\hat{\lambda}_{\mathbf{\Pi},k} := 0$ for $k = d + 1, \dots, n$. This is exactly the projection onto \mathcal{E}_d . □

Because of the low-rank representation

$$\mathbf{P}^{(r)} = \sum_{k=1}^{d_r} \lambda_k^{(r)} \mathbf{u}_k^{(r)} (\mathbf{u}_k^{(r)})^T$$

of the primal variable, the construction of the orthoprojector $\hat{\Pi} \in \mathcal{O}_d$ is here especially simple.

Algorithm 7 (Matrix-free projection onto orthoprojectors)

INPUT: $\mathbf{P} = \sum_{k=1}^{\kappa} \lambda_k \mathbf{u}_k \mathbf{u}_k^T \in \mathcal{S}(n)$, $d \in \mathbb{N}$.

- (i) PROJECTION ONTO \mathcal{O}_d :
Determine $s := \max\{k : \lambda_k \geq 1/2, k \leq \min\{d, \kappa\}\}$.
- (ii) MATRIX-FREE PRESENTATION:
Return $\Pi = \sum_{k=1}^s \mathbf{u}_k \mathbf{u}_k^T$.

OUTPUT: $\Pi = \text{proj}_{\mathcal{O}_d}(\mathbf{P})$.

5.5 Matrix-Free Robust PCA by RREAPER

Combining the matrix-free implementations of the primal and dual proximal mappings, we finally obtain a primal-dual method to solve rREAPER (9) without evaluating the primal variable $\mathbf{P}^{(r)}$ representing the relaxed orthoprojector explicitly.

Algorithm 8 (Matrix-free robust PCA)

INPUT: $\mathbf{X} \in \mathbb{R}^{n,N}$, $d \in \mathbb{N}$, $\alpha > 0$, and $\sigma, \tau > 0$ with $\sigma\tau < 1/\|\mathbf{X}\|_2^2$, and $\theta \in (0, 1]$.

INITIATION: $\mathbf{P}^{(0)} = \tilde{\mathbf{P}}^{(0)} := \mathbf{0} \in \mathbb{R}^{n,n}$, $\mathbf{Y}^{(0)} := \mathbf{0} \in \mathbb{R}^{n,N}$.

ITERATION:

- (i) DUAL UPDATE: Compute $\mathbf{Y}^{(r+1)}$ with Algorithm 6.
- (ii) PRIMAL UPDATE: Compute $\mathbf{P}^{(r+1)}$ with Algorithm 5.

PROJECTION: Compute $\hat{\Pi}$ with Algorithm 7.

OUTPUT: $\hat{\Pi}$ (Orthoprojector onto recovered subspace).

Due to the convexity of rREAPER, every minimizer of (9) is global. Thus, our proposed algorithm cannot get stuck in local minima. Since the Lagrangian (13) has a saddle-point by [40, Cor. 28.2.2], the convergence of the underlying primal-dual method to a saddle-point $(\hat{\mathbf{P}}, \hat{\mathbf{Y}})$ is guaranteed for parameter choices $\sigma\tau < 1/\|\mathbf{X}\|_2^2$ and $\theta = 1$ as proven in [5, Thm. 1]. Finally, the primal variable $\hat{\mathbf{P}}$ minimizes rREAPER by [40, Thm. 28.3].

However, using the thick-restarted Lanczos process to evaluate the proximal mapping of the regularizer in Proposition 2, we introduce a systematic error since the required eigenvalue decompositions are only computed approximately. Fortunately, we may control this error by choosing the accuracy δ in Algorithm 4 appropriately small. If the accuracies becomes higher during the primal-dual iteration, we will see that the underlying primal-dual iteration converges nevertheless.

Theorem 1 (Convergence of Algorithm 8) Let

$$\mathbf{Q}^{(r)} := \mathbf{P}^{(r-1)} - \tau \mathcal{X}^*(\mathbf{Y}^{(r)}) = \mathbf{U} \text{diag}(\boldsymbol{\lambda}) \mathbf{U}^T$$

be the argument in the r th primal-dual iteration of Algorithm 8, and $\tilde{\mathbf{Q}}^{(r)}$ the computed low-rank approximation of the leading positive eigenvalues greater than $\tau\alpha$ by the thick-restarted Lanczos process. Define the approximation error by

$$E_r := \|(\tilde{\mathbf{Q}}^{(r)} - \tau\alpha \mathbf{I}_n)_{\geq \mathbf{0}} - (\mathbf{Q}^{(r)} - \tau\alpha \mathbf{I}_n)_{\geq \mathbf{0}}\|_F,$$

where $(\cdot)_{\geq \mathbf{0}}$ denotes the projection to the positive-definite cone. Then, for $\sigma\tau < 1/\|\mathbf{X}\|_2^2$ and $\theta = 1$, the primal-dual iteration in Algorithm 8 converges to a global minimizer of rREAPER whenever $\sum_{r=0}^{\infty} E_r^{1/2} (\|\mathbf{Q}^{(r)}\|_F + \sqrt{d})^{1/2} < \infty$.

Proof The key idea of the proof is to show that the approximations $\tilde{\mathbf{P}}^{(r)} = \text{prox}_{\tau\alpha \mathcal{R}}(\tilde{\mathbf{Q}}^{(r)})$ are so-called type-one approximations of the proximal points $\mathbf{P}^{(r)}$ in the second step of Algorithm 1. For this, we have to compare the objective

$$F(\mathbf{P}) := \tau\alpha \|\mathbf{P}\|_{\text{tr}} + \iota_{\mathcal{C}}(\mathbf{P}) + \frac{1}{2} \|\mathbf{Q}^{(r)} - \mathbf{P}\|_F^2$$

of $\text{prox}_{\tau\alpha \mathcal{R}}$ at the points $\tilde{\mathbf{P}}^{(r)}$ and $\mathbf{P}^{(r)}$. By the triangle inequality, we obtain

$$\begin{aligned} & F(\tilde{\mathbf{P}}^{(r)}) - F(\mathbf{P}^{(r)}) \\ &= \tau\alpha \|\tilde{\mathbf{P}}^{(r)}\|_{\text{tr}} + \frac{1}{2} \|\mathbf{Q}^{(r)} - \tilde{\mathbf{P}}^{(r)}\|_F^2 \\ &\quad - \tau\alpha \|\mathbf{P}^{(r)}\|_{\text{tr}} - \frac{1}{2} \|\mathbf{Q}^{(r)} - \mathbf{P}^{(r)}\|_F^2 \\ &\leq \tau\alpha \|\tilde{\mathbf{P}}^{(r)} - \mathbf{P}^{(r)}\|_{\text{tr}} \\ &\quad + \frac{1}{2} (\|\tilde{\mathbf{P}}^{(r)} - \mathbf{P}^{(r)}\|_F + \|\mathbf{Q}^{(r)} - \mathbf{P}^{(r)}\|_F)^2 \\ &\quad - \frac{1}{2} \|\mathbf{Q}^{(r)} - \mathbf{P}^{(r)}\|_F^2 \\ &\leq \tau\alpha \|\tilde{\mathbf{P}}^{(r)} - \mathbf{P}^{(r)}\|_{\text{tr}} + \frac{1}{2} \|\tilde{\mathbf{P}}^{(r)} - \mathbf{P}^{(r)}\|_F^2 \\ &\quad + \|\tilde{\mathbf{P}}^{(r)} - \mathbf{P}^{(r)}\|_F \|\mathbf{Q}^{(r)} - \mathbf{P}^{(r)}\|_F \\ &\leq \tau\alpha \sqrt{n} \|\tilde{\mathbf{P}}^{(r)} - \mathbf{P}^{(r)}\|_F + \frac{1}{2} \|\tilde{\mathbf{P}}^{(r)} - \mathbf{P}^{(r)}\|_F^2 \\ &\quad + \|\tilde{\mathbf{P}}^{(r)} - \mathbf{P}^{(r)}\|_F \|\mathbf{Q}^{(r)} - \mathbf{P}^{(r)}\|_F. \end{aligned}$$

Using

$$\begin{aligned} \mathbf{P}^{(r)} &= \text{prox}_{\tau\alpha \mathcal{R}}(\mathbf{Q}^{(r)}) = \text{proj}_{\mathcal{C}}(\mathbf{Q}^{(r)} - \tau\alpha \mathbf{I}_n) \\ &= \text{proj}_{\mathcal{C}}((\mathbf{Q}^{(r)} - \tau\alpha \mathbf{I}_n)_{\geq \mathbf{0}}), \end{aligned}$$

the non-expansiveness of the projection and assuming without loss of generality that $E_r \leq 1$, we conclude

$$\begin{aligned} & F(\tilde{\mathbf{P}}^{(r)}) - F(\mathbf{P}^{(r)}) \\ &\leq \tau\alpha \sqrt{n} E_r + \frac{1}{2} E_r^2 + E_r (\|\mathbf{Q}^{(r)}\|_F + \|\mathbf{P}^{(r)}\|_F) \\ &\leq C E_r (\|\mathbf{Q}^{(r)}\|_F + \sqrt{d}), \quad C > 0. \end{aligned}$$

Thus, $\tilde{\mathbf{P}}^{(r)}$ fulfills the definition of the type-one approximation with precision $CE_r(\|\mathbf{Q}^{(r)}\|_F + \sqrt{d})$, see [39, p. 385]. Since the square roots of the precisions are summable, [39, Thm. 2] ensures the convergence of the inexactly performed primal-dual iteration to a saddle-point of the Lagrangian in (13) and thus to a solution $\hat{\mathbf{P}}$ of rREAPER. \square

6 Performance Analysis

Inspired by ideas of Lerman et al. [27], we examine the performance analysis of rREAPER. To this end, we assume that the ‘ideal’ subspace L of the given data $\mathbf{x}_k \in \mathbb{R}^n, k = 1, \dots, N$ has dimension $d_L \leq d$. As in [27], we determine the best fit of the data by two measures: the first one is the *distance of the data from the subspace*

$$\mathcal{R}_L = \mathcal{R}_L(\mathbf{X}) := \|(\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{X}\|_{2,1},$$

where $\mathbf{\Pi}_L$ denotes the orthogonal projector onto L . For the second measure, we assume that the projected data $\{\mathbf{\Pi}\mathbf{x}_k : k = 1, \dots, N\}, N \geq d_L$, form a *frame* in L meaning that there exist constants $0 < c_L \leq C_L < \infty$ such that

$$c_L \leq \sum_{k=1}^N |\langle \mathbf{u}, \mathbf{\Pi}_L \mathbf{x}_k \rangle|^2 = \sum_{k=1}^N |\langle \mathbf{u}, \mathbf{x}_k \rangle|^2 \leq C_L$$

for all $\mathbf{u} \in L$ with $\|\mathbf{u}\|_2 = 1$. In order to recover the entire subspace L , the data have obviously to cover each direction in L with sufficiently many data points. This well-localization of the data is measured by the *permeance statistic*

$$\mathcal{P}_L = \mathcal{P}_L(\mathbf{X}) := \min_{\substack{\mathbf{u} \in L \\ \|\mathbf{u}\|_2 = 1}} \sum_{k=1}^N |\langle \mathbf{u}, \mathbf{x}_k \rangle| \tag{23}$$

which can be seen as ℓ_1 counterpart of the lower frame bound. Clearly, \mathcal{P}_L becomes large if all direction in L are uniformly covered by the data. The lower frame bound and the permeance statistic come into the play in the following lemma, compared with [27, Sect. A2.3].

Lemma 2 *Let $\mathbf{\Pi}_L$ be the orthogonal projector onto a subspace L of \mathbb{R}^n of dimension d_L and $\mathbf{x}_k \in \mathbb{R}^n, k = 1, \dots, N, N \geq d_L$ which form the columns of the matrix \mathbf{X} . Then, for any $\mathbf{A} \in \mathbb{R}^{n,n}$, the following relations hold true:*

$$\|\mathbf{A}\mathbf{\Pi}_L\mathbf{X}\|_{2,2} \geq c_L \|\mathbf{A}\mathbf{\Pi}_L\|_F, \tag{24}$$

and

$$\|\mathbf{A}\mathbf{\Pi}_L\mathbf{X}\|_{2,1} \geq \mathcal{P}_L \|\mathbf{A}\mathbf{\Pi}_L\|_F \geq \frac{1}{\sqrt{d_L}} \mathcal{P}_L \|\mathbf{A}\mathbf{\Pi}_L\|_{\text{tr}}. \tag{25}$$

Proof We restrict our attention to (25). The relation (24) follows similar lines. Let $\mathbf{A}\mathbf{\Pi}_L$ have the singular value decomposition $\mathbf{A}\mathbf{\Pi}_L = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where the singular values $\sigma_k, k = 1, \dots, n$ are in descending order and $\sigma_{d_L+1} = \dots = \sigma_n = 0$ and we can arrange \mathbf{V} such that the transpose of the first d_L rows of \mathbf{V} belong to L . Then it holds

$$\|\mathbf{A}\mathbf{\Pi}_L\|_F^2 = \sum_{k=1}^{d_L} \sigma_k^2.$$

Using orthogonality of \mathbf{U} and concavity of the square root function, we obtain

$$\begin{aligned} \|\mathbf{A}\mathbf{\Pi}_L\mathbf{X}\|_{2,1} &= \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{X}\|_{2,1} = \|\mathbf{\Sigma}\mathbf{V}^T\mathbf{X}\|_{2,1} \\ &= \sum_{k=1}^N \left(\sum_{j=1}^{d_L} \sigma_j^2 \langle \mathbf{v}_j, \mathbf{x}_k \rangle^2 \right)^{\frac{1}{2}} \\ &= \|\mathbf{A}\mathbf{\Pi}_L\|_F \sum_{k=1}^N \left(\sum_{j=1}^{d_L} \frac{\sigma_j^2}{\|\mathbf{A}\mathbf{\Pi}_L\|_F^2} \langle \mathbf{v}_j, \mathbf{x}_k \rangle^2 \right)^{\frac{1}{2}} \\ &\geq \|\mathbf{A}\mathbf{\Pi}_L\|_F \sum_{k=1}^N \sum_{j=1}^{d_L} \frac{\sigma_j^2}{\|\mathbf{A}\mathbf{\Pi}_L\|_F^2} |\langle \mathbf{v}_j, \mathbf{x}_k \rangle| \\ &= \|\mathbf{A}\mathbf{\Pi}_L\|_F \sum_{j=1}^{d_L} \frac{\sigma_j^2}{\|\mathbf{A}\mathbf{\Pi}_L\|_F^2} \sum_{k=1}^N |\langle \mathbf{v}_j, \mathbf{x}_k \rangle| \\ &\geq \mathcal{P}_L \|\mathbf{A}\mathbf{\Pi}_L\|_F. \end{aligned}$$

The second estimate in (25) follows by (1). \square

Now we can estimate the reconstruction error of rREAPER.

Theorem 2 *Let $\mathbf{\Pi}_L$ be the orthogonal projector onto a subspace L of \mathbb{R}^n of dimension d_L and $\mathbf{x}_k \in \mathbb{R}^n, k = 1, \dots, N, N \geq d_L$ such that their projections onto L form a frame of L . Define \mathcal{P}_L by (23) and set $\gamma_L := \frac{1}{2\sqrt{2d_L}} \mathcal{P}_L$. Let $\hat{\mathbf{P}}$ be the solution of (9) and $\hat{\mathbf{\Pi}}$ the projection of $\hat{\mathbf{P}}$ onto \mathcal{O}_d . Then, for $\alpha \leq 2\gamma_L$ and $d \geq d_L$, the reconstruction error is bounded by*

$$\|\hat{\mathbf{\Pi}} - \mathbf{\Pi}_L\|_{\text{tr}} \leq \min \left\{ \frac{6\mathcal{R}_L}{\gamma_L - |\gamma_L - \alpha|}, \frac{6\mathcal{R}_L + 2d|\alpha - \gamma_L|}{\gamma_L} \right\}. \tag{26}$$

Proof Since $\hat{\mathbf{P}}$ is a minimizer of (9), we obtain

$$\begin{aligned} 0 &\leq \|(\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{X}\|_{2,1} - \|(\mathbf{I}_n - \hat{\mathbf{P}})\mathbf{X}\|_{2,1} \\ &\quad + \alpha (\|\mathbf{\Pi}_L\|_{\text{tr}} - \|\hat{\mathbf{P}}\|_{\text{tr}}) \tag{27} \\ &= \mathcal{R}_L - \|(\mathbf{I}_n - \hat{\mathbf{P}})\mathbf{X}\|_{2,1} + \alpha (d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}) \\ &\leq \mathcal{R}_L + \|(\mathbf{I}_n - \hat{\mathbf{P}})(\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{X}\|_{2,1} \end{aligned}$$

$$\begin{aligned}
 & - \|(\mathbf{I}_n - \hat{\mathbf{P}})\mathbf{\Pi}_L\mathbf{X}\|_{2,1} + \alpha(d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}) \\
 = & \mathcal{R}_L + \|(\mathbf{I}_n - (\hat{\mathbf{P}} - \mathbf{\Pi}_L))(\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{X}\|_{2,1} \\
 & - \|(\mathbf{I}_n - \hat{\mathbf{P}})\mathbf{\Pi}_L\mathbf{X}\|_{2,1} + \alpha(d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}) \\
 \leq & (1 + \|\mathbf{I}_n - (\hat{\mathbf{P}} - \mathbf{\Pi}_L)\|_2)\mathcal{R}_L - \|(\mathbf{I}_n - \hat{\mathbf{P}})\mathbf{\Pi}_L\mathbf{X}\|_{2,1} \\
 & + \alpha(d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}) \tag{28}
 \end{aligned}$$

It remains to estimate

$$\|(\mathbf{I}_n - \hat{\mathbf{P}})\mathbf{\Pi}_L\mathbf{X}\|_{2,1} = \|(\hat{\mathbf{P}} - \mathbf{\Pi}_L)\mathbf{\Pi}_L\mathbf{X}\|_{2,1}$$

from below. To this end, we decompose $\mathbf{\Delta} := \hat{\mathbf{P}} - \mathbf{\Pi}_L$ as

$$\begin{aligned}
 \mathbf{\Delta} = & \underbrace{\mathbf{\Pi}_L\mathbf{\Delta}\mathbf{\Pi}_L}_{\mathbf{\Delta}_1} + \underbrace{(\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{\Delta}\mathbf{\Pi}_L}_{\mathbf{\Delta}_2} + \underbrace{\mathbf{\Pi}_L\mathbf{\Delta}(\mathbf{I}_n - \mathbf{\Pi}_L)}_{\mathbf{\Delta}_2^T} \\
 & + \underbrace{(\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{\Delta}(\mathbf{I}_n - \mathbf{\Pi}_L)}_{\mathbf{\Delta}_3}.
 \end{aligned}$$

Since $\mathbf{0}_{n,n} \leq \hat{\mathbf{P}} \leq \mathbf{I}_n$, we obtain by conjugation with $\mathbf{\Pi}_L$, resp. $\mathbf{I}_n - \mathbf{\Pi}_L$ that $\mathbf{\Delta}_1 \leq \mathbf{0}_{n,n}$ and $\mathbf{0}_{n,n} \leq \mathbf{\Delta}_3$, so that $\|\mathbf{\Delta}_1\|_{\text{tr}} = -\text{tr } \mathbf{\Delta}_1$ and $\|\mathbf{\Delta}_3\|_{\text{tr}} = \text{tr } \mathbf{\Delta}_3$. Then we conclude

$$\begin{aligned}
 \text{tr}(\mathbf{\Delta}) &= \text{tr}(\hat{\mathbf{P}}) - d_L = \text{tr } \mathbf{\Delta}_1 + 2\text{tr } \mathbf{\Delta}_2 + \text{tr } \mathbf{\Delta}_3 \\
 &= \text{tr } \mathbf{\Delta}_1 + \text{tr } \mathbf{\Delta}_3 = \|\mathbf{\Delta}_3\|_{\text{tr}} - \|\mathbf{\Delta}_1\|_{\text{tr}},
 \end{aligned}$$

which implies

$$\begin{aligned}
 \|\mathbf{\Delta}\|_{\text{tr}} &\leq \|\mathbf{\Delta}_1\|_{\text{tr}} + \|\mathbf{\Delta}_2\|_{\text{tr}} + \|\mathbf{\Delta}_2^T\|_{\text{tr}} + \|\mathbf{\Delta}_3\|_{\text{tr}} \\
 &= 2\|\mathbf{\Delta}_1\|_{\text{tr}} + 2\|\mathbf{\Delta}_2\|_{\text{tr}} + \text{tr}(\hat{\mathbf{P}}) - d_L.
 \end{aligned}$$

Now we can estimate the last summand by (1) and Lemma 2 as

$$\begin{aligned}
 \|\mathbf{\Delta}\mathbf{\Pi}_L\mathbf{X}\|_{2,1} &= \|\mathbf{\Pi}_L\mathbf{\Delta}\mathbf{\Pi}_L\mathbf{X} + (\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{\Delta}\mathbf{\Pi}_L\mathbf{X}\|_{2,1} \\
 &= \sum_{k=1}^N \left(\|\mathbf{\Delta}_1\mathbf{x}_k\|_2^2 + \|\mathbf{\Delta}_2\mathbf{x}_k\|_2^2 \right)^{\frac{1}{2}} \\
 &\geq \frac{1}{\sqrt{2}} \sum_{k=1}^N (\|\mathbf{\Delta}_1\mathbf{x}_k\|_2 + \|\mathbf{\Delta}_2\mathbf{x}_k\|_2) \\
 &\geq \frac{1}{\sqrt{2d_L}} \mathcal{P}_L(\|\mathbf{\Delta}_1\|_{\text{tr}} + \|\mathbf{\Delta}_2\|_{\text{tr}}) \\
 &\geq \frac{1}{2\sqrt{2d_L}} \mathcal{P}_L(\|\mathbf{\Delta}\|_{\text{tr}} + d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}). \tag{29}
 \end{aligned}$$

By (28) and (29), we obtain

$$\begin{aligned}
 0 \leq & (1 + \|\mathbf{I}_n - (\hat{\mathbf{P}} - \mathbf{\Pi}_L)\|_2)\mathcal{R}_L \\
 & + (\alpha - \gamma_L)(d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}) - \gamma_L\|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{\text{tr}} \\
 \leq & (1 + \|\mathbf{I}_n - (\hat{\mathbf{P}} - \mathbf{\Pi}_L)\|_2)\mathcal{R}_L
 \end{aligned}$$

$$+ |\alpha - \gamma_L|\|d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}\| - \gamma_L\|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{\text{tr}}. \tag{30}$$

Using the triangular inequality $\|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{\text{tr}} \geq |d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}|$, we get

$$\begin{aligned}
 0 \leq & (1 + \|\mathbf{I}_n - (\hat{\mathbf{P}} - \mathbf{\Pi}_L)\|_2)\mathcal{R}_L \\
 & + (|\alpha - \gamma_L| - \gamma_L)\|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{\text{tr}}. \tag{31}
 \end{aligned}$$

We next employ the estimate

$$\|\mathbf{I}_n - (\hat{\mathbf{P}} - \mathbf{\Pi}_L)\|_2 \leq \|\mathbf{I}_n - \hat{\mathbf{P}}\|_2 + \|\mathbf{\Pi}_L\|_2 \leq 2, \tag{32}$$

which results in

$$\|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{\text{tr}} \leq \frac{3\mathcal{R}_L}{\gamma_L - |\gamma_L - \alpha|}$$

if $\alpha < 2\gamma_L$. Alternatively, we apply (32) in (30) together with $|d_L - \|\hat{\mathbf{P}}\|_{\text{tr}}| \leq d$, since $d_L \leq d$, and obtain

$$\|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{\text{tr}} \leq \frac{3\mathcal{R}_L + d|\alpha - \gamma_L|}{\gamma_L}.$$

The final assertion follows by

$$\|\hat{\mathbf{\Pi}} - \mathbf{\Pi}_L\|_{\text{tr}} \leq \|\hat{\mathbf{\Pi}} - \hat{\mathbf{P}}\|_{\text{tr}} + \|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{\text{tr}} \leq 2\|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{\text{tr}}.$$

□

The error depends basically on the quotient of the distance of the normalized data from the subspace and the permeance statistics, where the normalization is indirectly contained in the quotient. For $\alpha = \gamma_L$, the terms of the minimum in (26) are the same and both terms become small; the second term is preferred for α near zero or $2\gamma_L$. Compared to our experimental results, the estimate is in general too pessimistic.

Remark 5 In [27], the following estimate was given for the original REAPER

$$\begin{aligned}
 \|\hat{\mathbf{\Pi}} - \mathbf{\Pi}_L\|_{\text{tr}} &\leq 16\sqrt{d_L}\mathcal{R}_L(\mathbf{X}_{\text{in}}) \\
 &\quad / \left(\mathcal{P}_L(\mathbf{X}_{\text{in}}) - 4\sqrt{d_L}\mathcal{R}_L(\mathbf{X}_{\text{in}}) \right. \\
 &\quad \left. - 4\sqrt{d_L}\|\mathbf{X}_{\text{out}}\| \|\mathbf{((\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{X}_{\text{out}})^{\sim}\|} \right)_+, \tag{33}
 \end{aligned}$$

where the data points $\mathbf{X} = [\mathbf{X}_{\text{in}}|\mathbf{X}_{\text{out}}]$ were divided into inliers $\mathbf{X}_{\text{in}} \in \mathbb{R}^{n, N_{\text{in}}}$ and outliers $\mathbf{X}_{\text{out}} \in \mathbb{R}^{n, N_{\text{out}}}$ and $(\cdot)^{\sim}$ normalizes the columns of a matrix one. Following the lines of the proof in those paper, we would obtain the estimate

$$\begin{aligned}
 \|\hat{\mathbf{\Pi}} - \mathbf{\Pi}_L\|_{\text{tr}} &\leq \left(16\sqrt{d_L}\mathcal{R}_L(\mathbf{X}_{\text{in}}) + 8\alpha d_L\sqrt{d_L} \right)
 \end{aligned}$$

$$\begin{aligned} & \left/ \left(\mathcal{P}_L(\mathbf{X}_{in}) - 4\sqrt{d_L}\mathcal{R}_L(\mathbf{X}_{in}) \right. \right. \\ & \left. \left. - 4\sqrt{d_L}\|\mathbf{X}_{out}\| \left\| \left((\mathbf{I}_n - \mathbf{\Pi}_L)\mathbf{X}_{out} \right)^\sim \right\| \right)_+ \end{aligned} \quad (34)$$

for rREAPER. Therefore, rREAPER inherits the robustness from REAPER as well as the theoretical guarantees for certain data models. However, in our numerical examples, we realized that the above estimate is often not defined due to division by zero, especially in the presents of already small inlier noise, i.e., if the columns of \mathbf{X}_{in} are only near the subspace L . Moreover, it is not clear how the division into in- and outliers can be achieved for real data sets. In our estimate (26), we do not distinguish between in- and outliers. Using alternatively $\|\mathbf{I}_n - (\hat{\mathbf{P}} - \mathbf{\Pi}_L)\|_2 \leq 1 + \|\hat{\mathbf{P}} - \mathbf{\Pi}_L\|_{tr}$ in (31), we would obtain the estimate

$$\|\hat{\mathbf{\Pi}} - \mathbf{\Pi}_L\|_{tr} \leq \frac{8\sqrt{2d_L}\mathcal{R}_L}{(\mathcal{P}_L - 2\sqrt{2d_L}\mathcal{R}_L - |\mathcal{P}_L - 2\sqrt{2d_L}\alpha|)_+}$$

7 Incorporating the Offset

So far, we have assumed that the offset \mathbf{b} in the robust PCA problem is given, so that we can just search for a low-dimensional linear subspace which represents the data well. While in the classical PCA (4), the affine subspace always passes through the mean value (5) of the data, it is not clear which value \mathbf{b} must be chosen in order to minimize

$$\begin{aligned} E(\mathbf{A}, \mathbf{b}) & := \sum_{i=1}^N \|(\mathbf{A}\mathbf{A}^T - \mathbf{I}_n)(\mathbf{b} - \mathbf{x}_i)\|_2 \\ & \text{subject to } \mathbf{A}^T\mathbf{A} = \mathbf{I}_d. \end{aligned} \quad (35)$$

Clearly, if $(\hat{\mathbf{A}}, \hat{\mathbf{b}})$ is a minimizer of E , then, for every $\mathbf{b} \in \text{ran}(\hat{\mathbf{A}})$, also $(\hat{\mathbf{A}}, \hat{\mathbf{b}} + \mathbf{b})$ is a minimizer.

A common choice for the offset is the *geometric median* of the data points defined by

$$\hat{\mathbf{b}} := \underset{\mathbf{b} \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{k=1}^N \|\mathbf{b} - \mathbf{x}_k\|_2,$$

which can be computed, e.g., by the Weiszfeld algorithm and its generalizations, see, e.g., [2,36,42,49]. Other choices arising, e.g., from Tyler’s M -estimator or other robust statistical approaches [19,24,26,33,47], were proposed in the literature. However, they do in general not minimize (35) as the following example from [35] shows: given three points in \mathbb{R}^n which form a triangle with angles smaller than 120° , the geometric median is the point in the inner of the triangle from which the points can be seen under an angle of 120° . In contrast, the line ($d = 1$) having smallest distance from the three points

is the one which passes through those two points with the largest distance from each other.

In the following, we show that there always exists an optimal hyperplane of dimension $d = n - 1$ in \mathbb{R}^n determined by a minimizer of E in (35) that contains at least n data points. Further, if the number N of data points is odd, then every optimal hyperplane contains at least n data points. Recall that the hyperplane spanned by the columns of $\mathbf{A} = (\mathbf{a}_1 | \dots | \mathbf{a}_{n-1}) \in \text{St}(n, n - 1)$ with offset \mathbf{b} is given by

$$\{\mathbf{x} = \mathbf{A}\mathbf{t} + \mathbf{b} : \mathbf{t} \in \mathbb{R}^{n-1}\} = \{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{a}^\perp, \mathbf{x} \rangle = \beta\},$$

where $\mathbf{a}^\perp \perp \mathbf{a}_i, i = 1, \dots, n - 1$ is a unit normal vector of the hyperplane, which is uniquely determined up to its sign and $\langle \mathbf{a}^\perp, \mathbf{b} \rangle = -\beta$.

The following lemma describe the placement of the data points with respect to the halfspaces determined by a minimizing hyperplane.

Lemma 3 *Let $\mathbf{x}_k \in \mathbb{R}^n, k = 1, \dots, N$. Let $(\hat{\mathbf{A}}, \hat{\mathbf{b}})$ be a minimizer of E and $N = M + M_+ + M_-$ with*

$$\begin{aligned} M & := |\{\mathbf{x}_k : \langle \hat{\mathbf{a}}^\perp, \mathbf{x}_i \rangle = \hat{\beta}\}|, \\ M_+ & := |\{\mathbf{x}_k : \langle \hat{\mathbf{a}}^\perp, \mathbf{x}_i \rangle > \hat{\beta}\}|, \\ M_- & := |\{\mathbf{x}_k : \langle \hat{\mathbf{a}}^\perp, \mathbf{x}_i \rangle < \hat{\beta}\}|. \end{aligned}$$

Then it holds $|M_+ - M_-| \leq M$. In particular, it holds $M \geq 1$ if N is odd. Also for even N there exists a minimizing hyperplane with $\hat{\mathbf{b}} = \mathbf{x}_k$ for some $k \in \{1, \dots, N\}$.

Proof W.l.o.g. assume that $M_+ \geq M_-$. If $M_+ = 0$, then all data points are in the minimizing hyperplane and we are done. Otherwise, the value

$$\varepsilon := \min\{\langle \hat{\mathbf{a}}^\perp, \mathbf{x}_k \rangle > 0 : k = 1, \dots, N\}$$

is positive, and we consider the shifted hyperplane $\{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{a}^\perp, \mathbf{x} \rangle = \beta + \varepsilon\}$, which contains at least one data point. The sum of the distances of the data points from this hyperplane is

$$E(\hat{\mathbf{A}}, \hat{\mathbf{b}}) - \varepsilon M_+ + \varepsilon(M + M_-).$$

Since $(\hat{\mathbf{A}}, \hat{\mathbf{b}})$ is a minimizer of E , this implies that $M_+ \leq M + M_-$. If $M = 0$, then $M_- = M_+$ and the shifted hyperplane is also minimizing. However, this case cannot appear for odd N so that $M \geq 1$ for odd N . This finishes the proof. \square

Theorem 3 *Let $\mathbf{x}_\ell \in \mathbb{R}^n, \ell = 1, \dots, N$. Then there exists a minimizer $(\hat{\mathbf{A}}, \hat{\mathbf{b}})$ of E such that the corresponding minimizing hyperplane contains at least n data points. If N is odd, every minimizing hyperplane contains at least n data points.*

Proof By Lemma 3, there exists a data point \mathbf{x}_ℓ such that $(\hat{\mathbf{A}}, \hat{\mathbf{b}})$ with $\hat{\mathbf{b}} = \mathbf{x}_\ell$ is a minimizer of E and for odd N every minimizing hyperplane passes through a data point. Let $\hat{\mathbf{a}}^\perp$ be a unit vector orthogonal to the columns of $\hat{\mathbf{A}}$. Set $\mathbf{y}_k := \mathbf{x}_k - \mathbf{x}_\ell, k = 1, \dots, N$. Next, we show: if the subspace normal to $\hat{\mathbf{a}}^\perp$ contains M linearly independent vectors $\mathbf{y}_1, \dots, \mathbf{y}_M$ with $0 \leq M \leq n - 2$, then exactly one of the following situations apply. (i) The remaining vectors \mathbf{y}_k with $k = M + 1, \dots, N$ are linearly dependent from the first M vectors and thus in the same linear subspace $\text{span}\{\mathbf{y}_k : k = 1, \dots, M\}$. (ii) We find a further independent vector, say \mathbf{y}_{M+1} , contained in the subspace normal to $\hat{\mathbf{a}}^\perp$ such that we can increase M to $M + 1$. Repeating this argumentation until $M = n - 2$, we are done since $\mathbf{y}_k + \mathbf{x}_\ell, k = 1, \dots, n - 1$ and \mathbf{x}_ℓ itself are in the subspace corresponding to $(\hat{\mathbf{A}}, \hat{\mathbf{b}})$.

Because the vectors \mathbf{y}_k with $k = 1, \dots, M$ are independent and are contained in the subspace normal to $\hat{\mathbf{a}}^\perp$ by assumption, there exists a matrix

$$\mathbf{B} := (\mathbf{u}_1 | \dots | \mathbf{u}_M | \mathbf{u}_{M+1} | \dots | \mathbf{u}_{n-1}) \in \text{St}(n, n - 1),$$

with $\text{ran}(\mathbf{A}) = \text{ran}(\mathbf{B})$, whose first columns have the same span as $\mathbf{y}_1, \dots, \mathbf{y}_M$, i.e.,

$$\text{span}\{\mathbf{u}_k : k = 1, \dots, M\} = \text{span}\{\mathbf{y}_k : k = 1, \dots, M\}.$$

This especially implies $\mathbf{y}_k \perp \mathbf{u}_\ell$ for $k = 1, \dots, M$ and $\ell = M + 1, \dots, n - 1$. Note that the normal unit vector of $\text{ran}(\mathbf{B})$ is also $\hat{\mathbf{a}}^\perp$, and that the objectives coincides, i.e.,

$$\sum_{k=1}^N \|(\hat{\mathbf{A}}\hat{\mathbf{A}}^\text{T} - \mathbf{I}_n)\mathbf{y}_k\|_2 = \sum_{k=1}^N \|(\mathbf{B}\mathbf{B}^\text{T} - \mathbf{I}_n)\mathbf{y}_k\|_2. \tag{36}$$

Now, let the matrix-valued function

$$\phi_{\mathbf{B}} : [-\pi, \pi) \rightarrow \text{St}(n, n - 1)$$

be defined by

$$\phi_{\mathbf{B}}(\alpha) := \mathbf{Q}_{\mathbf{B}} \mathbf{R}(\alpha) \mathbf{C},$$

where the three building factors are given by

$$\begin{aligned} \mathbf{Q}_{\mathbf{B}} &:= (\mathbf{B} | \hat{\mathbf{a}}^\perp), \\ \mathbf{R}(\alpha) &:= \begin{pmatrix} \mathbf{I}_{n-2} & \mathbf{0}_{n-2} & \mathbf{0}_{n-2} \\ \mathbf{0}_{n-2}^\text{T} & \cos(\alpha) & \sin(\alpha) \\ \mathbf{0}_{n-2}^\text{T} & -\sin(\alpha) & \cos(\alpha) \end{pmatrix}, \\ \mathbf{C} &:= \begin{pmatrix} \mathbf{I}_{n-1} \\ \mathbf{0}_{n-1}^\text{T} \end{pmatrix}. \end{aligned}$$

Figuratively, the function $\phi_{\mathbf{B}}$ takes the orthonormal columns of \mathbf{B} and rotates the last vector \mathbf{u}_{n-1} by the angle α in the

plane spanned by \mathbf{u}_{n-1} and $\hat{\mathbf{a}}^\perp$. Clearly, we have $\phi_{\mathbf{B}}(0) = \mathbf{B}$. Due to (36), the function

$$F(\alpha) = \sum_{k=1}^N \|(\phi_{\mathbf{B}}(\alpha)\phi_{\mathbf{B}}(\alpha)^\text{T} - \mathbf{I}_n)\mathbf{y}_k\|_2 = \sum_{k=1}^N f_k(\alpha)$$

has moreover a minimum in $\alpha = 0$. For the summands of F , we obtain

$$\begin{aligned} f_k(\alpha) &= \|(\phi_{\mathbf{B}}(\alpha)\phi_{\mathbf{B}}(\alpha)^\text{T} - \mathbf{I}_n)\mathbf{y}_k\|_2 \\ &= \|(\mathbf{Q}_{\mathbf{B}} \mathbf{R}(\alpha) \mathbf{C} \mathbf{C}^\text{T} \mathbf{R}(\alpha)^\text{T} \mathbf{Q}_{\mathbf{B}}^\text{T} - \mathbf{I}_n)\mathbf{y}_k\|_2 \\ &= \|(\mathbf{C} \mathbf{C}^\text{T} \mathbf{R}(\alpha)^\text{T} \mathbf{Q}_{\mathbf{B}}^\text{T} - \mathbf{R}(\alpha)^\text{T} \mathbf{Q}_{\mathbf{B}}^\text{T})\mathbf{y}_k\|_2 \\ &= \|(\mathbf{C} \mathbf{C}^\text{T} - \mathbf{I}_n)\mathbf{R}(\alpha)^\text{T} \mathbf{Q}_{\mathbf{B}}^\text{T}\mathbf{y}_k\|_2 \\ &= \left| \sin(\alpha)\langle \mathbf{u}_{n-1}, \mathbf{y}_k \rangle + \cos(\alpha)\langle \hat{\mathbf{a}}^\perp, \mathbf{y}_k \rangle \right|, \end{aligned}$$

since $\mathbf{Q}_{\mathbf{B}}$ and $\mathbf{R}(\alpha)$ are orthogonal by construction. Hence, we get

$$F(\alpha) = \sum_{k=M+1}^N \left| \sin(\alpha)\langle \mathbf{u}_{n-1}, \mathbf{y}_k \rangle + \cos(\alpha)\langle \hat{\mathbf{a}}^\perp, \mathbf{y}_k \rangle \right|.$$

Here the first M summands vanish because of the mentioned orthogonality $\mathbf{y}_k \perp \mathbf{u}_{n-1}$ and $\mathbf{y}_k \perp \hat{\mathbf{a}}^\perp$ for $k = 1, \dots, M$.

If all remaining given points \mathbf{y}_k with $k = M + 1, \dots, N$ are in $\text{span}\{\mathbf{y}_1, \dots, \mathbf{y}_M\}$, then the corresponding remaining summands of $F(\alpha)$ become zero too, and the first situation (i) applies; so we are done.

If this is not the case, consider only those \mathbf{y}_k with $k = M + 1, \dots, N$ that are linearly independent of the $\mathbf{y}_k, k = 1, \dots, M$. Let us denote the corresponding non-empty index set by \mathcal{S} . Assume that there exists a $k \in \mathcal{S}$ such that f_k is not differentiable in $\alpha = 0$. This is only possible if the argument of the absolute value vanishes implying

$$f_k(0) = |\langle \hat{\mathbf{a}}^\perp, \mathbf{y}_k \rangle| = 0.$$

Thus, the vector \mathbf{y}_k is in the subspace spanned by the columns of $\hat{\mathbf{A}}$, and we are done. Otherwise, if $f_k(0) \neq 0$ for all $k \in \mathcal{S}$, then it is differentiable in $\alpha = 0$ and, by straightforward differentiation, we obtain

$$f_k''(0) = -f_k(0) < 0.$$

But then $\alpha = 0$ cannot be a minimizer of F which is a contradiction. Hence, this case cannot occur and the proof is complete. \square

If the target dimension d of the minimizing subspace is strictly less than $n - 1$, then it does not have to contain any data point as the following example shows.

Counterexample 1 (*Lower-dimensional subspace approximation*) Initially, we consider the approximation of some given points in \mathbb{R}^3 by an one-dimensional subspace—a line. More precisely, for a fixed $T \gg 1$, we consider the six given points

$$(\cos(\phi), \sin(\phi), \pm T)^T \text{ with } \phi \in \{0, 2\pi/3, 4\pi/3\}.$$

We thus have two well-separated clusters around $(0, 0, T)^T$ and $(0, 0, -T)^T$.

Obviously, the optimal line has somehow to go through each cluster. One possible candidate for the approximation line is simply the axis $\{(0, 0, t) : t \in \mathbb{R}\}$, whose distance to the given points is by construction 6—for each cluster 3. Now, assume that the line goes through one given point, say $(1, 0, T)^T$. If T is very large, then we can neglect the slope of the line. Only considering the distances within the cluster around $(0, 0, T)^T$, we notice that the distance increases from 3 to approximately $2\sqrt{3}$. Although the axis is maybe not the optimal line, the distance to the given points is smaller than for a line going through a data point. Therefore, we can conclude that the optimal line has not to contain any given point.

The same construction can be done for arbitrary subspaces of dimension $d < n - 1$. For example, consider just the points

$$(\cos(\phi), \sin(\phi) | \pm T \mathbf{e}_k^T)^T$$

with

$$\phi \in \{0, 2\pi/3, 4\pi/3\}, k = 1, \dots, d,$$

where \mathbf{e}_k is the k th unit vector. Using the same argumentation as above, the distance to the subspace $\text{span}\{\pm \mathbf{e}_k : k = 1, \dots, d\}$ is smaller than to any d -dimensional subspace containing at least one data point.

8 Numerical Examples

In this section, we demonstrate the performance of rREAPER by numerical examples implemented in MATLAB® (R2019b, 64-bit) and calculated using an Intel® Core™ i7-8700 CPU (6 × 3.20 GHz) and 32 GiB main memory. For synthetic data, we employ the so-called *haystack model* introduced by Lerman et al. in [27]. The idea is to generate a set of inliers $\mathbf{X}_{\text{in}} \in \mathbb{R}^{n, N_{\text{in}}}$ lying near an d_L -dimensional subspace—here the subspace spanned by the first d_L unit vectors—and a set of outliers $\mathbf{X}_{\text{out}} \in \mathbb{R}^{n, N_{\text{out}}}$ located somewhere in the surrounding space. More precisely, the elements of the in- and

outliers are drawn from the Gaussians

$$(\mathbf{X}_{\text{in}})_{\ell, k} \sim \begin{cases} \mathcal{N}(0, \sigma_{\text{in}}^2/d_L) & \text{if } \ell \leq d_L, \\ \mathcal{N}(0, \sigma_{\text{noise}}^2/(n - d_L)) & \text{otherwise,} \end{cases}$$

and

$$(\mathbf{X}_{\text{out}})_{\ell, k} \sim \mathcal{N}(0, \sigma_{\text{out}}^2/n).$$

The complete data set then consists of the matrix

$$\mathbf{X} = [\mathbf{X}_{\text{in}} | \mathbf{X}_{\text{out}}].$$

Further, we compare the results of the matrix-free rREAPER with several other robust PCA methods which can be applied to high-dimensional images. More precisely, we consider the following methods:

- *PCA-L1 method* [21] maximizes the L1 dispersion $\|\mathbf{P}\mathbf{X}\|_{1,1}$. The method is based on a greedy search algorithm that determines the maximum $\hat{\mathbf{P}}$ by finding the rank-one projections to the principal components. The method finds a local maximum, but there is no guarantee to compute the global one. Indeed, the results heavily depends on the start value, which is the reason why we restart the algorithm several times (actually 10 times). Further, in contrast to the other three methods, this one is not rotationally invariant due to the objective function which often results in lower quality.
- *Nested Weiszfeld algorithm* [35] aims to approximate the minimizer of $\|(\mathbf{I}_n - \mathbf{P})\mathbf{X}\|_{2,1}$. The projection is determined by a greedy-like method by computing the principle components sequentially. In general, this does not yield a (local) minimizer of $\|(\mathbf{I}_n - \mathbf{P})\mathbf{X}\|_{2,1}$.
- *RI-PCA* [9,34], which minimizes again the data fidelity $\|(\mathbf{I}_n - \mathbf{P})\mathbf{X}\|_{2,1}$. The minimization is performed by a gradient descent method on the Grassmannian manifold which is equivalent to a conditional gradient algorithm—also known as Frank–Wolfe algorithm. Convergence to a (local) minimizer is only guaranteed under certain assumptions on the anchor set.
- *Fast Median Subspace method (FMS)* [25] is a heuristic method to minimize the energy $\|(\mathbf{I}_n - \mathbf{P})\mathbf{X}\|_{2,p} := \sum_{k=1}^N \|(\mathbf{I}_n - \mathbf{P})\mathbf{x}_k\|_2^p$ with $0 < p < 2$ iteratively. For this, a sequence of subspaces is constructed on the basis of weighted PCAs. FMS always converge to a stationary point (or to a continuum of stationary points), which are usually (local) minimizers of the energy functional.
- *Geodesic Gradient Descent (GGD)* [31] intends to minimize $\|(\mathbf{I}_n - \mathbf{P})\mathbf{X}\|_{2,1}$ by a geodesic descent performed on the Grassmannian. Under certain conditions on the problem and starting value, this method linearly converges to the underlying unknown subspace.

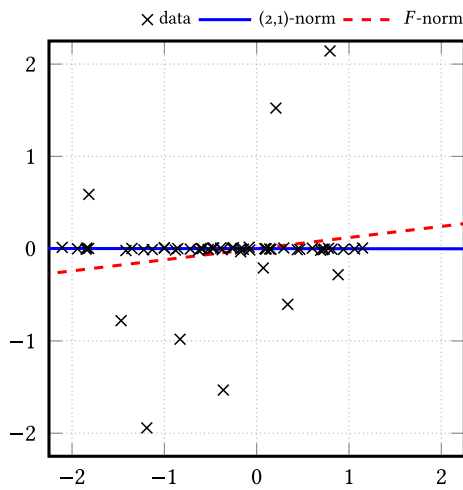


Fig. 1 Performance of rREAPER with (2,1)-norm and Frobenius norm in the data fidelity term, respectively. The first one appears to be more robust against outliers

8.1 (2,1)-Norm versus Frobenius Norm

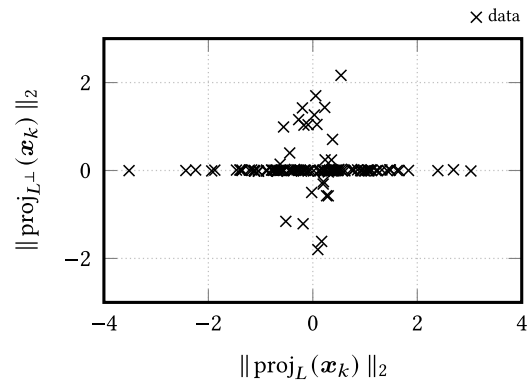
This example with simple synthetic data will show that the (2,1)-norm in the data term is more robust against outliers than the Frobenius norm

$$\left(\sum_{k=1}^N \|(P - I_n) \mathbf{x}_k\|_2^2 \right)^{\frac{1}{2}} = \|PX - X\|_F.$$

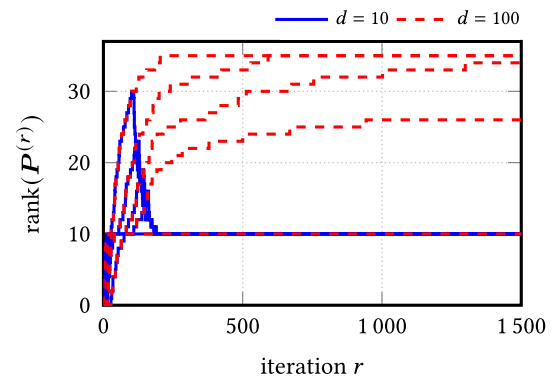
For the Frobenius norm here abbreviated as F -norm, we have only to replace the projection onto $\mathfrak{B}_{2,\infty}$ with the projection to the Frobenius norm ball

$$\text{proj}_{\mathfrak{B}_F}(\mathbf{Y}) = \frac{\mathbf{Y}}{[\|\mathbf{Y}\| - 1]_+ + 1}.$$

We want to recover a line in the plane. Since this recovery problem is invariant under rotations, we restrict ourselves to $\text{span}\{(1, 0)^T\}$. The data are generated randomly using the haystack model and consist of 50 points near the considered axis—we added a small amount of noise in the second coordinate—and of 10 outliers located somewhere in the plane, see Fig. 1. Besides the data points, the recovered lines using rREAPER with the (2,1)-norm (solid line) as data fidelity and the Frobenius norm (dashed line) with parameters $d = 1$ and $\alpha = 5$ are presented. In this toy example, rREAPER yields nearly a perfect result regardless of the outliers and is in particular more robust than the same model with the Frobenius norm.



(a) Representation of the data set.



(b) Evolution of the rank.

Fig. 2 Performance of rREAPER for different upper dimension estimators $d = 10, 100$ and regularization parameters $\alpha = 1/4, 1/2, 3/4, 1, 2$ (top down for each d)

8.2 Nuclear Norm and Truncated Hypercube Constraints

In this example, we are interested how the rank reduction is influenced by the nuclear norm and the projection to the truncated hypercube. In this synthetic experiment, we approximate the given data $\mathbf{x}_k \in \mathbb{R}^{100}$ by a 10-dimensional subspace. The data are again generated randomly with respect to the haystack model with $\sigma_{\text{in}} = \sigma_{\text{out}} = 1$ and $\sigma_{\text{noise}} = 0.1$, where 100 points lie near the subspace L spanned by the first ten unit vectors and additional 25 outliers. In Fig. 2a, the data set is represented by the distance to the subspace L and to the orthogonal complement L^\perp .

We apply rREAPER in Algorithm 8 with different parameter combination. For the target dimension, we choose in our first experiment $d = 10$, which is the wanted dimension, and second one $d = 100$, which does not truncate the unit hypercube at all. The influence of the regularization parameter α on the rank of $\mathbf{P}^{(r)}$ is shown in Fig. 2b, where the lines from top to down correspond to the regularization parameters $\alpha = 1/4, 1/2, 3/4, 1, 2$. Since we start the iteration with the zero-rank matrix $\mathbf{P}^{(0)} := \mathbf{0}$, the first iterations for $d = 10$

Table 1 Reconstruction error $\|\hat{\Pi} - \Pi_L\|_{tr}$ for different upper dimension estimators $d = 10, 100$ and regularization parameters $\alpha = 1/4, 1/2, 3/4, 1, 2$

dim. d	Regularization α				
	0.25	0.5	0.75	1	2
10	0.0648	0.0646	0.0645	0.0645	0.0646
100	25.0523	25.0523	24.0569	16.0576	0.0646

and $d = 100$ coincides up to the point, where the trace of $P^{(r)}$ exceeds the value 10. The reconstruction errors are recorded in Table 1.

Considering only the results for $d = 100$ (dashed lines), we see that the nuclear norm reduces the rank of the iteration variable $P^{(r)}$ with an increasing regularization parameter. Further, the rank during the primal-dual algorithm is very sensitive to the regularization parameter. For $d = 10$ (solid lines), the situation changes dramatically. After the initial stages, the rank of $P^{(r)}$ decreases nearly to the target dimension. Since the matrices $P^{(r)}$ are no orthogonal projections, rank and trace do not coincide. Due to this fact, the rank is not strictly bounded by the maximal trace of the truncated hypercube. Nevertheless, the projection to the truncated hypercube significantly reduces the rank.

For an optimal rank evolution during the matrix-free primal-dual method, the projection to the truncated hypercube by Algorithm 2 appears to be important. Moreover, the projection makes the rank evolution less sensitive to the regularization parameter α so that a wider range of regularization parameters can be applied without losing the computational benefits of the low rank. Thus, the truncated hypercube projection is an elementary key component of the algorithm.

Finally, we compare the performance of rREAPER in this experiment with the other robust PCA methods mentioned above. For this, we repeat the experiment 100 times with random data. For rREAPER, we choose the parameters $d = 10$ and $\alpha = 0.75$. The mean reconstruction errors and the mean run times have been recorded in Table 2. In this experiment, the PCA-L1 fail to approximate the true underlying subspace. Interestingly, rREAPER here always yields a slightly better approximation than R1-PCA, FMS, and GGD. The reason for this behavior is that the nuclear norm regularization suppress the Gaussian noise within the computed principle components.

8.3 Choosing the Parameters

In the last example, we have seen that the target dimension is achieved by choosing either d or α appropriately. To understand the relation between the parameters in more detail and finally to get a clue how to choose α if d is just taken large enough, we provide some additional experiments.

Table 2 Mean reconstruction error $\|\hat{\Pi} - \Pi_L\|_{tr}$ and mean run time (in seconds) for different robust PCA methods

	Mean error	Mean time
rREAPER	0.0651	0.5953
PCA-L1	3.4235	0.0490
Nested Weiszfeld	0.3583	0.1613
R1-PCA	0.0662	0.4856
FMS	0.0662	0.0230
GGD	0.0662	0.0149

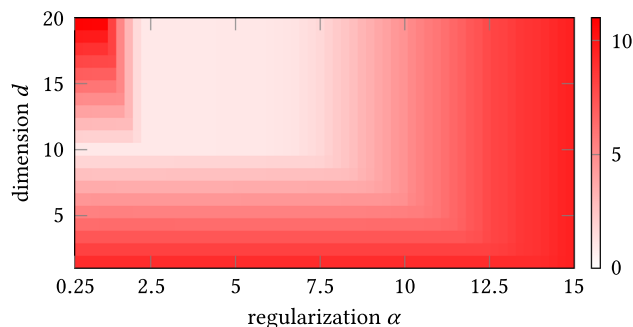
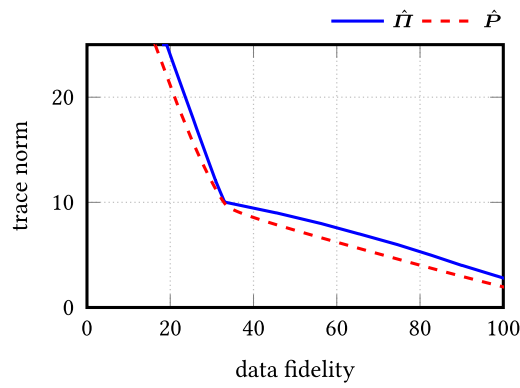


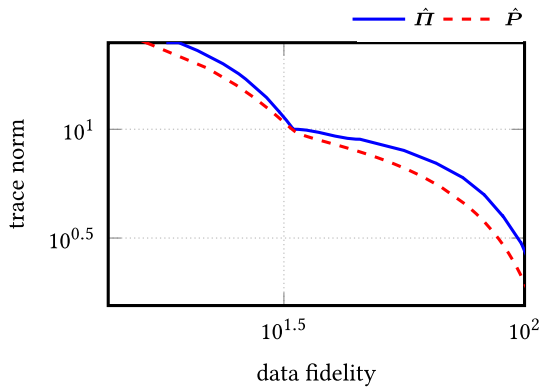
Fig. 3 Mean reconstruction error $\|\hat{\Pi} - \Pi_L\|_{tr}$ for varying α and d . For every parameter choice, the experiment has been repeated 25 times

First, we repeat the experiment in Sect. 8.2 sequentially 25 times and consider the mean of the reconstruction error $\|\hat{\Pi} - \Pi_L\|$ of all 25 experiments, see Fig. 3. For each choice of α and d , the experiment has been repeated with the same sets of data. For this specific haystack model, the unknown subspace is recovered by choosing either $d = 10$ or $\alpha \in [2.25, 6.75]$. Hence, the first conclusion is that the reconstruction error is much less sensitive to choice of α than to those of d .

Next, considering again the variational nature of rREAPER in (9), we recall that $\|PX - X\|_{2,1}$ describes the error between the ‘projected data’ PX and the given data X , and that the $\|P\|_{tr}$ controls the rank of P . In order to find a good subspace, both terms have to be small. Notice that the data fidelity becomes small if the rank/trace becomes large and vice versa. Consequently, α has to be chosen such that both complementary aspects are balanced. One approach coming from inverse problems is the so-called L -curve, see for instance [14,15]. The L -curve is a usual or a log-log plot of the data fidelity and the regularizer. For the haystack data set, the corresponding L -curve—once computed for the outcome \hat{P} of rREAPER, and once for its projection $\hat{\Pi} = \text{proj}_{\mathcal{O}_{25}}(\hat{P})$ —is shown in Fig. 4, where $d = 25$. Notice that both L -curves have a singularity, especially in the log-log plot. For the sake of clarity, we also plotted the data fidelity and the trace norm over α , see Fig. 5. The corner of the L -curve for $\hat{\Pi}$ corresponds to $\alpha = 9.2$.



(a) The L-curve.

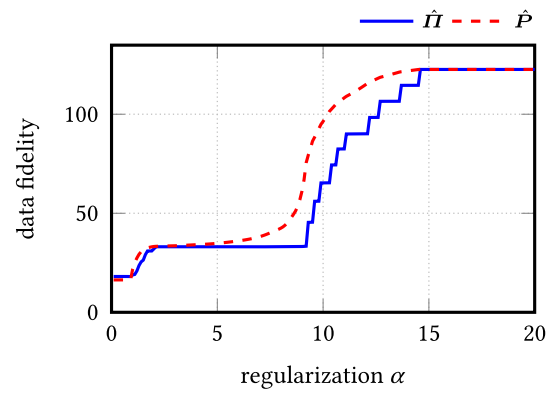


(b) The log-log L-curve.

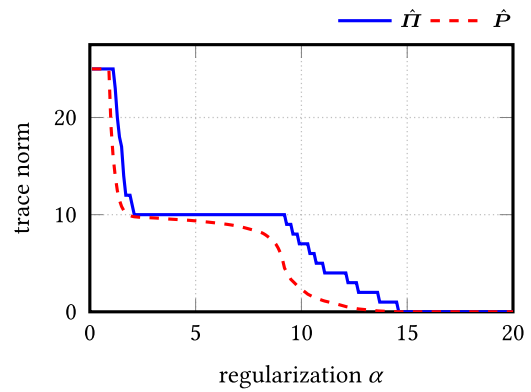
Fig. 4 The L -curve for $d = 25$ of the data set in Sect. 8.2 consisting of 100 inliers ($\sigma_{in} = 1$) and 25 outliers ($\sigma_{out} = 1$) with noise $\sigma_{noise} = 0.1$

Further, the L -curve slows down at the singularity and is nearly constant for $\alpha \in [2.1, 9.2]$. The interval corresponds to the plateaus of the data fidelity and the trace norm in dependence of the regularization parameter α , see Fig. 5. At the interval, data fidelity and regularizer are balanced. More importantly, for every $\alpha \in [2.1, 9.2]$, the determined subspace has dimension 10 and thus coincides with the true dimension.

We repeated this experiment around 500 times for random chosen d , d_L , N_{in} , N_{out} , and σ_{noise} and observe the same behavior of the L -curve meaning that the edge of the curve has been clearly visible and corresponds to the true subspace dimension. Therefore, it seems that the unknown subspace dimension can be detected and that an appropriate regularization parameter α may be chosen by studying the L -curve of the given data set. Notice that the L -curve, also well studied in the literature, is a heuristic parameter choice rule. Finally let us mention that if the inlier noise is too large or if the outliers hide the linear structure of the inliers, then the L -curve loses its singularity.



(a) Data fidelity.



(b) Regularizer.

Fig. 5 The two components of the L -curves in Fig. 4 in dependence of the regularization parameter α

8.4 Face Approximation

The idea to use the principle components of face images—the so-called eigenfaces—for recognition, classification and reconstruction was considered in various paper and goes probably back to [45]. In this experiment, we adopt this idea to show that our matrix-free REAPER can handle high-dimensional data. Since the computation of an optimal offset is non-trivial as discussed in Sect. 7, we choose just the geometric median. For the experiment, we use the ‘Extended Yale Face Data Set B’ [12,23] consisting of several series of faces images under different lighting conditions but with the same facial expression. The basis of our noisy data set are 64 images of size 640×480 pixels with integer values between 0 and 255. We place some artifacts in the first four images covering the right eye, the nose, the right ear and the mouth, respectively, and serving as outliers. The complete employed data set is shown in Fig. 6a.

It is well-known that such images can be well approximated by a subspace covering around five directions [10]. To remove the artifacts by unsupervised learning, we therefore approximate the full data set including the artificial

Fig. 6 The used data set based on the Extended Yale Face Data Set B and its projections onto the determined subspace using matrix-free rREAPER



(a) Data set of the experiment with Gaussian noise and artifacts in the first four images.



(b) Projections onto the 5-dimensional rREAPER subspace.

Fig. 7 Restoration of the first sample by projecting to the principle components determined by several robust PCA's



(a) Original sample.



(b) Noise and artefact.



(c) RREAPER projection.



(d) PCA-L1 projection.



(e) Weiszfeld projection.



(f) R1-PCA projection.

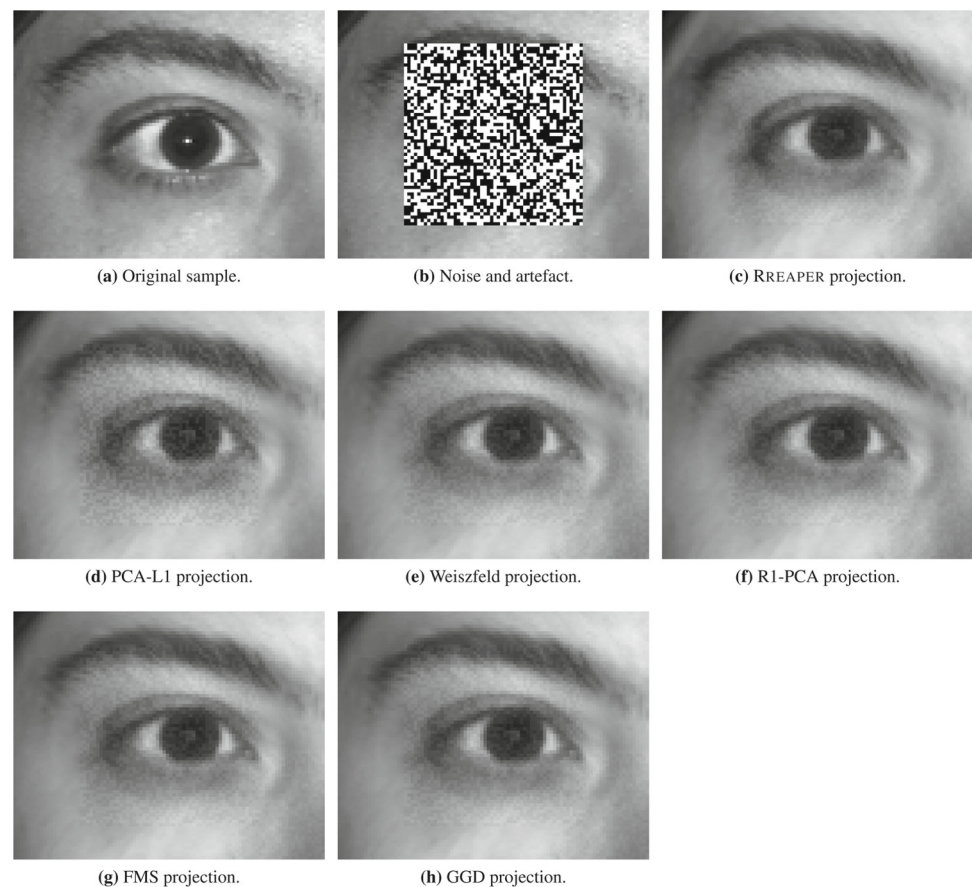


(g) FMS projection.



(h) GGD projection.

Fig. 8 Restoration of the artifact in the first sample by projecting to the principle components determined by several robust PCA's



face images by a five-dimensional subspace ($d = 5$) using rREAPER and project the corrupted images to the computed subspace. An typical effect of this procedure is that dark regions are lightened, shadows are removed, and reflections at skin and eyes are cleared away. Besides this effects, the major part of the unwanted artefacts is removed, see Fig. 6b. Note that in this example the projection $\hat{\mathbf{I}}$ corresponds to a $307,200 \times 307,200$ matrix, which would require 703.125 GiB for double precision whereas the matrix-free representation only requires around 14.063 MiB since the rank of the primal variable has been bounded by six due to the choice of $\alpha := 4.2 \cdot 10^4$. The primal-dual algorithm for rREAPER has converged after 30 iterations.

Next, we compare rREAPER with the robust PCA methods mentioned above. Figure 7 shows examples of the projection of the first corrupted image to the computed subspace of the four considered methods. The restoration of the artifact covering the right eye is shown in Fig. 8 in more details. Taking a closer look in Fig. 8, we notice that PCA-L1, although very fast, yields the worst result, which has also been reported in [35]. The nested Weiszfeld algorithm and R1-PCA nearly remove the artifact. The result of rREAPER is comparable but appears to be slightly smoother, which is an effect of the additional regularization.

Comparing the required run times, we notice that PCA-L1 with 10 restarts is the fastest method with 10.189 (in seconds) due to its linear structure, followed by FMS with 12.313, and the nested Weiszfeld algorithm with 18.875. R1-PCA, rREAPER and GGD behave similarly with 43.441, 43.439 and 55.968, respectively. Note that the run times are only comparable to a limited extent due to the different nature of the algorithms and hence different stopping criteria. For instance, GGD here requires a very small step size to stay numerically at the Grassmannian. We can clearly see that rREAPER is slower than the sequential or heuristic approaches. It seems that the numerical effort of rREAPER is however comparable with the non-convex methods like R1-PCA. rREAPER converges however to a global minimizer, where the non-convex methods may be stuck in local minima, which here happens for GGD.

9 Conclusion

Convex models are usually preferable over non-convex ones due to their unique local minimum. While robust PCA models that can handle high-dimensional data are usually non-convex, a convex relaxation was proposed by the

REAPER model. Relying on the projector approach, it is however not applicable for high-dimensional data in its original form. To manage such data, we have combined primal-dual optimization techniques from convex analysis with sparse factorization techniques from the Lanczos algorithm. Moreover, we extended the model by penalizing the nuclear norm of the operator and called it rREAPER. This results in the first convex variational method for robust PCA that can handle high-dimensional data since the required memory is reduced from $O(nN + n^2)$ to $O(nN + nr)$, where r is the maximal rank during the primal-dual iteration and has usually the same magnitude as d . The rREAPER minimization algorithm always converges to a global minimizer of the regularized objective. Moreover, using the L -curve method an advantage of the new method seems to be that the dimension of the low-dimensional subspace must not be known in advance, but may be overestimated. We intend to further investigate this direction from the point of view of multi-objective optimization [8,48]. We addressed the problem of the bias in robust PCA, but more research in this directions appears to be necessary. Further other sparsity promoting norms than the nuclear norm could be involved. Our method can be enlarged to 3D images as videos, 3D stacks of medical or material images, where tensor-free methods will come into the play. Finally, it may be interesting to couple PCA ideas with approaches from deep learning to better understand the structure of both.

Acknowledgements The authors want to thank G. Schneck for providing the idea of the proof for Theorem 3 and the unknown reviewers for there valuable comments and suggestions to improve this manuscript. Funding by the German Research Foundation (DFG) within the project STE 571/16-1 and by the Austrian Science Fund (FWF) within the project P28858 is gratefully acknowledged.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Beck, A.: First-Order Methods in Optimization. Society for Industrial and Applied Mathematics (SIAM), Mathematical Optimization Society, Philadelphia (2017)
2. Beck, A., Sabach, S.: Weiszfeld's method: old and new results. *J. Optim. Theory Appl.* **164**(1), 1–40 (2015)
3. Burger, M., Sawatzky, A., Steidl, G.: First order algorithms in variational image processing. In: Glowinski, R., Osher, S.J., Yin, W. (eds.) *Splitting Methods in Communication, Imaging, Science, and Engineering*, pp. 345–407. Springer, Cham (2016)
4. Candes, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? *J. ACM* **58**(3), 11 (2011)
5. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
6. Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numerica* **25**, 161–319 (2016)
7. Cherapanamjeri, Y., Jain, P., Netrapalli, P.: Thresholding based efficient outlier robust PCA. *Proc. Mach. Learn. Res.* **65**, 1–36 (2017)
8. Ciak, R., Shafei, B., Steidl, G.: Homogeneous penalizers and constraints in convex image restoration. *J. Math. Imaging Vis.* **47**(3), 210–230 (2013)
9. Ding, C., Zhou, D., He, X., Zha, H.: R_1 -PCA: rotational invariant L_1 -norm principal component analysis for robust subspace factorization. In: *Proceedings of the 23rd International Conference on Machine Learning*, pp. 281–288. ACM, New York (2006)
10. Epstein, R., Hallinan, P.W., Yuille, A.L.: 5 ± 2 eigenimages suffice: an empirical investigation of low-dimensional lighting models. In: *Proceedings of the Workshop on Physics-Based Modeling in Computer Vision*, 18–19 June 1995, Cambridge, USA, pp. 108–116 (1995)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Readings in Computer Vision*, pp. 726–740. Elsevier, Amsterdam (1987)
12. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(6), 643–660 (2001)
13. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, 4th edn. The John Hopkins University Press, Baltimore (2013)
14. Hanke, M., Hansen, P.C.: Regularization methods for large scale problems. *Surv. Ind. Math.* **3**, 253–264 (1993)
15. Hansen, P.C.: The L -curve and its use in the numerical treatment of inverse problems. In: Johnston, P. (ed.) *Computational Inverse Problems in Electrocardiology*, *Advances in Computational Bioengineering*, pp. 119–142. WIT Press, Southampton (2000)
16. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, Cambridge (1991)
17. Huber, P.J., Ronchetti, E.M.: *Robust Statistics*, 2nd edn. Wiley, New York (2009)
18. Ke, Q., Kanade, T.: Robust ℓ_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 739–746. IEEE (2005)
19. Kent, J.T., Tyler, D.E., Vard, Y.: A curious likelihood identity for the multivariate t -distribution. *Commun. Stat. Simul. Comput.* **23**(2), 441–453 (1994)
20. Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A.: A general framework for increasing the robustness of PCA-based correlation clustering algorithms. In: *Scientific and Statistical Database Management. Lecture Notes in Computer Science*, vol. 5069, pp. 418–435 (2008)
21. Kwak, N.: Principal component analysis based on L_1 -norm maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(9), 1672–1680 (2008)
22. Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand.* **45**(4), 255–282 (1950)

23. Lee, K., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(5), 684–698 (2005)
24. Lellmann, J., Neumayer, S., Nimmer, M., Steidl, G.: Methods for finding the offset in robust subspace fitting. *PAMM* **19**(1) (2019)
25. Lerman, G., Maunu, T.: Fast, robust and non-convex subspace recovery. *Inf. Inference J. IMA* **7**(2), 277–336 (2018). <https://doi.org/10.1093/imaia/iax012>
26. Lerman, G., Maunu, T.: An overview of robust subspace recovery. *Proc. IEEE* **106**(8), 1380–1410 (2018)
27. Lerman, G., McCoy, M., Tropp, J.A., Zhang, T.: Robust computation of linear models by convex relaxation. *Found. Comput. Math.* **15**(1), 363–410 (2015)
28. Leroy, A.M., Rousseeuw, P.J.: Robust regression and outlier detection. Wiley Series in Probability and Mathematical Statistics (1987)
29. Li, C.K., Mathias, R.: The Lidskii–Mirsky–Wielandt theorem—additive and multiplicative versions. *Numer. Math.* **81**, 377–413 (1999)
30. Liu, Y., Wang, S., Juhe, S.: Finding the projection onto the intersection of a closed half-space and a variable box. *Oper Res Lett* **41**, 259–315 (1993). <https://doi.org/10.1016/j.orl.2013.01.014>
31. Maunu, T., Zhang, T., Lerman, G.: A well-tempered landscape for non-convex robust subspace recovery. *J. Mach. Learn. Res.* **20**(37), 1–59 (2019)
32. McCoy, M., Tropp, J.A.: Two proposals for robust PCA using semidefinite programming. *Electron. J. Stat.* **5**, 1123–1160 (2011)
33. Meng, X.L., Van Dyk, D.: The EM algorithm—an old folk-song sung to a fast new tune. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **59**(3), 511–567 (1997)
34. Neumayer, S., Nimmer, M., Setzer, S., Steidl, G.: On the rotational invariant l_1 -norm PCA. *Linear Algebra Appl.* **587**, 243–270 (2020)
35. Neumayer, S., Nimmer, M., Setzer, S., Steidl, G.: On the robust PCA and Weiszfeld’s algorithm. *Appl. Math. Optim.* (To appear)
36. Ostresh Jr., L.M.: On the convergence of a class of iterative methods for solving the Weber location problem. *Oper. Res.* **26**(4), 597–609 (1978)
37. Pearson, K.: On lines and planes of closest fit to systems of points in space. *Phil. Mag.* **2**(11), 559–572 (1901)
38. Podosinnikova, A., Setzer, S., Hein, M.: Robust PCA: optimization of the robust reconstruction error over the Stiefel manifold. In: German Conference on Pattern Recognition, pp. 121–131. Springer, Berlin (2014)
39. Rasch, J., Chambolle, A.: Inexact first-order primal-dual algorithms. *Comput. Optim. Appl.* **76**(2), 381–430 (2020)
40. Rockafellar, R.T.: *Convex Analysis*. Princeton University Press, Princeton (1970)
41. Rousseeuw, P.J., Leroy, A.M.: Robust regression and outlier detection, vol. 589. John Wiley & Sons, New York (2005)
42. Setzer, S., Steidl, G., Teuber, T.: On vector and matrix median computation. *J. Comput. Appl. Math.* **236**(8), 2200–2222 (2012)
43. Stewart, G.W.: Accelerating the orthogonal iteration for the eigenvectors of a Hermitian matrix. *Numer. Math.* **13**(4), 362–376 (1969)
44. Tsakiris, M.C., Vidal, R.: Dual principal component pursuit. *J. Mach. Learn. Res. (JMLR)* **19**, 684–732 (2018)
45. Turk, M., Pentland, A.: Eigenfaces for recognition. *J. Cogn. Neurosci.* **3**(1), 71–86 (1991)
46. Tyler, D.E.: A distribution-free M -estimator of multivariate scatter. *Ann. Stat.* **15**(1), 234–251 (1987)
47. Tyler, D.E.: A distribution-free M -estimator of multivariate scatter. *Ann. Stat.* **15**(1), 234–251 (1987)
48. van den Berg, E., Friedlander, M.P.: Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* **31**(2), 890–912 (2008)
49. Weiszfeld, E.: Sur le point pour lequel les sommes des distances de n points donnés et minimum. *Tôhoku Math. J.* **43**, 355–386 (1937)
50. Wu, K., Simon, H.: Thick-restart Lanczos method for large symmetric eigenvalue problems. *SIAM J. Matrix Anal. Appl.* **22**(2), 602–616 (2000)
51. Xu, H., Caramanis, C., Sanghavi, S.: Robust PCA via outlier pursuit. *IEEE Trans. Inf. Theory* **58**(3), 3047–3064 (2012)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Robert Beinert received his PhD in Mathematics from the University of Göttingen (Germany). After a Postdoc at the University of Graz (Austria), he is currently a research fellow at the TU Berlin (Germany). His research interests include inverse problems, especially bilinear and quadratic formulations, as well as harmonic and convex analysis with applications in signal and image processing like phase retrieval.



Gabriele Steidl received her PhD and Habilitation in Mathematics from the University of Rostock (Germany). After positions as associated professor for Mathematics at the TU Darmstadt and full professor at the University of Mannheim and TU Kaiserslautern (Germany), she is currently professor at the TU Berlin (Germany). She was a Postdoc, resp. visiting Professor at the Univ. of Debrecen, Zürich, ENS Cachan/Paris Univ. Paris Est, Sorbonne/IHP Paris and consultant of the Fraunhofer ITWM Kaiserslautern. Gabriele Steidl is in the Editorial Board of various journals among them SIAM Imaging Sciences, JMIV, IPI, Journal of Fourier Analysis and its Applications, and serves as program director of SIAG IS. Her research interests include harmonic and convex analysis with applications in image processing as well as non-smooth optimization.