A constrained singular value decomposition method that integrates sparsity and orthogonality

Vincent Guillemot, Derek Beaton, Arnaud Gloaguen, Tommy Lofstedt, Brian Levine, Nicolas Raymond, Arthur Tenenhaus, Hervé Abdi

PLOS ONE

RESEARCH ARTICLE

A constrained singular value decomposition method that integrates sparsity and orthogonality

Vincent Guillemot[1]*, Derek Beaton[2], Arnaud Gloaguen[3], Tommy Löfstedt[4], Brian Levine[2], Nicolas Raymond[5], Arthur Tenenhaus[3], Hervé Abdi[6]*

1 Bioinformatics and Biostatistics Hub, Institut Pasteur, Paris, France, **2** The Rotman Research Institute Institution at Baycrest, Toronto, ON, Canada, **3** L2S, UMR CNRS 8506, CNRS–CentraleSupélec–Université Paris-Sud, Université Paris-Saclay, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France, **4** Department of Radiation Sciences, Umeå University, Umeå, Sweden, **5** IRMAR, UMR 6625, Université de Rennes, Rennes, France, **6** School of Behavioral and Brain Sciences, The University of Texas at Dallas, Richardson, TX, United States of America

* vincent.guillemot@pasteur.fr (VG); herve@utdallas.edu (HA)

Citation: Guillemot V, Beaton D, Gloaguen A, Löfstedt T, Levine B, Raymond N, et al. (2019) A constrained singular value decomposition method that integrates sparsity and orthogonality. PLoS ONE 14(3): e0211463. https://doi.org/10.1371/journal.pone.0211463

Editor: Shyamal D Peddada, University of Pittsburgh Graduate School of Public Health, UNITED STATES

Received: June 30, 2018

Accepted: January 15, 2019

Published: March 13, 2019

Data Availability Statement: The datasets whose analyses are presented in the article are publicly available on GitHub (https://github.com/HerveAbdi/data4PCCAR). Due to confidentiality issues, the original OSIQ dataset cannot be made public. Instead, a simulated example is provided based on the original dataset. The simulation process is described in the paper.

Funding: This article benefitted from a EURIAS fellowship at the Paris Institute for Advanced

Abstract

We propose a new sparsification method for the singular value decomposition—called the constrained singular value decomposition (CSVD)—that can incorporate multiple constraints such as sparsification and orthogonality for the left and right singular vectors. The CSVD can combine different constraints because it implements each constraint as a projection onto a convex set, and because it integrates these constraints as projections onto the intersection of multiple convex sets. We show that, with appropriate sparsification constants, the algorithm is guaranteed to converge to a stable point. We also propose and analyze the convergence of an efficient algorithm for the specific case of the projection onto the balls defined by the norms L_1 and L_2. We illustrate the CSVD and compare it to the standard singular value decomposition and to a non-orthogonal related sparsification method with: 1) a simulated example, 2) a small set of face images (corresponding to a configuration with a number of variables much larger than the number of observations), and 3) a psychometric application with a large number of observations and a small number of variables. The companion `R`-package, `csvd`, that implements the algorithms described in this paper, along with reproducible examples, are available for download from https://github.com/vguillemot/csvd.

Introduction

The singular value decomposition (SVD) [1–3]—the tool "par excellence" of multivariate statistics—constitutes the core of many multivariate methods such as, to name but a few, principal component analysis [4], canonical correlation analysis [5], multiple correspondence analysis [6], and partial least squares methods [7]. To analyze data tables whose rows typically correspond to observations and columns to variables, these statistical methods use the SVD to generate *orthogonal* optimal linear combinations of the variables—called components or factor

Studies (France), co-funded by Marie Skłodowska-Curie Actions, under the European Union's 7th Framework Programme for research, and from a funding from the French State programme "Investissements d'avenir", managed by the Agence Nationale de la Recherche (ANR-11-LABX-0027-01 Labex RFIEA+). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

scores—that extract the most *important* information in the original data. In most cases, only the components that explain the largest proportion of the data variance are kept for further investigation. The coefficients—called loadings—of the linear combination used to compute a component are also often used to understand or "interpret" the corresponding components and this interpretation is greatly facilitated (particularly when the number of variables is large) when, for a given component, only a few variables have large loadings and the other variables have negligible loadings. If this pattern does not naturally hold, several procedures can be used to select the variables that are important for a component. The early psychometric school, for example, would use rotations, such as VARIMAX, [8] of the components in a low dimensional subspace; whereas recent approaches favor computationally based methods such as bootstrap ratios [7], or select important variables using an explicit non-linear optimization method such as the LASSO [9]. Closely related to the current work, in the specific case of principal component analysis (for an extensive review of sparsification for PCA see [10]), Witten et al. (see Section 3.2 in [11]) propose to implement either an orthogonality constraint, or a sparsity constraint (but not both simultaneously, see also, for related ideas, [12, 13]). Along the same lines, Benidis et al. [14] proposed, recently, an algorithm, based on Procrustes approach, for sparse principal component analysis that includes an orthogonality constraint on the loadings.

Unfortunately, in the more general case of having concurrently the sparsity and the orthogonality constraints active on both left and right pseudo-singular vectors, the components obtained from the LASSO and its derivatives are not orthogonal and this often makes their interpretation difficult. To palliate this problem, we present and illustrate a new LASSO-like sparsification method for the SVD, called *constrained singular value decomposition* (CSVD), that incorporates orthogonality constraints on both the rows and the columns of a matrix.

1 Notations

Matrices are denoted by uppercase bold letters (e.g., $\mathbf{X}$), vectors by lowercase bold (e.g., $\mathbf{x}$), and their elements by lower case italic (e.g., $x_{i,j}$). Matrices, vectors, and elements from the same matrix all use the same letter (e.g., $\mathbf{A}$, $\mathbf{a}$, a). The transpose operation is denoted by the superscript "$^\top$", the inverse of a square matrix $\mathbf{A}$ is denoted by $\mathbf{A}^{-1}$. The identity matrix is denoted $\mathbf{I}$, vectors or matrices of ones are denoted by $\mathbf{1}$, matrices or vectors of zeros are denoted by $\mathbf{0}$ (when multiplied with or added to other matrices, matrices $\mathbf{I}$, $\mathbf{1}$, and $\mathbf{0}$ are assumed to be conformable, in case of doubt their size is specified). When provided with a square matrix, the diag(·) operator returns a vector with the diagonal elements of the matrix, and when provided with a vector, the diag(·) operator returns a diagonal matrix with the elements of the vector as the diagonal elements of this matrix. When provided with a square matrix, the trace(·) operator gives the sum of the diagonal elements of this matrix. The L_2 norm of vector $\mathbf{x}$, denoted $\|\mathbf{x}\|_2$ is defined as $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top\mathbf{x}}$, The L_1 norm of vector $\mathbf{x}$, denoted $\|\mathbf{x}\|_1$ is defined as $\|\mathbf{x}\|_1 = \Sigma(|x_n|)$. A vector $\mathbf{x}$ is *normalized* by dividing this vector by its L_2 norm (and so a normalized vector has an L_2 norm equal to 1). The Frobenius norm of matrix $\mathbf{X}$, denoted $\|\mathbf{X}\|_F$ is defined as $\|\mathbf{X}\|_F^2 = \text{trace}\,(\mathbf{X}^\top\mathbf{X})$. The Frobenius inner product of two rectangular matrices $\mathbf{A}$ and $\mathbf{B}$ of same dimensions, denoted $\langle\mathbf{A}, \mathbf{B}\rangle_F$ is defined as $\langle\mathbf{A}, \mathbf{B}\rangle_F = \text{trace}\,(\mathbf{A}\mathbf{B}^\top)$. The concatenation of an I by J matrix $\mathbf{X}$ and an I by 1 vector $\mathbf{y}$ is the I by $J + 1$ matrix denoted $[\mathbf{X}, \mathbf{y}]$ obtained by the juxtaposition of $\mathbf{y}$ on the right side of matrix $\mathbf{X}$. The orthogonal complement of the space linearly spanned by the columns of a rectangular matrix $\mathbf{M}$ is denoted $\mathbf{M}^\perp$. Two rectangular matrices $\mathbf{A}$ and $\mathbf{B}$ of same dimensions are said to be orthogonal if and only if $\langle\mathbf{A}, \mathbf{B}\rangle_F = 0$.

With x being a real scalar and γ a non-negative real number, the *scalar soft-thresholding function* denoted $s(x, \gamma)$ is defined as

$$s(x,\gamma) = \begin{cases} x+\gamma & \text{if } x < -\gamma, \\ 0 & \text{if } |x_i| \leq \gamma, \\ x-\gamma & \text{if } x > \gamma; \end{cases} \tag{1}$$

and the *vector soft-thresholding function* denoted $S(\mathbf{x}, \gamma)$ is defined as:

$$S(\mathbf{x},\gamma) = \begin{bmatrix} s(x_1,\gamma) \\ \vdots \\ s(x_N,\gamma) \end{bmatrix}. \tag{2}$$

This function shrinks all the components of $\mathbf{x}$ toward 0, and set the smallest components to 0.

The projection of a vector $\mathbf{x}$ onto a space $\mathscr{S}$ is denoted by proj($\mathbf{x}$, $\mathscr{S}$). The L_2-ball of radius ρ, denoted $\mathscr{B}_{L_2}(\rho)$, is defined as

$$\mathscr{B}_{L_2}(\rho) = \{\mathbf{x} \,|\, \|\mathbf{x}\|_2 \leq \rho\} \tag{3}$$

and the L_1-ball of radius ρ, denoted denoted $\mathscr{B}_{L_1}(\rho)$, is defined as

$$\mathscr{B}_{L_1}(\rho) = \{\mathbf{x} \,|\, \|\mathbf{x}\|_1 \leq \rho\}. \tag{4}$$

Singular values are denoted δ, eigenvalues are denoted $\lambda = \delta^2$.

2 Unconstrained singular value decomposition

The SVD of a data matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ of rank $L \leq \min(I, J)$ gives the solution of the following problem: Find a least-squares optimal, rank R (with $R \leq L$) approximation of $\mathbf{X}$, denoted $\hat{\mathbf{X}}_{[R]}$. Specifically, the SVD solves the following optimization problem [1, 2, 15]:

$$\underset{\hat{\mathbf{X}}_{[R]} \in \mathcal{M}_{I,J}(R)}{\arg\min} \ \frac{1}{2}\|\mathbf{X} - \hat{\mathbf{X}}_{[R]}\|_F^2 = \underset{\hat{\mathbf{X}}_{[R]} \in \mathcal{M}_{I,J}(R)}{\arg\min} \ \frac{1}{2}\left\{\text{trace}\left((\mathbf{X} - \hat{\mathbf{X}}_{[R]})^\top \left(\mathbf{X} - \hat{\mathbf{X}}_{[R]}\right)\right)\right\}, \tag{5}$$

where $\mathcal{M}_{I,J}(R)$ is the set of all real $I \times J$ matrices of rank R.

Recall that the SVD decomposes $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{P}\boldsymbol{\Delta}\mathbf{Q}^\top, \tag{6}$$

where $\mathbf{P}^\top\mathbf{P} = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$ and $\boldsymbol{\Delta} = \text{diag}(\boldsymbol{\delta})$ with $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_L > 0$, and L is the rank of $\mathbf{X}$. The matrix $\mathbf{P} \in \mathbb{R}^{I \times L}$ (respectively $\mathbf{Q} \in \mathbb{R}^{J \times L}$) contains the left (respectively right) singular vectors of $\mathbf{X}$ and the diagonal matrix $\boldsymbol{\Delta}$ contains the singular values of $\mathbf{X}$. If $\mathbf{p}_\ell$ (respectively $\mathbf{q}_\ell$) denotes the ℓ-th column of $\mathbf{P}$ (respectively $\mathbf{Q}$), and δ_ℓ the ℓ-th element of $\boldsymbol{\delta}$, then, for any $R \leq L$, the optimal matrix $\hat{\mathbf{X}}_{[R]}$ is obtained as:

$$\hat{\mathbf{X}}_{[R]} = \sum_{\ell=1}^{R} \delta_\ell \mathbf{p}_\ell \mathbf{q}_\ell^\top \tag{7}$$

with $\mathbf{p}_\ell^\top \mathbf{p}_\ell = \mathbf{q}_\ell^\top \mathbf{q}_\ell = 1$, and $\mathbf{q}_\ell^\top \mathbf{q}_{\ell'} = \mathbf{p}_\ell^\top \mathbf{p}_{\ell'} = 0, \forall\, \ell \neq \ell'$.

A classic, albeit non-optimal and potentially numerically unstable, algorithm (described in Algorithm 1) to obtain the unconstrained singular value decomposition of $\mathbf{X}$ is based on the "power iteration method." This algorithm—originally developed for the eigen-decomposition of a square matrix—provides the first singular triplet that comprises the first singular value

and first left and right singular vectors. In order to ensure orthogonality between successive singular vectors, the first rank one approximation of $\mathbf{X}$, computed as

$$\hat{\mathbf{X}}_{[1]} = \delta_1 \mathbf{p}_1 \mathbf{q}_1^\top, \tag{8}$$

is subtracted from $\mathbf{X}$. This procedure—called deflation (see Appendix A)—gives a new matrix

$$\mathbf{X}^{(2)} = \mathbf{X} - \delta_1 \mathbf{p}_1 \mathbf{q}_1^\top, \tag{9}$$

which is orthogonal to $\hat{\mathbf{X}}_{[1]}$. The power iteration method is then applied to the deflated matrix $\mathbf{X}^{(2)}$, giving a second rank one approximation denoted $\hat{\mathbf{X}}_{[3]} = \delta_2 \mathbf{p}_2 \mathbf{q}_2^\top$. The deflation is then applied to $\mathbf{X}^{(2)}$ to give the new residual matrix $\mathbf{X}^{(3)}$ orthogonal to $\mathbf{X}^{(2)}$, and so on, until nothing is left to subtract because, then, $\mathbf{X}$ has been completely decomposed. This way, the optimization problem from Eq 5 can be re-expressed as:

$$\begin{aligned} &\underset{\substack{\delta_\ell,\, \mathbf{p}_\ell,\, \mathbf{q}_\ell \\ \ell=1,\ldots,R}}{\arg\min} \frac{1}{2} \left\| \mathbf{X} - \sum_{\ell=1}^{R} \delta_\ell \mathbf{p}_\ell \mathbf{q}_\ell^\top \right\|_F^2 \\ &\text{subject to} \begin{cases} \mathbf{p}_\ell^\top \mathbf{p}_\ell = 1, \\ \mathbf{p}_\ell^\top \mathbf{p}_{\ell'} = 0, \\ \mathbf{q}_\ell^\top \mathbf{q}_\ell = 1, \\ \mathbf{q}_\ell^\top \mathbf{q}_{\ell'} = 0, \end{cases} \quad \forall \ell' \neq \ell. \end{aligned} \tag{10}$$

Algorithm 1: The power iteration method for the unconstrained SVD. The algorithm consists in alternating the multiplication of the data matrix by the left and right vectors followed by a normalization step. After convergence, the data matrix is deflated and the process is re-iterated.

```
Data: X, ε, R
Results: SVD of X
Define X^(1) = X;
for ℓ = 1, ..., R do
  p^(0) and q^(0) are randomly initialized;
  δ^(0) = 0;
  δ^(1) = p^(0)⊤ X q^(0);
  s = 0;
  while |δ^(s+1) − δ^(s)| ≥ ε do
    p^(s+1) ← normalize (X q^(s));
    q^(s+1) ← normalize (X⊤ p^(s+1));
    δ^(s+1) = p^(s+1)⊤ X q^(s+1);
    s ← s + 1;
  end
  X^(ℓ+1) = X^(ℓ) − δ^(s) p^(s) q^(s)⊤;
end
```

As an alternative to the deflation approach used in Algorithm 1, the orthogonality constraint can be eliminated and integrated into the power iteration algorithm by replacing the normalization steps by the projection of the result of the current iteration onto the intersection of the L_2 ball and the space orthogonal to the previously found left or right singular vectors (see Algorithm 2). This projection onto the intersection of these two spaces can be implemented in a number of ways [16], we chose here to use the projection onto convex sets algorithm (POCS, see, e.g., [17, Page 101])—an iterative algorithm easily implementable and generalizable to the projection onto the intersection of more than two convex sets (see

Algorithm 3). Recall that, when normalized, a vector $\mathbf{x}$ is projected onto the L_2 ball and that a vector $\mathbf{x}$ is projected onto $\mathbf{V}^{\perp}$ by multiplying it by $\mathbf{I} - \mathbf{V}^{\top}\mathbf{V}$. In POCS, these two projection steps are iterated until convergence.

Algorithm 2 An alternative algorithm of the power iteration for the unconstrained SVD: The deflation step is replaced by a projection onto the space orthogonal to the space defined by the already computed lower rank version of the data matrix. Note that $\mathbf{0}^{\perp}$ corresponds to the whole space, so it is either $\mathbb{R}^I$ or $\mathbb{R}^J$.

Data: $\mathbf{X}$, ε, R
Result: SVD of $\mathbf{X}$
Define $\mathbf{P} = \mathbf{0}$;
Define $\mathbf{Q} = \mathbf{0}$;
for $\ell = 1, \ldots, R$ **do**
  $\mathbf{p}^{(0)}$ and $\mathbf{q}^{(0)}$ are randomly initialized;
  $\delta^{(0)} \leftarrow 0$;
  $\delta^{(1)} \leftarrow \mathbf{p}^{(0)\top}\mathbf{X}\mathbf{q}^{(0)}$;
  $s \leftarrow 0$;
  while $|\delta^{(s+1)} - \delta^{(s)}| \geq \varepsilon$ **do**
    $\mathbf{p}^{(s+1)} \leftarrow \text{proj}(\mathbf{X}\mathbf{q}^{(s)}, \mathscr{B}_{L_2}(1) \cap \mathbf{P}^{\perp})$;
    $\mathbf{q}^{(s+1)} \leftarrow \text{proj}(\mathbf{X}^{\top}\mathbf{p}^{(s+1)}, \mathscr{B}_{L_2}(1) \cap \mathbf{Q}^{\perp})$;
    $\delta^{(s+1)} \leftarrow \mathbf{p}^{(s+1)\top}\mathbf{X}\mathbf{q}^{(s+1)}$;
    $s \leftarrow s+1$;
  end
  $\delta_{\ell} \leftarrow \delta^{(s+1)}$;
  $\mathbf{P} \leftarrow [\mathbf{P}, \mathbf{p}^{(s+1)}]$;
  $\mathbf{Q} \leftarrow [\mathbf{Q}, \mathbf{q}^{(s+1)}]$;
end

Algorithm 3 Projection onto the intersection of K convex sets (POCS).

Data: $\mathbf{x}$, $\mathscr{S}_1, \ldots, \mathscr{S}_K$, ε
Results: Projection of $\mathbf{x}$ onto $\bigcap_{k=1}^{K} \mathscr{S}_k$
Define $\mathbf{x}^{(0)} = \mathbf{x}$;
while $|\mathbf{x}^{(s+1)} - \mathbf{x}^{(s)}| \geq \varepsilon$ **do**
  $\mathbf{x}^{(s+1)} \leftarrow \text{proj}(\mathbf{x}^{(s)}, \mathscr{S}_1))$;
  for $k = 2, \ldots, K$ **do**
    $\mathbf{x}^{(s+1)} \leftarrow \text{proj}(\mathbf{x}^{(s+1)}, \mathscr{S}_k))$;
  end
  $s \leftarrow s + 1$
end

Algorithm 1 is obviously faster than Algorithm 2 as implemented using Algorithm 3, because the orthogonality constraint in Algorithm 1 is performed with one operation whereas Algorithm 2 always requires several operations. However, the main benefit of Algorithm 2 is that it easily can be extended to include additional constraints, as illustrated below.

3 Constrained singular value decomposition

3.1 Previous work

Algorithm 4: The Algorithm of Witten et al. [11]: The penalized matrix decomposition (PMD) approach.

Data: $\mathbf{X}$, ε, R
Resulet: SVD of $\mathbf{X}$
Define $\mathbf{X}^{(1)} = \mathbf{X}$;
for $\ell = 1, \ldots, R$ **do**
  $\mathbf{p}^{(0)}$ and $\mathbf{q}^{(0)}$ are randomly initialized;
  $\delta^{(0)} = 0$;

$\delta^{(1)} = \mathbf{p}^{(0)\top}\mathbf{X}\mathbf{q}^{(0)}$;
$s = 0$;
while $|\delta^{(s+1)} - \delta^{(s)}| \geq \varepsilon$ **do**
$\mathbf{p}^{(s+1)} \leftarrow$ normalize $(S(\mathbf{X}\mathbf{q}^{(s)}, \lambda_{1,\ell}))$, with $\lambda_{1,\ell}$ such that $\|\mathbf{p}^{(s+1)}\|_1 = c_{1,\ell}$;
$\mathbf{q}^{(s+1)} \leftarrow$ normalize $(S(\mathbf{X}^\top \mathbf{p}^{(s+1)}, \lambda_{2,\ell}))$, with $\lambda_{2,\ell}$ such that $\|\mathbf{q}^{(s+1)}\|_1 = c_{2,\ell}$;
$\delta^{(s+1)} = \mathbf{p}^{(s+1)\top}\mathbf{X}\mathbf{q}^{(s+1)}$;
$s \leftarrow s + 1$;
end
$\mathbf{X}^{(\ell+1)} = \mathbf{X}^{(\ell)} - \delta^{(s)}\mathbf{p}^{(s)}\mathbf{q}^{(s)\top}$;
end

Recently, several authors have proposed sparse variants of the SVD (see e.g., [4, 11, 15, 18, 19] for reviews), or, more specifically, of PCA [14, 20]. For most of these sparse variants, the sparsification is obtained by adding new constraints to Eq 10. For example, the penalized matrix decomposition (PMD) approach by Witten et al. [11] solves the following optimization problem for the first pair of left and right singular vectors:

$$\underset{\substack{\delta_\ell,\mathbf{p}_\ell,\mathbf{q}_\ell \\ \ell=1,\ldots,R}}{\arg\min} \frac{1}{2}\left\|\mathbf{X} - \sum_{\ell=1}^{R}\delta_\ell\mathbf{p}_\ell\mathbf{q}_\ell^\top\right\|_F^2$$
$$\text{subject to} \begin{cases} \mathbf{p}_\ell^\top\mathbf{p}_\ell = 1, \\ \mathbf{q}_\ell^\top\mathbf{q}_\ell = 1, \\ C_1(\mathbf{p}_\ell) \leq c_{1,\ell}, \\ C_2(\mathbf{q}_\ell) \leq c_{2,\ell}, \end{cases} \tag{11}$$

where C_1 and C_2 are convex penalty functions from $\mathbb{R}^I$ (respectively $\mathbb{R}^J$) to $\mathbb{R}^+$ (such as, e.g., the LASSO, or the fused LASSO constraints) and with $c_{1,\ell}$ and $c_{2,\ell}$ being positive constants. The PMD procedure is described in Algorithm 4. In PMD, the next pseudo-singular triplet is estimated by solving the same optimization problem where $\mathbf{X}$ is replaced by a deflated matrix. In contrast to Eq 11, however, the added constraints create a nonlinear optimization problem and this makes the deflated matrices non-orthogonal to the previous rank one optimal matrix [21]. This lack of orthogonality makes the interpretation of the components somewhat difficult because the conclusions about one component involve all correlated components and because the same information is explained (to different degrees) by all correlated components. In the specific case of PCA, Witten et al. proposed, alternatively, to impose an orthogonality constraint on the left singular vectors, without the sparsity constraint, and to leave the sparsity constraint active only on the right vectors (i.e., the loadings). However, this procedure does not solve the problem of having both constraints simultaneously active on the left and right singular vectors (See Equation 3.17 and the subsequent algorithm in [11] for more details).

In order to eliminate the problems created by the non-orthogonality of the singular vectors, we present below a new optimal sparsification method, called the *constrained singular value decomposition* (CSVD) that implements orthogonality constraints on successive sparsified singular vectors.

3.2 Current work: The constrained SVD (CSVD)

The constrained SVD still decomposes the matrix $\mathbf{X}$ into singular values and vectors, but imposes, in addition, on the singular vectors constraints that induce sparsity of the weights. Such sparsity-inducing constraints are common in fields where the data comprise large

numbers of variables [22] (e.g., tens of thousands, as in genomics [23], to millions, as in neuroimaging [24, 25]). Although the theory of sparsity-inducing constraints is well documented, we are interested in a general formulation that could also be applied for several types of sparsification, as well as more sophisticated constraints.

Specifically, we consider the following optimization problem:

$$\begin{aligned} &\underset{\substack{\delta_\ell,\mathbf{p}_\ell,\mathbf{q}_\ell \\ \ell=1,\ldots,R}}{\arg\min}\ \frac{1}{2}\left\|\mathbf{X}-\sum_{\ell}^{R}\delta_\ell\mathbf{p}_\ell\mathbf{q}_\ell^\top\right\|_F^2 \\ &\text{subject to}\ \begin{cases}\mathbf{p}_\ell^\top\mathbf{p}_\ell=1,\\ \mathbf{p}_\ell^\top\mathbf{p}_{\ell'}=0,\\ \mathbf{q}_\ell^\top\mathbf{q}_\ell=1,\\ \mathbf{q}_\ell^\top\mathbf{q}_{\ell'}=0,\end{cases}\quad \forall \ell'\neq\ell, \\ &\text{and to}\ \begin{cases}C_1(\mathbf{p}_\ell)\leq c_{1,\ell},\\ C_2(\mathbf{q}_\ell)\leq c_{2,\ell},\end{cases} \end{aligned} \tag{12}$$

where C_1 and C_2 are convex penalty functions from $\mathbb{R}^I$ (respectively $\mathbb{R}^J$) to $\mathbb{R}^+$, (which could be, e.g., the LASSO, group-LASSO, or fused LASSO constraints) and with $c_{1,\ell}$ and $c_{2,\ell}$ being positive constants: smaller values of $c_{1,\ell}$ and $c_{2,\ell}$ lead to solutions that are more sparse. See, however, e.g., [11, 26], and, as developed in Appendix B, only some ranges of values of $c_{1,\ell}$ and $c_{2,\ell}$ will lead to solutions.

An equivalent, but more convenient, form of the problem described in Eq 12 can be derived by considering two orthogonal matrices $\mathbf{P}$, and $\mathbf{Q}$, and a diagonal matrix $\mathbf{\Delta} = \text{diag}\{\boldsymbol{\delta}\}$ such that

$$\frac{1}{2}\|\mathbf{X}-\mathbf{P}\mathbf{\Delta}\mathbf{Q}^\top\|_F^2=\frac{1}{2}\|\mathbf{X}\|_F^2+\frac{1}{2}\sum_\ell\delta_\ell^2-\sum_\ell\delta_\ell\mathbf{p}_\ell^\top\mathbf{X}\mathbf{q}_\ell\ . \tag{13}$$

The term $\|\mathbf{X}\|_F^2$ is constant and $\sum\delta_\ell^2$ does not depend on $\mathbf{p}_\ell$ or $\mathbf{q}_\ell$, and so for a given δ_ℓ positive, the solutions of $\arg\max \delta_\ell\mathbf{p}_\ell^\top\mathbf{X}\mathbf{q}_\ell$ are the same as the solutions of $\arg\max \mathbf{p}_\ell^\top\mathbf{X}\mathbf{q}_\ell$. In addition the maximum is reached when $\delta_\ell=\mathbf{p}_\ell^\top\mathbf{X}\mathbf{q}_\ell$ (see, e.g., [11]). Consequently, minimizing $\|\mathbf{X}-\mathbf{P}\mathbf{\Delta}\mathbf{Q}^\top\|_F^2$ from Eq 13 is equivalent to maximizing each term of the sum $\sum\mathbf{p}_\ell^\top\mathbf{X}\mathbf{q}_\ell$. Therefore, Eq 12 is equivalent to the following 1-dimensional maximization problem for $\ell\geq 1$, given the previous vectors $\mathbf{p}_{\ell'}$ and $\mathbf{q}_{\ell'}$ for all $0\leq\ell'<\ell$:

$$\begin{aligned} &\underset{\mathbf{p},\mathbf{q}}{\arg\max}\ \{\mathbf{p}^\top\mathbf{X}\mathbf{q}\} \\ &\text{subject to}\ \begin{cases}\mathbf{p}^\top\mathbf{p}=1,\\ \mathbf{q}^\top\mathbf{q}=1,\end{cases}\ \text{and}\ \forall\ \ell'<\ell\begin{cases}\mathbf{p}^\top\mathbf{p}_{\ell'}=0,\\ \mathbf{q}^\top\mathbf{q}_{\ell'}=0,\end{cases}\ \text{and}\ \begin{cases}C_1(\mathbf{p})\leq c_{1,\ell},\\ C_2(\mathbf{q})\leq c_{2,\ell},\end{cases} \end{aligned} \tag{14}$$

where we set (for convenience) $\mathbf{p}_0$ and $\mathbf{q}_0$ to $\mathbf{0}$ (note that for the first pseudo-singular vectors, the orthogonality constraint is not active because all vectors are orthogonal to $\mathbf{0}$).

To facilitate the resolution of this optimization problem, the unicity constraint on the L_2 norm of the singular vectors (i.e., $\mathbf{p}^\top\mathbf{p}=1$ and $\mathbf{q}^\top\mathbf{q}=1$) needs to be relaxed and replaced by

an equivalent inequality. Specifically, the optimization problem in Eq 14 is reframed as

$$\begin{aligned} &\arg\max_{\mathbf{p},\mathbf{q}} \{\mathbf{p}^\top \mathbf{X}\mathbf{q}\} \\ &\text{subject to } \begin{cases} \mathbf{p}^\top\mathbf{p} \leq 1, \\ \mathbf{q}^\top\mathbf{q} \leq 1, \end{cases} \text{and } \forall\ \ell' < \ell \begin{cases} \mathbf{p}^\top\mathbf{p}_{\ell'} = 0, \\ \mathbf{q}^\top\mathbf{q}_{\ell'} = 0, \end{cases} \text{and } \begin{cases} C_1(\mathbf{p}) \leq c_{1,\ell}, \\ C_2(\mathbf{q}) \leq c_{2,\ell}. \end{cases} \end{aligned} \tag{15}$$

Eq 15 defines a biconcave maximization problem with convex constraints. This problem can be solved using block relaxation [27]: An iterative algorithm that consists in a series of two-part iterations in which (Part 1) the expression in Eq 15 is maximized for $\mathbf{p}$ with $\mathbf{q}$ being fixed, and is then (Part 2) maximized for $\mathbf{q}$ with $\mathbf{p}$ being fixed. Part 1 of the iteration can be re-expressed as the following optimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{p}} \left\{\frac{1}{2}\|\mathbf{p} - \mathbf{X}\mathbf{q}\|_2^2\right\} \\ &\text{subject to} \begin{cases} \mathbf{p} \in \mathscr{B}_{L_2}(1), \\ \mathbf{p} \in \mathscr{B}_{L_1}(c_1), \\ \mathbf{p} \in \mathbf{P}^\perp. \end{cases} \end{aligned} \tag{16}$$

Eq 16 shows that finding the optimal value for $\mathbf{p}$ (i.e., Part 1 of the alternating procedure) is equivalent to finding the projection of the vector $\mathbf{X}\mathbf{q}$ onto the subspace of $\mathbb{R}^I$ defined by the intersection of all the convex spaces involved by the constraints. During Part 2 of the alternating procedure, the vector $\mathbf{p}$ is fixed and therefore Part 2 can be expressed as the following minimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{q}} \left\{\frac{1}{2}\|\mathbf{q} - \mathbf{X}^\top\mathbf{p}\|_2^2\right\} \\ &\text{subject to} \begin{cases} \mathbf{q} \in \mathscr{B}_{L_2}(\mathbf{1}), \\ \mathbf{q} \in \mathscr{B}_{L_1}(c_2), \\ \mathbf{q} \in \mathbf{Q}^\perp. \end{cases} \end{aligned} \tag{17}$$

Eqs 16 and 17 replace the L_1 and L_2 constraints in the minimization problem expressed in Eq 11 by projections onto the intersection of the convex sets (POCS) defined by these constraints. Because the intersection of several convex sets is also a convex set [28], the block relaxation algorithm from Eq 15 is essentially composed of sequential series of operations applied until convergence of the two projections onto their respective convex sets. This strategy can obviously be extended to incorporate multiple additional constraints as long as these constraints define convex subspaces. As shown in Appendix D, the CSVD algorithm is guaranteed to converge to a stable point because it is a member of the more general class of the *block successive upper-bound minimization (BSUM) algorithms*.

In the specific case of the projection on the intersection of the balls L_1 and L_2, POCS can be replaced by a fast and exact algorithm called PL1L2 ([26], see Appendix C for details). This algorithm (see Algorithm 5) differs from the more general Algorithm 4 only by the specification of the projection method onto the L_1 ball which is implemented as a simple and fast algorithm based on the soft-thresholding operator.

Note that, Algorithm 2—presented in Section 2 for the unconstrained SVD and using POCS for the projection onto an intersection of convex sets—can be easily generalized to incorporate a new sparsity constraint, by simply applying POCS to the intersection of 3 convex

sets (instead of just 2 convex sets): the L_2-ball of radius 1, an L_1-ball, and the orthogonal subspace to the space defined by the previously found left or right pseudo-singular vectors.

Algorithm 5: General algorithm for the Constrained Singular Value Decomposition. The projection onto the L_1-ball can be replaced by another projection onto a convex set, making it possible to adapt this algorithm to other purposes.

```
Data: X, ε, R, c_{1,ℓ} and c_{2,ℓ} for ℓ in 1, ..., R
Results: CSVD of X
Define P = 0;
Define Q = 0;
for ℓ = 1, ..., R do
  p^(0) and q^(0) are randomly initialized;
  δ^(0) ← 0;
  δ^(1) ← p^(0)ᵀ X q^(0);
  s ← 0;
  while |δ^(s+1) − δ^(s)| ≥ ε do
    p^(s+1) ← proj(X q^(s), B1(c_{1,ℓ}) ∩ B2(1) ∩ P^⊥);
    q^(s+1) ← proj(Xᵀ p^(s+1), B1(c_{2,ℓ}) ∩ B2(1) ∩ Q^⊥);
    δ^(s+1) ← p^(s+1)ᵀ X q^(s+1);
    s ← s + 1;
  end
  δ_ℓ ← δ^(s+1);
  P ← [P, p^(s+1)];
  Q ← [Q, q^(s+1)];
end
```

In the following sections, we illustrate—using simulated and real data—the effect and importance of the orthogonality constraint and show how this constraint improves the interpretability of the analysis.

4 Empirical comparative evaluation of the CSVD

In this section, we empirically evaluate, illustrate the constrained singular value decomposition (CSVD), and compare its performance to the performance of the plain SVD and the closely related sparsification method of Witten et al. [11], the PMD method. To do so, we used: 1) some simulated datasets, 2) one simulated dataset mimicking a real dataset, and 3) one real dataset (the characteristics of these datasets are listed in Table 1).

With the first (simulated) dataset we evaluated how the SVD, PMD, and CSVD recover the ground truth for a relatively large dataset with more variables than observations contains a mixture of signal and Gaussian noise.

Datasets two and three were chosen to each illustrate a particular aspect of the data. The second dataset investigates the $N \ll P$ problem and comprises six face images consisting of $230 \times 240 = 55,200$ pixels—with each pixel measuring light intensity on a scale going from 0 to 255. The third dataset illustrates the effects of sparsification on a dataset corresponding to a traditional psychometric problem. This simulated has been created to match the pattern of loadings of a real dataset that was collected from 2, 100 participants who—as part of a larger

Table 1. Characteristics of the various datasets used to assess the performance of the CSVD and related methods.

Dataset	*I* (# of rows)	*J* (# of columns)	Rank
Simulated	150	600	149
Face Data	6	55,200	6
Memory	2,100	30	30

https://doi.org/10.1371/journal.pone.0211463.t001

Table 2. The different values of the sparsity parameters for the CSVD and PMD.

c_1	c_2	Resulting degree of sparsity	Notation
$1+\varepsilon_1$	$1+\varepsilon_2$	The sparsest level, most of the coefficients are close zero	H (High)
$\frac{1}{3}\sqrt{I}$	$\frac{1}{3}\sqrt{J}$	Very sparse	M (Medium)
$\frac{2}{3}\sqrt{I}$	$\frac{2}{3}\sqrt{J}$	Somewhat sparse	L (Low)
$\sqrt{I}$	$\sqrt{J}$	No sparsity, corresponds to the regular SVD	N (None)

https://doi.org/10.1371/journal.pone.0211463.t002

project on memory—answered an online version of the "object-spatial imagery questionnaire" (OSIQ, [29])—a psychometric instrument measuring mental imagery for objects and places. Using a 5-point rating scale, participants rated their agreement for 30 items that should span a 2-dimensional space corresponding to the spatial and object imagery psychometric factors. This dataset is used to compare sparsification and the standard traditional psychometric approach relying on (Varimax) rotation to recover a two dimensional factor structure.

For Datasets two and three, we applied three degrees of sparsity (low, medium, and high). As detailed in Appendix B, only some values of c_1 and c_2 will lead to solutions (specifically, c_1 has to be chosen between 1 and $\sqrt{I}$ and c_2 between 1 and $\sqrt{J}$). Table 2 lists the values chosen for c_1 and c_2 and their interpretation for PMD and the CSVD. Also, for technical reasons, the values of c_1 and c_2 corresponding to the maximum sparsity for $\mathbf{P}$ and $\mathbf{Q}$ are set, respectively, to $1+\varepsilon_1$ and $1+\varepsilon_2$ (instead of 1) with ε_1 and ε_2 being two small real positive values.

4.1 Simulated data

With these simulated data, we evaluate the ability of the CSVD to recover known singular triplets, their sparsity structure, and the orthogonality of the estimated left and right singular vectors. These simulated data were created by adding a matrix of Gaussian noise to a 150 by 600 matrix of rank 5 built from its SVD decomposition.

Specifically, the data matrix $\mathbf{X}$ was created as

$$\mathbf{X} = \mathbf{X}_M + \mathbf{E}, \tag{18}$$

where

- $\mathbf{X}_M = \mathbf{P}_M\boldsymbol{\Delta}_M\mathbf{Q}_M^\top$ is the rank 5 matrix of the ground truth with:
  - $\boldsymbol{\Delta}_M$ a 5×5 the diagonal matrix of the singular values equal to $\boldsymbol{\Delta}_M = \text{diag}(\boldsymbol{\delta}) = \text{diag}([15, 14, 13, 12, 11])$
  - $\mathbf{P}_M$ an $I = 150 \times 5 = 750$ by 5, orthogonal matrix with $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$,
  - $\mathbf{Q}_M$ a $J = 600 \times 5 = 3,000$ by 5 orthogonal matrix with $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$,
- $\mathbf{E}$ is an $I = 150 \times J = 600$ matrix containing $I \times J = 90,000$ independent realizations of a Gaussian variable with mean equal to 0 and standard deviation equal to 0.01.

Matrices $\mathbf{P}_M$ and $\mathbf{Q}_M$ were designed to be both sparse and orthogonal. Specifically, matrix $\mathbf{P}_M$ was generated with the following model

$$\mathbf{P}_M = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 & \mathbf{p}_5 \\ \mathbf{p}'_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{p}'_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{p}'_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{p}'_4 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{p}'_5 \end{bmatrix}, \tag{19}$$

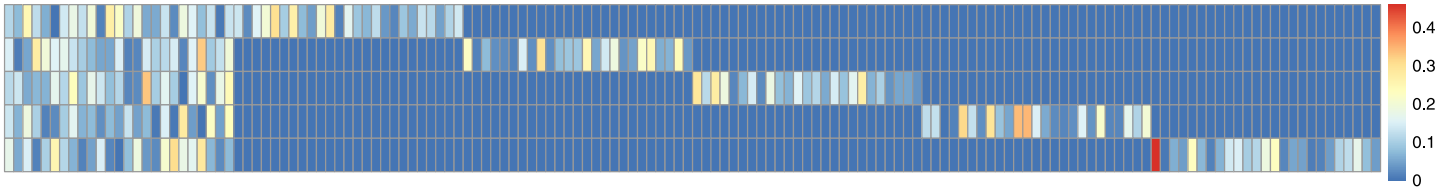

Fig 1. Simulated data. Heatmap of $\mathbf{P}_M^\top$: the *true* left singular vectors (in an horizontal representation: one line equals one singular vector).

https://doi.org/10.1371/journal.pone.0211463.g001

where $\mathbf{0}$ denotes 25×1 vectors of 0s, and where all $\mathbf{p}$ and $\mathbf{p}'$ were 25×1 vectors with norm equal to $2^{-\frac{1}{2}}$ and such that where $\mathbf{p}_\ell^\top \mathbf{p}_{\ell'} = 0, \forall\ \ell \neq \ell'$. A similar model was used for $\mathbf{Q}_M$ which was generated with the following model

$$\mathbf{Q}_M = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 & \mathbf{q}_4 & \mathbf{q}_5 \\ \mathbf{q}_1' & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{q}_2' & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{q}_3' & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{q}_4' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{q}_5' \end{bmatrix}, \tag{20}$$

where $\mathbf{0}$ were 120×1 vectors of 0s, and where all $\mathbf{q}$ $\mathbf{q}'$ vectors were 120×1 vectors with norm equal to $2^{-\frac{1}{2}}$ and such that $\mathbf{q}_\ell^\top \mathbf{q}_{\ell'} = 0, \forall\ \ell \neq \ell'$.

With the structure described in Eqs 19 and 20, the low-rank matrix to recover from $\mathbf{X}$ is then composed of 2 parts: 1) a common part (no sparsity) to all 5 components (i.e., the part corresponding to the $\mathbf{p}$ and $\mathbf{q}$ vectors), and 2) one part specific to each component (i.e., the $\mathbf{p}'$ and the $\mathbf{q}'$ vectors) with the corresponding part in the other 4 components being sparse.

Figs 1 and 2 show heatmaps of the true left-singular vectors ($\mathbf{P}_M$) and right-singular vectors ($\mathbf{Q}_M$).

We analyzed $\mathbf{X}$ with the CSVD and PMD. For both methods, the L_1 constraint was set to $c_1 = 5$ for the left-singular vectors and $c_2 = 11$ for the right-singular vectors, based on the sparsity of the ground truth.

We asked each method to return 7 vectors in order to evaluate if the methods could recover the ground truth (i.e., the 5-dimensional sub-space) but also how they would behave after this 5-dimensional subspace had been recovered. Fig 3 shows the boxplots of the distribution of the squared difference between the estimated singular vectors and the ground truth for PMD and the CSVD: Both methods correctly uncover the singular vectors. Fig 4 shows the correlations between the first 7 estimated singular vectors compared to the ground truth: Although the first 5 singular vectors are correctly estimated, and, roughly, orthogonal to the previous singular vectors, the 2 last vectors estimated by PMD are correlated to some of the previously

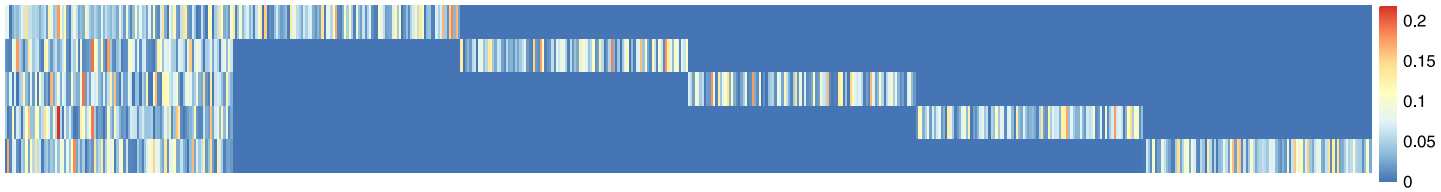

Fig 2. Simulated data. Heatmap of $\mathbf{Q}_M^\top$: the *true* right singular vectors (in an horizontal representation: one line equals one singular vector).

https://doi.org/10.1371/journal.pone.0211463.g002

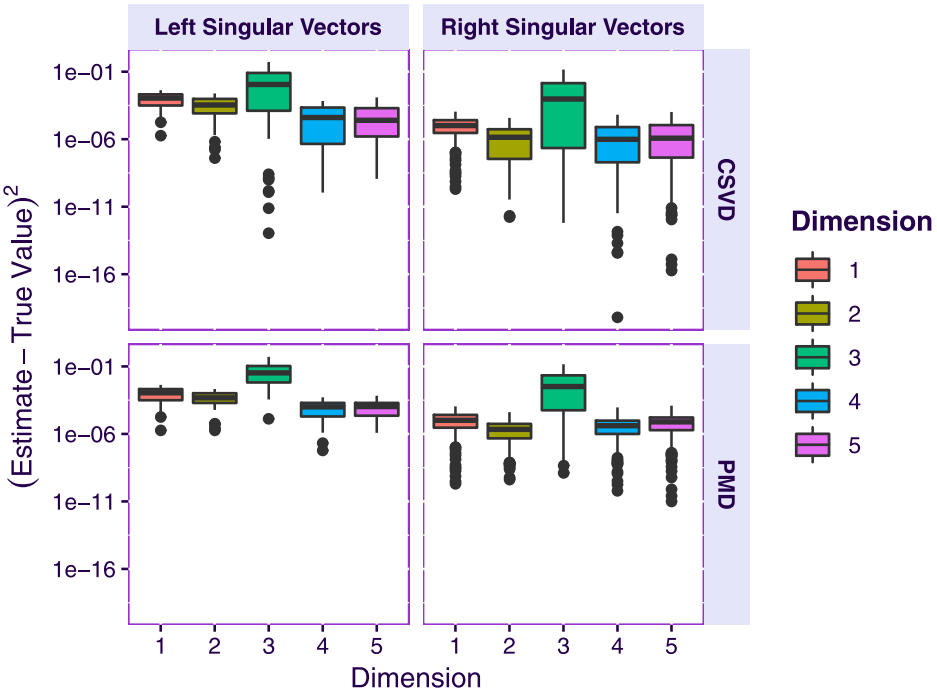

Fig 3. Simulated data. Boxplots of the squared difference between the estimated singular vectors and the ground truth.

https://doi.org/10.1371/journal.pone.0211463.g003

computed vectors. This demonstrates the failure of the standard deflation technique to impose the orthogonality constraint when a non-linear optimization methods is used. By contrast with the PMD approach, the orthogonality constraint of the CSVD prevented this problem.

Fig 5 displays the computational times of the CSVD and PMD as a function of the number of computed components: the CSVD is faster than PMD for the estimation of the first component, but this advantage diminishes when the number of computed components increases. This pattern indicates that the orthogonality constraint increases the computational time.

Table 3 contains the estimated singular values for the CSVD and PMD as well as the ground-truth singular values. The estimated pseudo-singular values are comparable for both methods and behave in a similar way compared to the ground truth and to the regular SVD: The first singular values are slightly smaller than their ground truth values, but at some point—which varies depending on the imposed degree of sparsity—the estimated number of pseudo-singular values is larger than the ground truth.

Additionally, broader settings were considered for the comparison of SVD, CSVD and PMD on simulated data. Specifically, we considered a case with a low signal to noise ratio, and a case where the noise is structured: PMD and CSVD performed equally poorly on noisy data, but were unaffected by a structured noise. These additional results are reported in S1 Table.

To sum up: 1) the CSVD and PMD produce highly similar estimates of the first singular vectors; 2) the CSVD and PMD both recover the true sparsity structure of the ground-truth data; 3) for singular vectors of an order higher than the rank of the matrix, PMD produces singular vectors correlated with the previous ones; and 4) the CSVD is computationally more efficient than PMD but this advantage shrinks as the number of computed components increases.

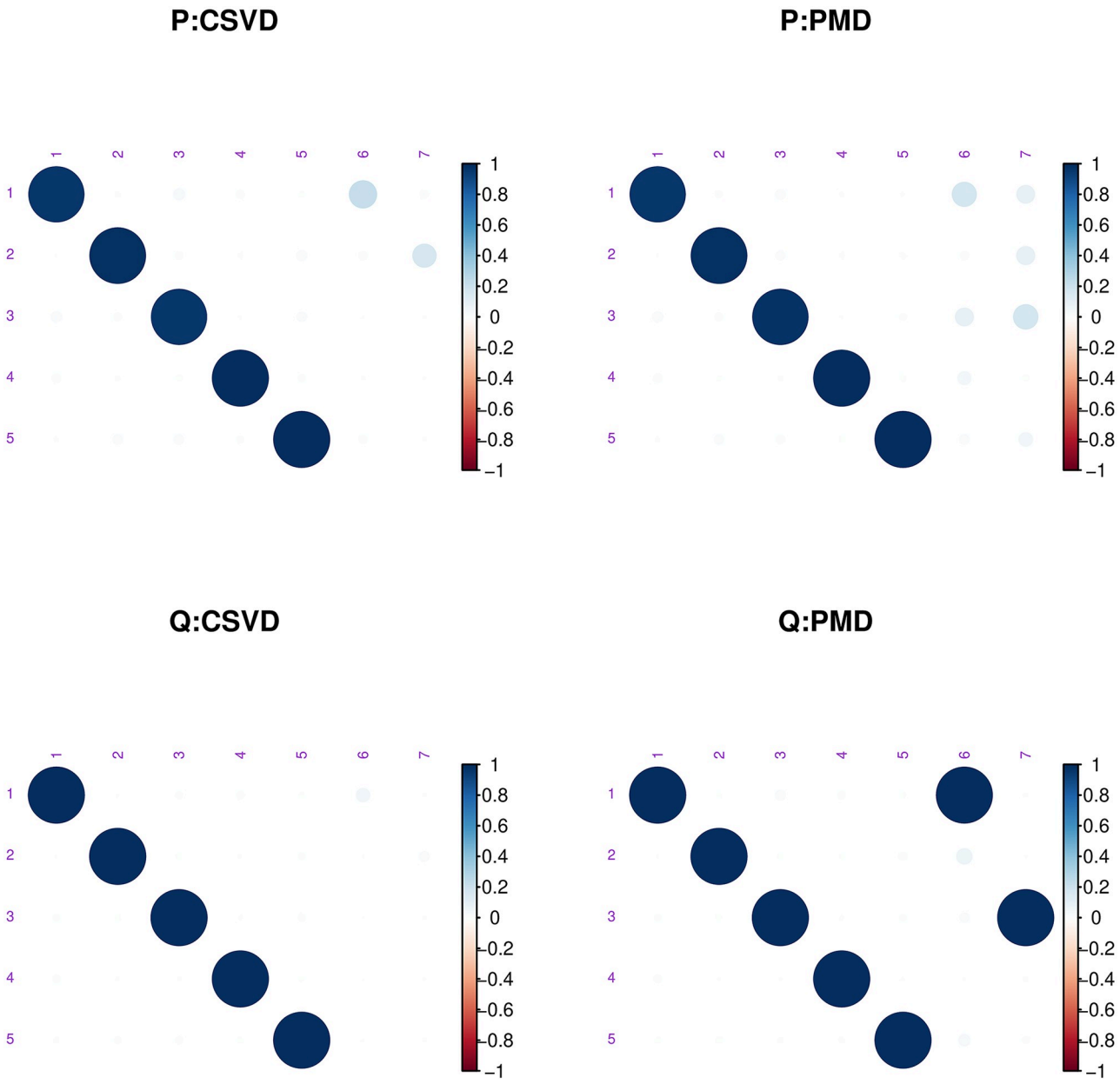

Fig 4. Simulated data. Heatmap of the correlations between the estimated left (**P**) and right (**Q**) singular vectors with the ground truth for the CSVD and PMD. Each cell of the heatmap represents the correlation between one of the 7 estimated (left or right) vectors with the 5 true vectors. Each heatmap contains 5 rows (the ground truth) and 7 columns (the estimated vectors). On the left, are the results obtained with the CSVD and on the right, the results obtained with PMD.

https://doi.org/10.1371/journal.pone.0211463.g004

4.2 The face data

The dataset consists in six 240 × 230 = 55, 200 gray scale digitized (range from 0 to 255) face images (three men and three women) that were extracted from a larger face database (see [30, 31]) and are available as the dataset `sixFaces` from the `R`-package `data4PCCAR` (obtained from the Github repository `HerveAbdi/data4PCCAR`).

Each image was unfolded (i.e., "vectorized") into a 240 × 230 = 55, 200 element vector, which was re-scaled to norm one. A plain SVD was then performed on the 6 × 55, 200 matrix

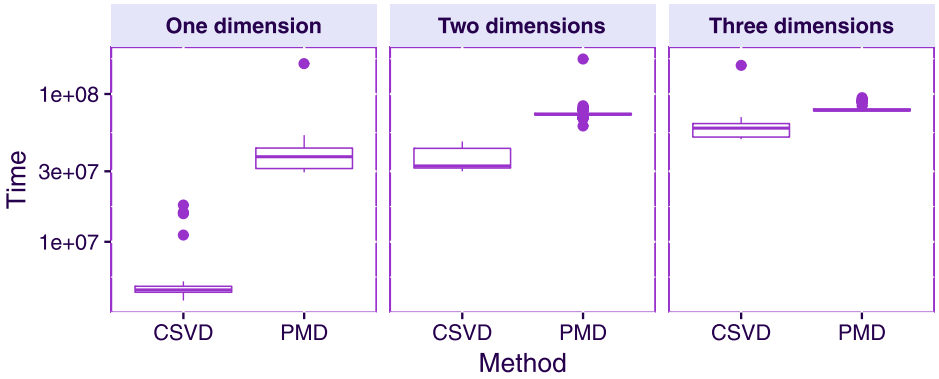

Fig 5. Simulated data. Computational time of PMD and the CSVD when estimating one sparse singular triplet (left) and two sparse singular triplets (right).

https://doi.org/10.1371/journal.pone.0211463.g005

(see Fig 6) obtained from the concatenation of the 6 face vectors. The SVD extracted 6 components with the first one extracting a very large proportion of the total variance (i.e., $\lambda_1 = 5.616$, $\tau_1 = 94\%$, see also Fig 7 left panel). This very large first eigenvalue indicates that these face images are highly correlated (see Fig 8)—An interpretation confirmed by the very similar values of the coordinates of the six faces for the first component. But this large first eigenvalue reflects also, in part, that the data were not centered, because, with all entries of the matrix being positive, the first left (respectively right) singular vector (i.e., the first component) is respectively, the 6-element long vector of a weighted mean of the pixels across the faces (respectively the 55, 200-element long vector of a weighted mean of the faces across the pixels) and so all elements of the first pair of singular vectors have the same sign, see Table 4 and the picture of the first "eigen-face" [32] in the left of the top row of Fig 9). The second component differentiates females and males (see the map of the faces for Components 1 versus 2 in Fig 9, and the picture of the second "eigen-face" in the right of the top row of Fig 9).

We applied the CSVD and PMD to the face set using three different sparsity levels (low, medium, and high). Fig 9 shows the plot of the first two components for these three levels of sparsity. Both the CSVD and PMD tend to isolate the woman faces on the first dimension and the male faces on the second dimension. The corresponding first two eigen-faces, (see Figs 10 and 11) show that both the CSVD and PMD tend to extract characteristic features of the female faces (first eigen-face) or the male faces (second eigen-face). In contrast, the first and second eigen-faces for the plain SVD (plotted in Fig 12) show respectively a weighted average face (i.e., a linear combination with positive coefficients of the faces) and a mixture between male (with positive coefficients) and female (with negative coefficients) faces. This

Table 3. Estimated singular values and ground truth. In the "ground truth" column, a value of 0 indicates that the corresponding real singular value does not exist (i.e., because the underlying matrix has rank 5).

Order	CSVD	PMD	SVD	Ground Truth
1	14.77	14.77	14.97	15.00
2	13.62	13.64	13.99	14.00
3	12.53	12.59	12.99	13.00
4	11.86	11.90	11.97	12.00
5	10.34	10.45	10.93	11.00
6	0.21	2.94	0.04	0
7	0.15	2.98	0.04	0

https://doi.org/10.1371/journal.pone.0211463.t003

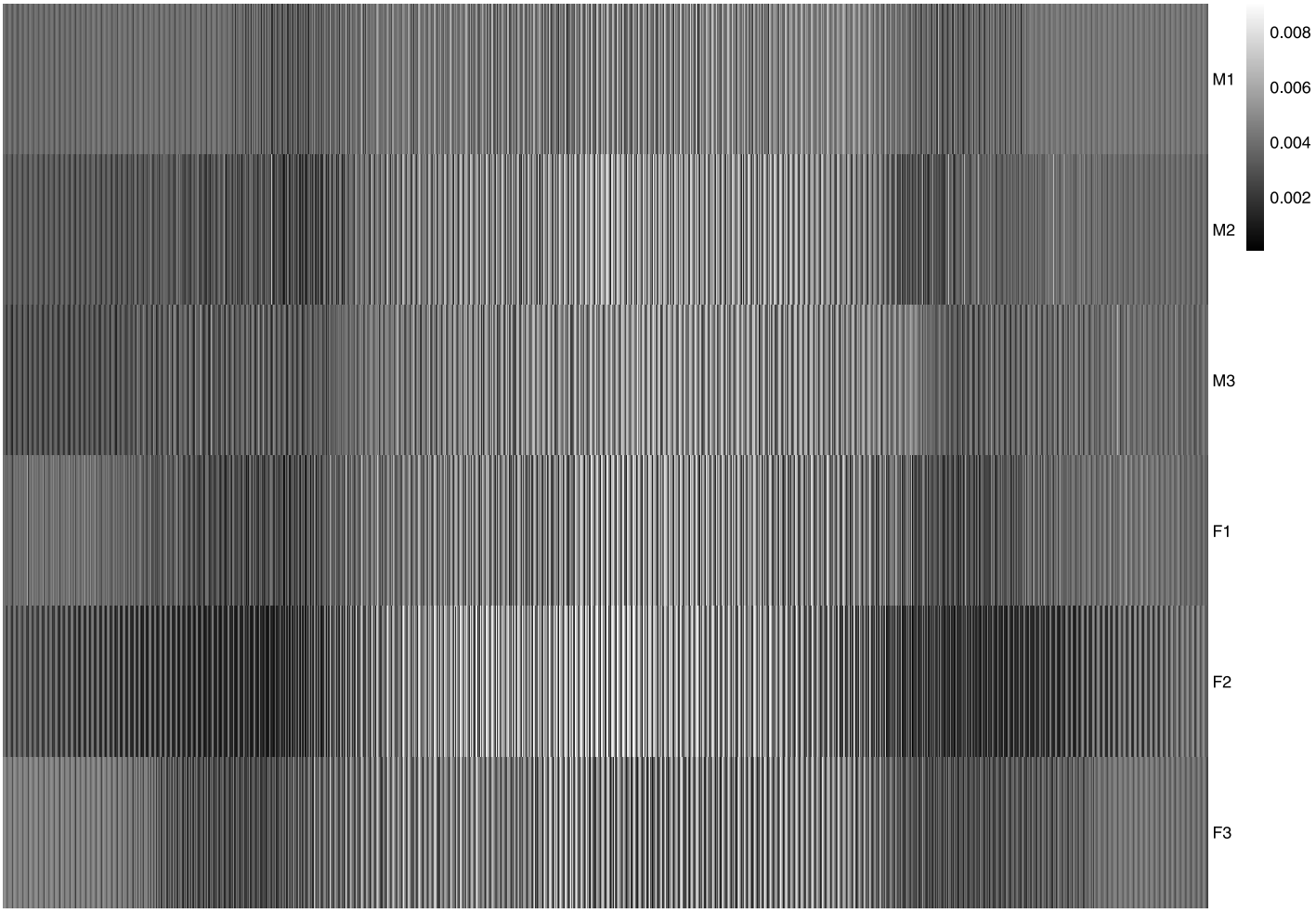

Fig 6. Face data. The data matrix of the face dataset: 6 faces by 55,200 voxels. The female faces are denoted F1, F2, and F3, the male faces M1, M2, and M3.

https://doi.org/10.1371/journal.pone.0211463.g006

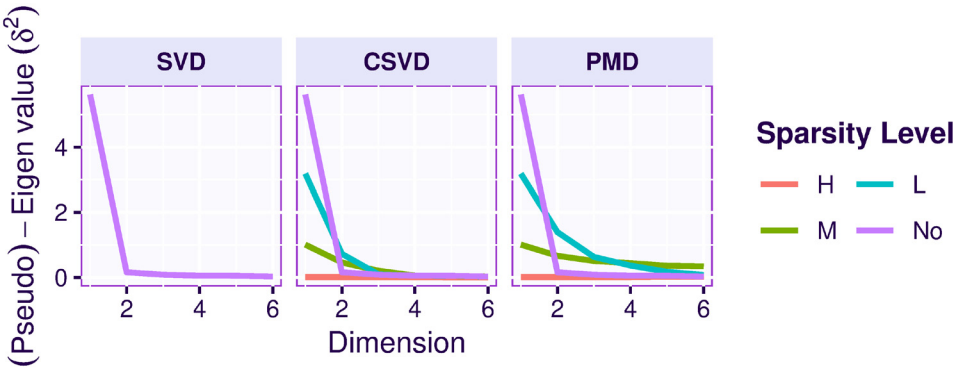

Fig 7. Face data. Eigenvalues and pseudo-eigenvalues per dimension. Left column: eigenvalues obtained from regular SVD; middle column: pseudo-eigenvalues (i.e., variance of the factor scores) for the CSVD; right column: pseudo-eigenvalues (i.e., variance of the factor scores) for the PMD. For PMD and the CSVD, each line represents a level of sparsity: none (No), low (L), medium (M), and high (H).

https://doi.org/10.1371/journal.pone.0211463.g007

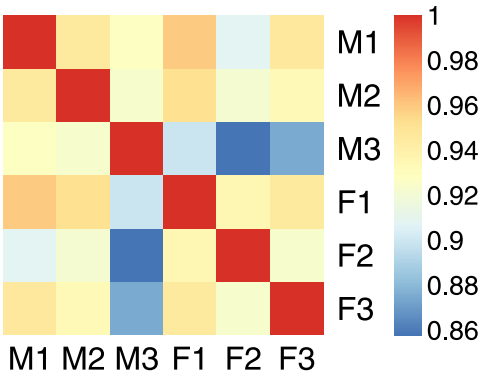

Fig 8. Face data. Cosine matrix for the 6 faces of the face dataset. The female faces are denoted F1, F2, and F3, the male faces M1, M2, and M3.

https://doi.org/10.1371/journal.pone.0211463.g008

interpretation for the plain SVD is confirmed by Fig 13 where all faces load almost identically on Dimension 1, and where Dimension 2 separates men from women.

Overall the CSVD and PMD behave similarly and both show (compared to the plain SVD), that introducing sparsity can make the results easier to interpret because groups of individuals (men or women) can be identified and linked to small subset of variables (i.e., here pixels). However CSVD and PMD differ in the number of components suggested by their scree plot as indicated by Fig 7. The differences between the loadings estimated by both methods are also seen in Fig 14, which depicts the cross-product between the 6 right singular vectors for three different sparsity levels and for both methods. This figure shows that the risk of re-injecting variability, that was already described in previous components, increases with the sparsity parameter and the number of required components.

4.3 Psychometric example: The mental imagery questionnaire

These simulated data were created to match the loading structure of an original dataset obtained from 2,100 participants who—as part of a larger project on memory—answered an online version of the "object-spatial imagery questionnaire" (OSIQ, [29])—a psychometric instrument measuring mental imagery for objects and places. Using a 5-point rating scale, participants rated their agreement for 30 items (e.g., "I am a good Tetris player") that should span a 2-dimensional space corresponding to the hypothesized spatial and object imagery psychometric factors.

The simulated data were obtained from an original data set by first performing a (centered and un-scaled) PCA on the original dataset and keeping only the loadings and the eigenvalues.

Table 4. Face example. Left singular vectors (i.e., face loadings) and associated eigenvalues.

Dimension	Faces						Eigenvalue	Percentage
	1	2	3	4	5	6	λ	τ
1	−.41	−.41	−.40	−.41	−.40	−.41	5.616	93.61
2	.14	.09	.76	−.16	−.53	−.30	0.160	2.66
3	−.40	.13	.29	−.14	.67	−.52	0.086	1.43
4	.08	−.68	.34	−.41	.27	.42	0.055	0.91
5	.45	−.54	−.04	.50	.11	−.50	0.052	0.87
6	−.66	−.25	.25	.60	−.16	.22	0.031	0.52

https://doi.org/10.1371/journal.pone.0211463.t004

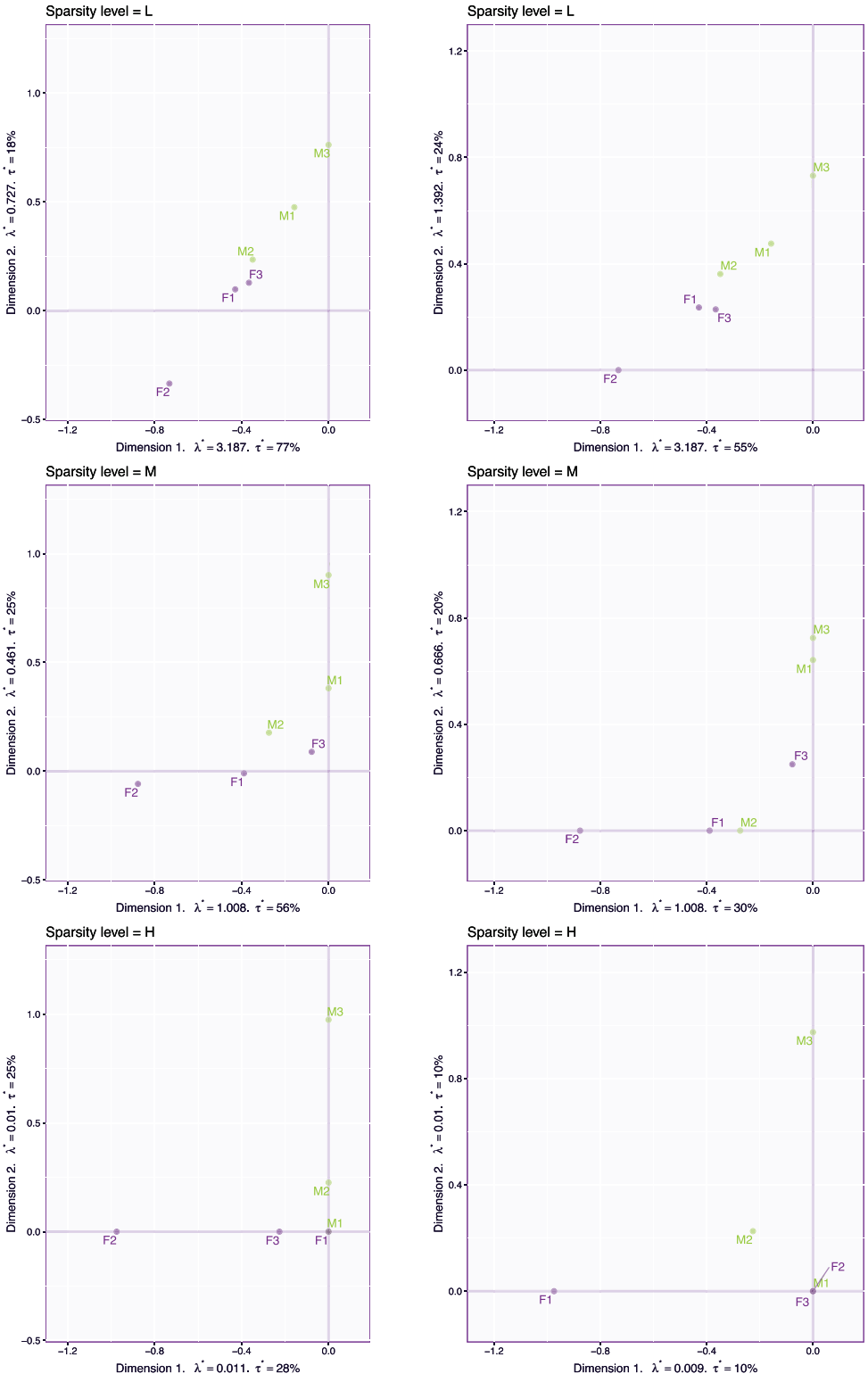

Fig 9. Face data. First two sparse left singular vectors (**P**) for the CSVD (left) and PMD (right) for three levels of sparsity: low (L), medium (M), and high (H). The results for the CSVD are reported on the left, and the results for PMD are reported on the right.

https://doi.org/10.1371/journal.pone.0211463.g009

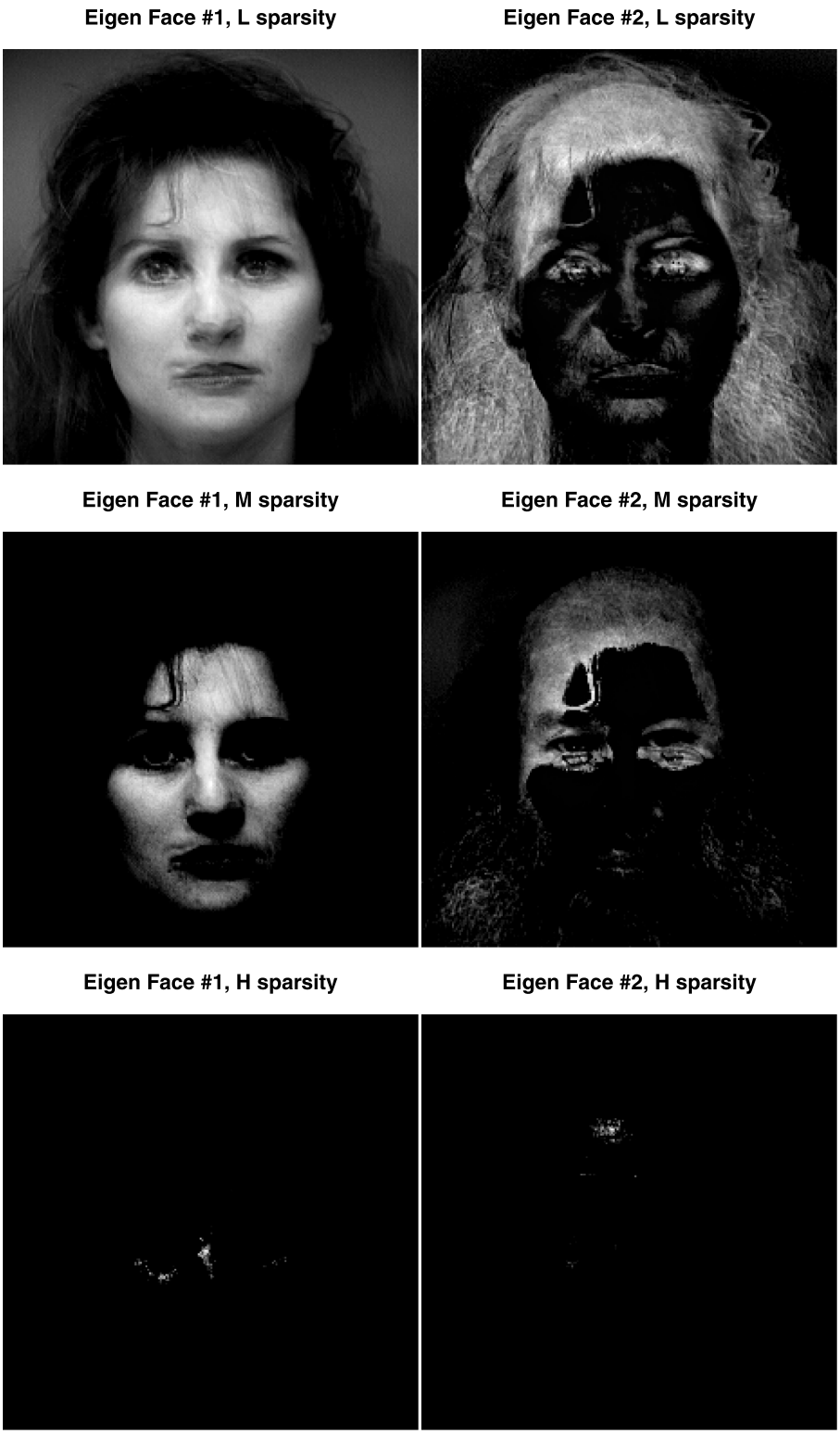

Fig 10. Face data. The pseudo-eigenfaces for Dimension 1 on the left column and Dimension 2 on the right column. For this graph, only the CSVD was applied, with three different levels of sparsity: low on the top row (L), medium on the middle row (M), and high on the bottom row (H).

https://doi.org/10.1371/journal.pone.0211463.g010

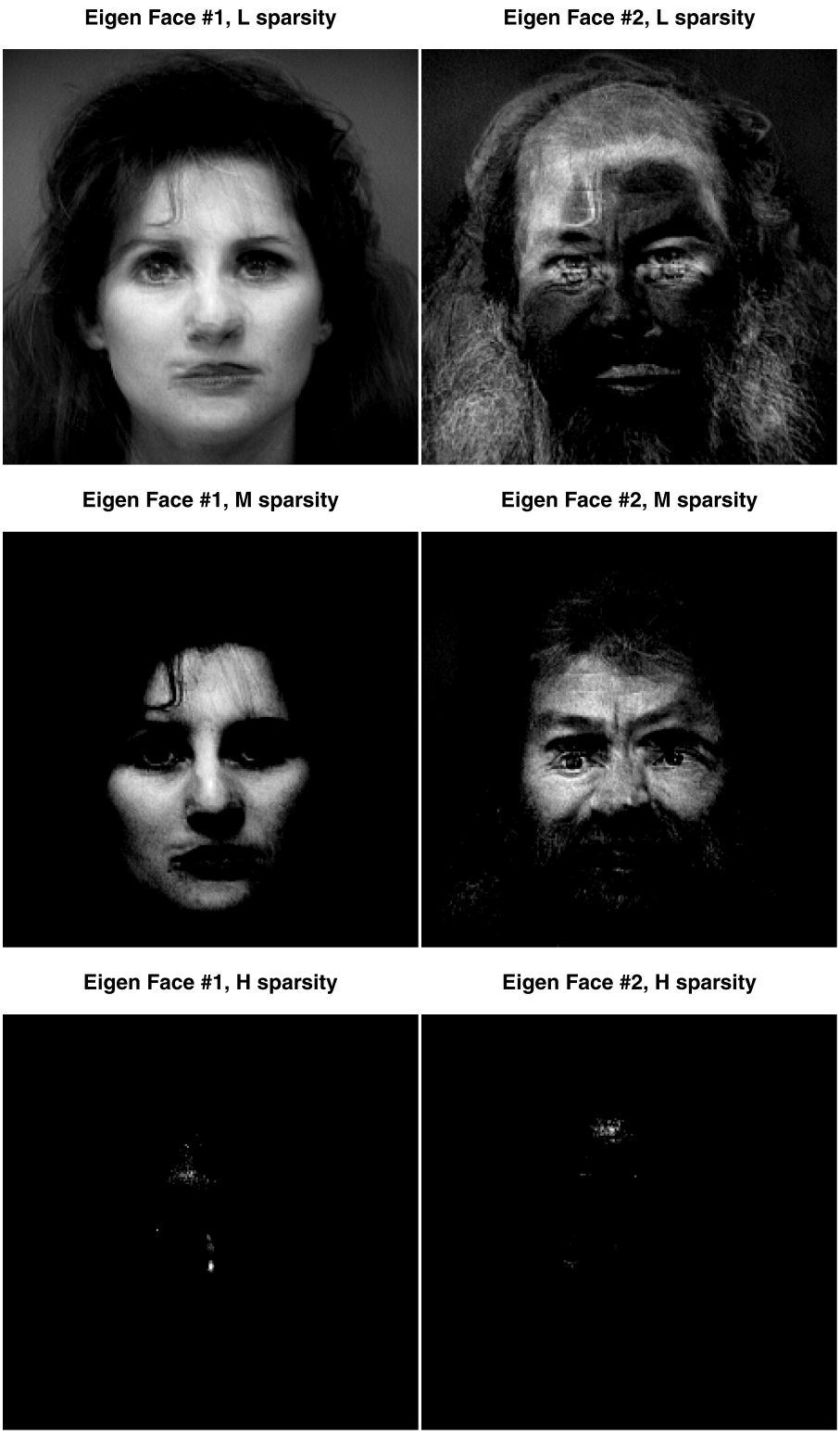

Fig 11. Face data. The pseudo-eigenfaces for Dimension 1 on the left column and Dimension 2 on the right column. For this graph, only PMD was applied, with three different levels of sparsity: low on the top row (L), medium on the middle row (M), and high on the bottom row (H).

https://doi.org/10.1371/journal.pone.0211463.g011

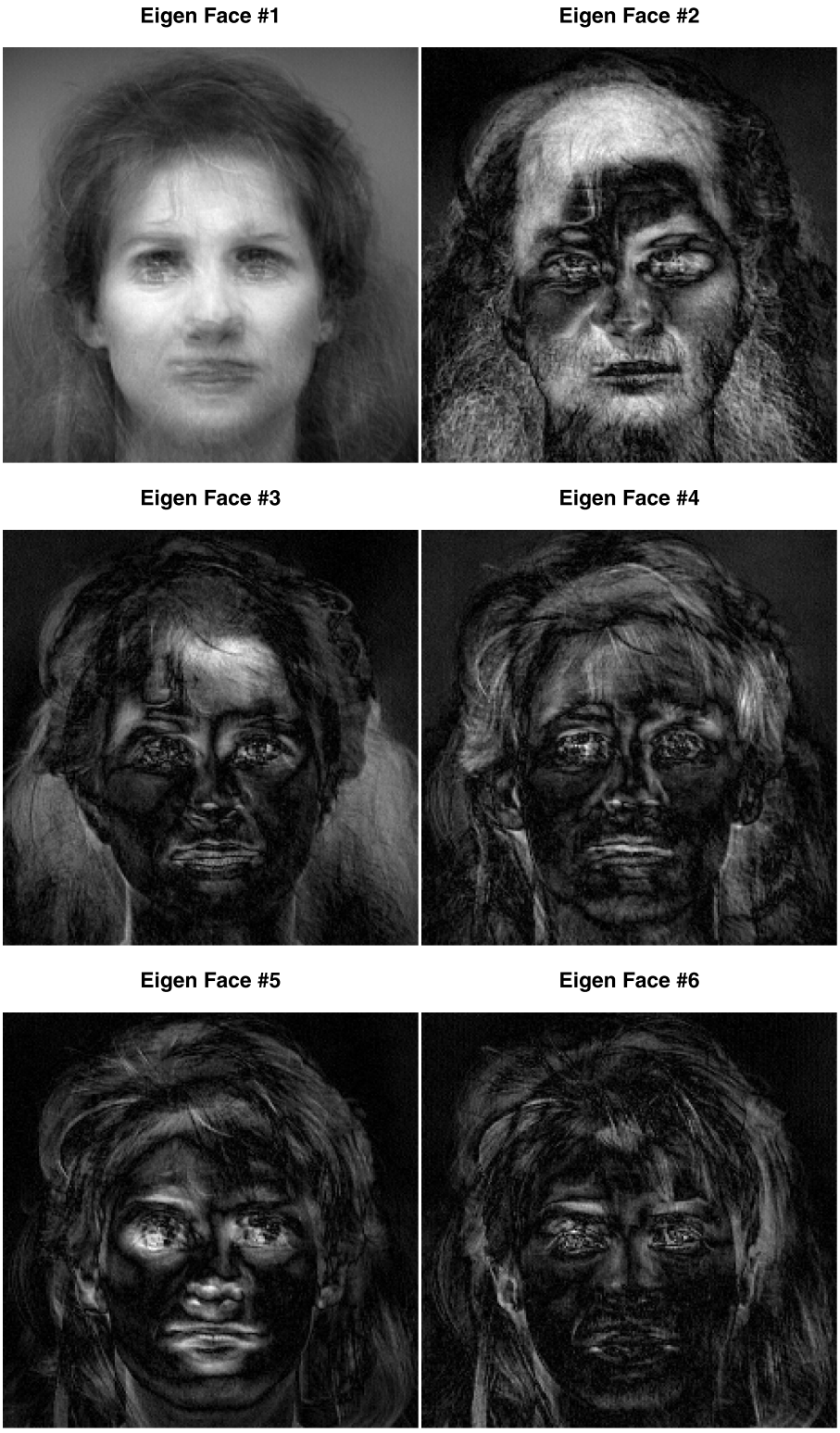

Fig 12. Face data. The six eigenfaces obtained from the plain SVD.

https://doi.org/10.1371/journal.pone.0211463.g012

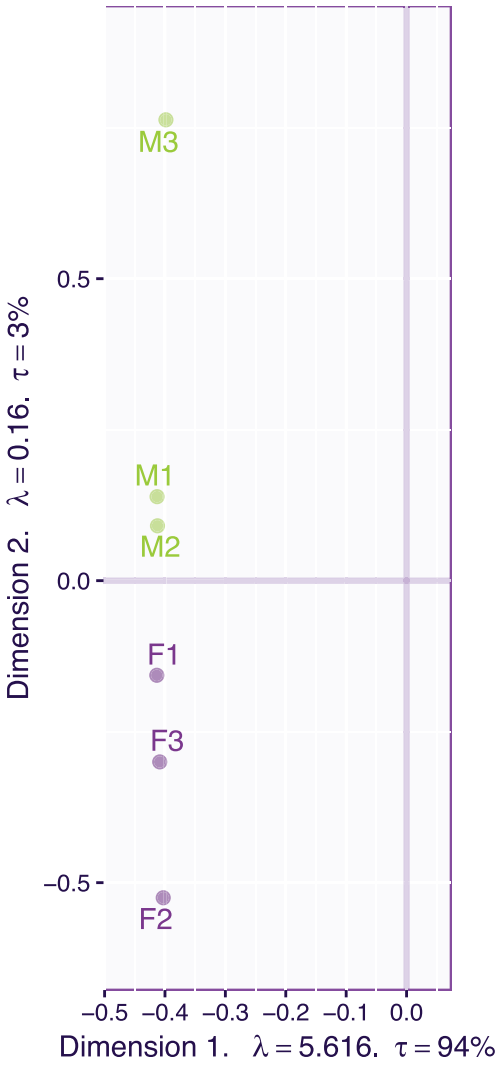

Fig 13. Face data. The two first left singular vectors of the plain SVD of the non-centered data.

https://doi.org/10.1371/journal.pone.0211463.g013

Random pseudo-observations were generated by randomly sampling (with a uniform probability distribution) points in the factor space and then building back the corresponding data matrix from these random factor scores and the loadings. The final simulated data matrix was then obtained by scaling this new data matrix so that it only contains integer values whose distribution match, as best as possible, the original data matrix. This way, the simulated data matrix contains random values whose means, variances, and loadings roughly match the original data matrix. The `R`-code used to create the simulated data can be found from the R-package `data4PCCAR` (available from from the Github repository `HerveAbdi/data4PCCAR`; the simulated data matrix can also be found in the same `R`-package.

The 2,100 (participants) by 30 (items) data matrix was pre-processed by centering and normalizing each variable and was then analyzed by PCA (i.e., an SVD of the pre-processed matrix). Fig 15 plots the loadings for the 30 items for the first two components of the PCA. In this figure, each item is labeled by its number in the questionnaire (see [29] for details and list of questions), and its a priori category (i.e., "object" vs "spatial") is indicated with the first letter

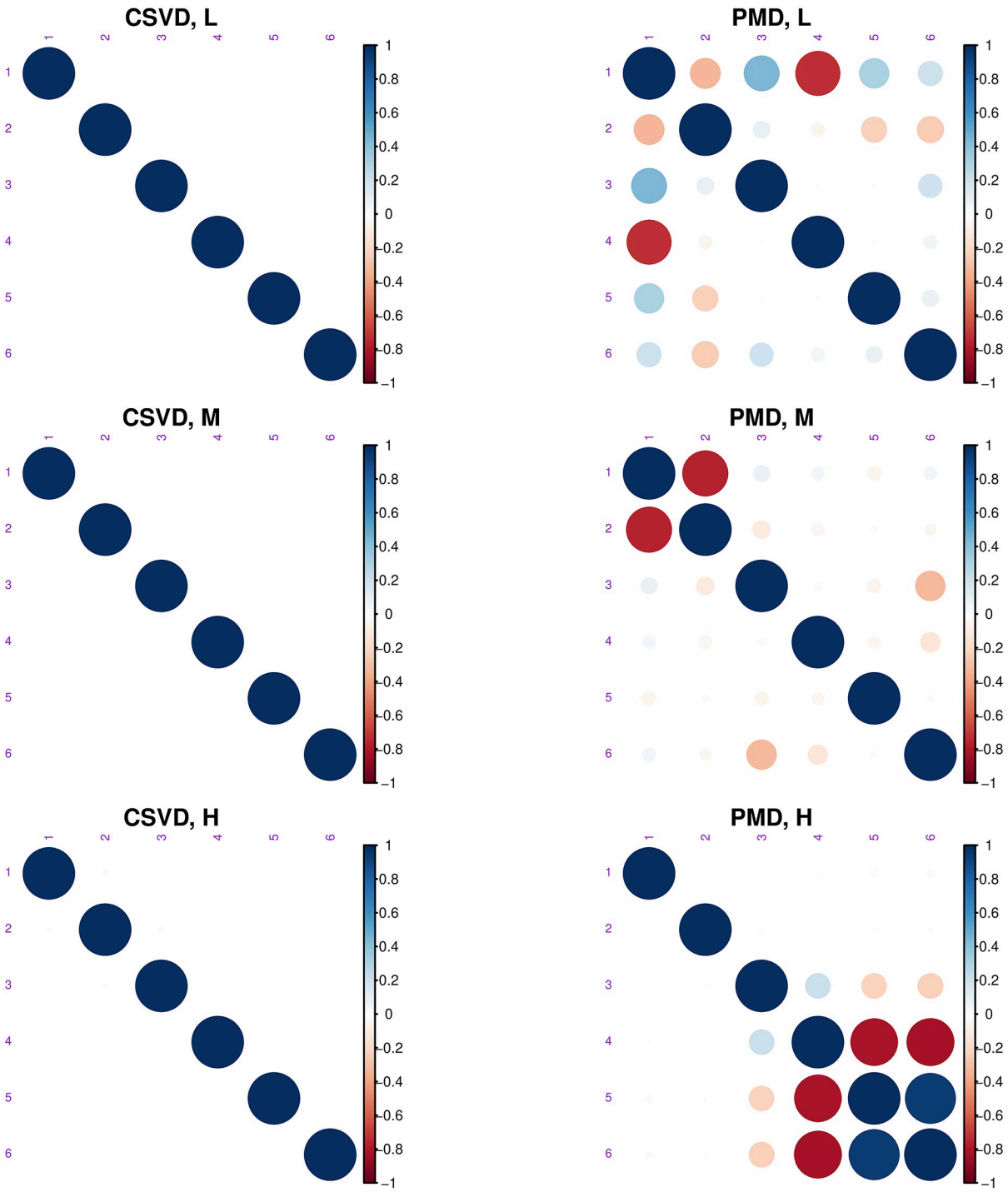

Fig 14. Face data. Cross-product matrix of the 6 right pseudo-singular vectors for three levels of sparsity: low (L), medium (M), and high (H). The results for the CSVD are reported on the left, and the results for PMD are reported on the right.

https://doi.org/10.1371/journal.pone.0211463.g014

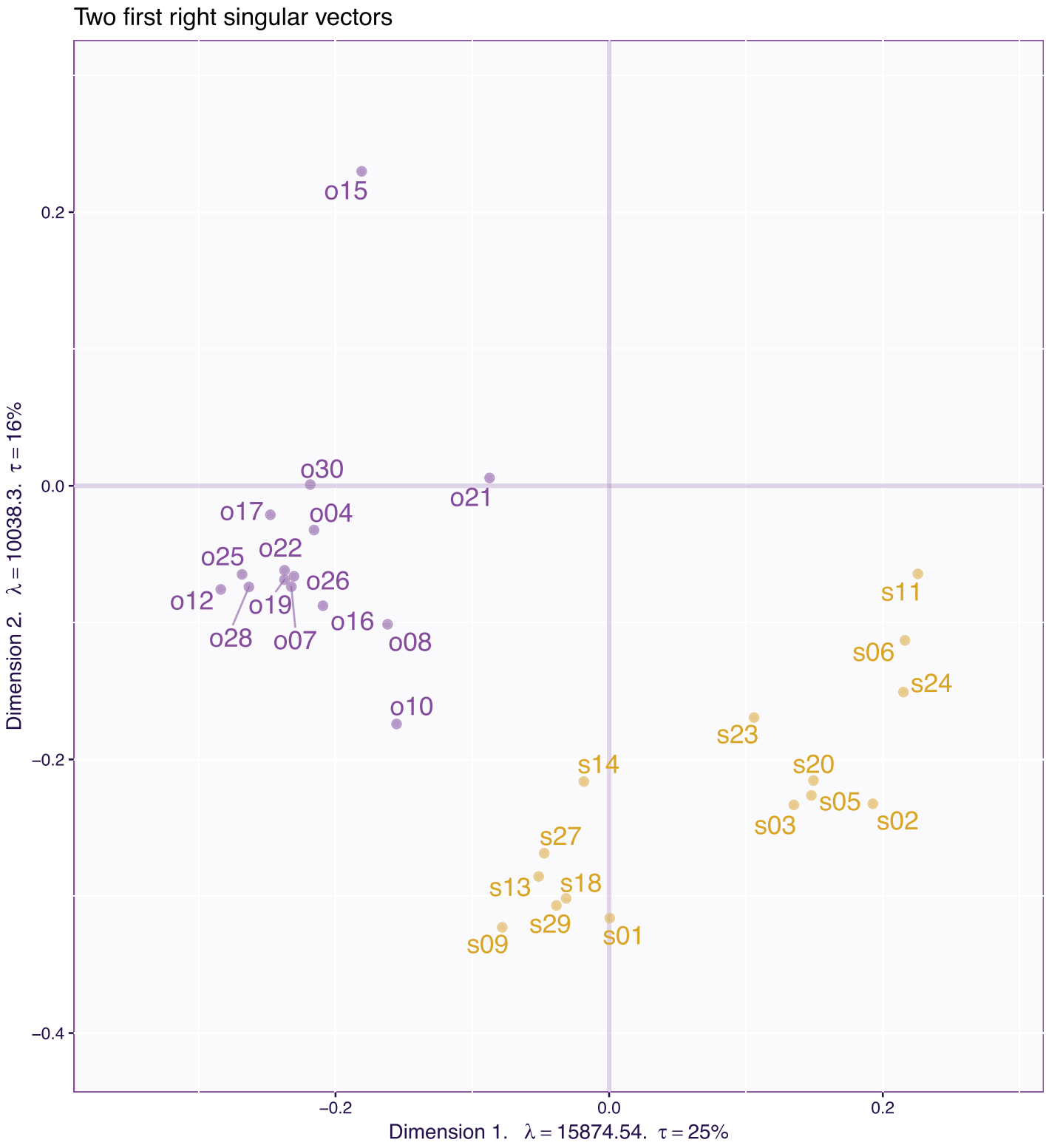

Fig 15. OSIQ data. Loadings of the first two principal components.

https://doi.org/10.1371/journal.pone.0211463.g015

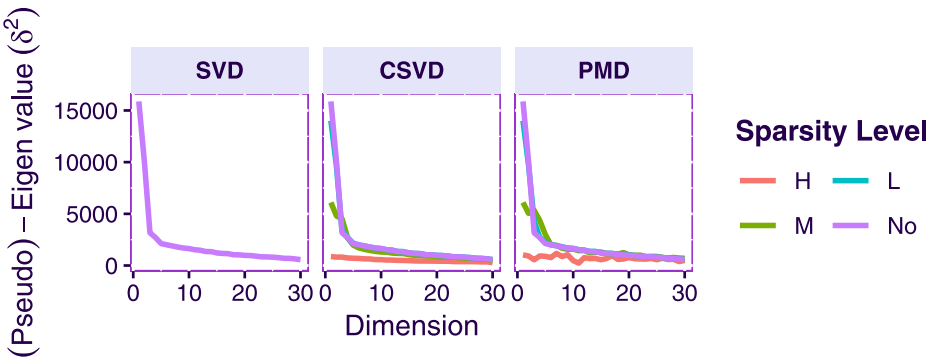

Fig 16. OSIQ data. Scree plots for SVD, the CSVD and PMD for different values of the sparsity parameter.

https://doi.org/10.1371/journal.pone.0211463.g016

(o vs s) and color (blue for "object" vs gold for "spatial"). The scree plot (see Fig 16 left) and the plot of the loadings for the first two dimensions (Fig 15) supports a two factor model (with the plane created by Dimensions 1 and 2 explaining 44% of the total variance). The pattern of the loadings, however, reveals that some items load, as predicted, on only one factor (most object items and some spatial items) but that roughly half of spatial items (i.e., s02, s03, s03, s05, s06, s11, s20, s23, s24) and, at least, one object item (i.e., o15) load on both Dimensions 1 and 2. These items are ambiguous because they can reflect either only one of the hypothesized factors or a combination of both factors. To simplify the interpretation, a standard psychometric approach would keep only the unambiguous items, re-run the analysis with these items, and "prettify" the solution with an orthogonal rotation such as Varimax [33].

To evaluate the effects of sparsification, we used three levels of sparsity (in addition to the "no sparsity" condition corresponding to the plain SVD): Low (L), Medium (M), and High (H). As expected, and illustrated by the scree plots (see Fig 16), sparsification reduced the amount of variance (i.e., the pseudo-eigenvalues) explained by the sparsified components.

Fig 17 plots the item loadings for Dimensions 1 and 2 for both the CSVD (left column) and PMD (right column) as a function of the levels of sparsity (L/M/H). For the low and intermediate levels of sparsity. For the first two levels of sparsity (L and M) the CSVD and PMD give similar results, possibly because the factor structure of the items on the first dimension is strong enough to be recovered without the orthogonality constraints. For the highest level of sparsity, the CSVD and PMD single out the same item (o28) on the first dimension (an unsurprising result because the maximized criteria are equivalent) but single out different items (s29 vs s18) on the second dimension.

Fig 18 plots the correlations between the loadings estimated with the CSVD or PMD for all 30 dimensions and shows again that the components extracted by PMD are correlated with other components.

Visual inspection of the plain PCA analysis suggests that the items are roughly clustered into three groups (pure object, pure spatial, and mixed spatial/objects). To better characterize these three groups of items, we ran an additional analysis in which we set the sparsity parameters to values (specifically $c_1 \approx 0.55\sqrt{I = 2, 100} \approx 25.31$ and $c_2 \approx 0.47\sqrt{J = 30} \approx 2.56$ for, respectively the left and right singular vectors) that would generate three pure dimensions for the item loadings (see Fig 19). With this analysis, the first two dimensions isolate the pure items and the third dimension extracts the mixed items. To confirm this interpretation, we ran a plain PCA on the pure items (see Fig 20 left) followed by a Varimax rotation for two dimensions (see Fig 20 right). The Varimax rotated space recovered a solution equivalent to the

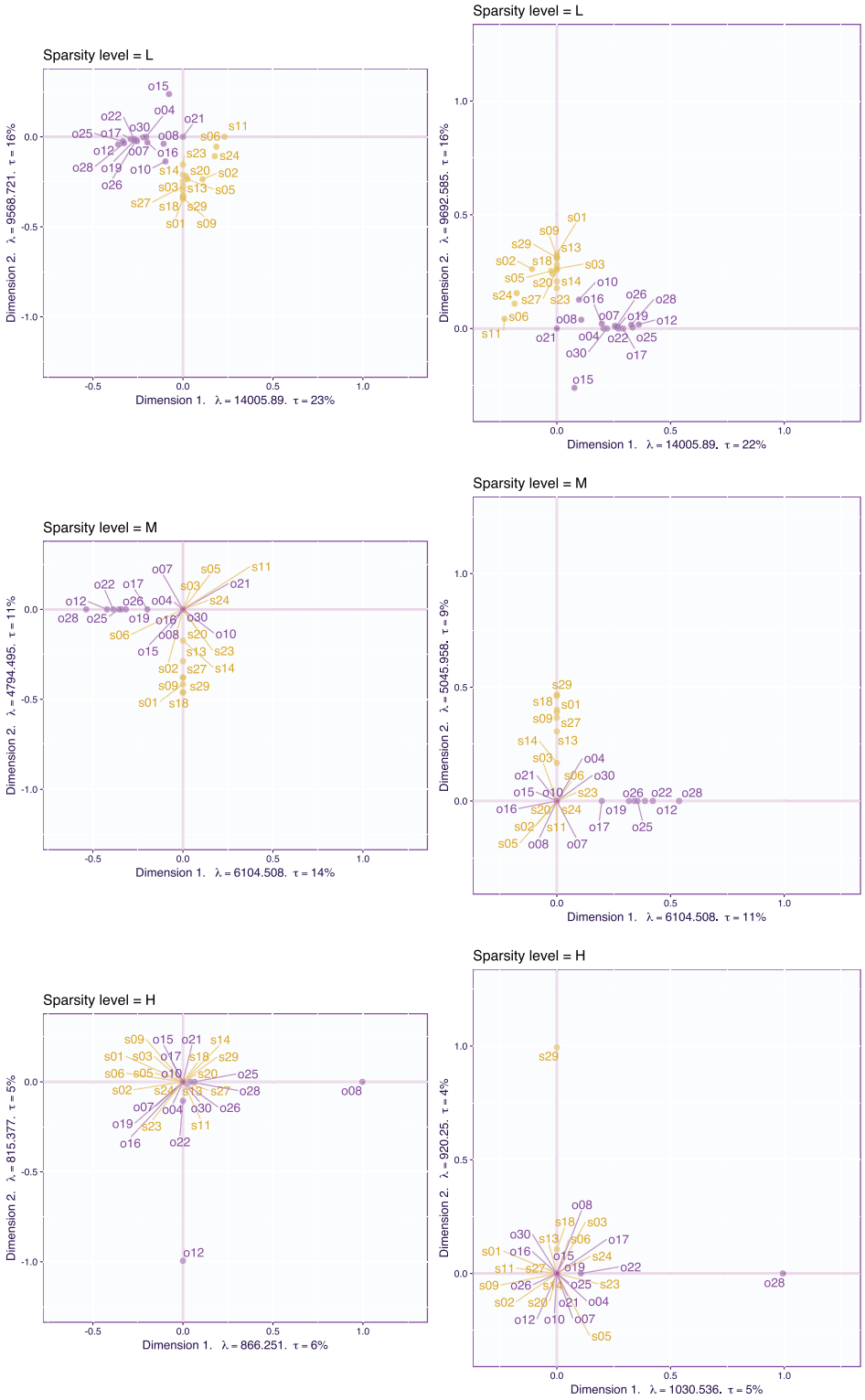

Fig 17. OSIQ data. Loadings of Dimensions 1 and 2 with an increasing degree of sparsity for both the CSVD (left column) and PMD (right column).

https://doi.org/10.1371/journal.pone.0211463.g017

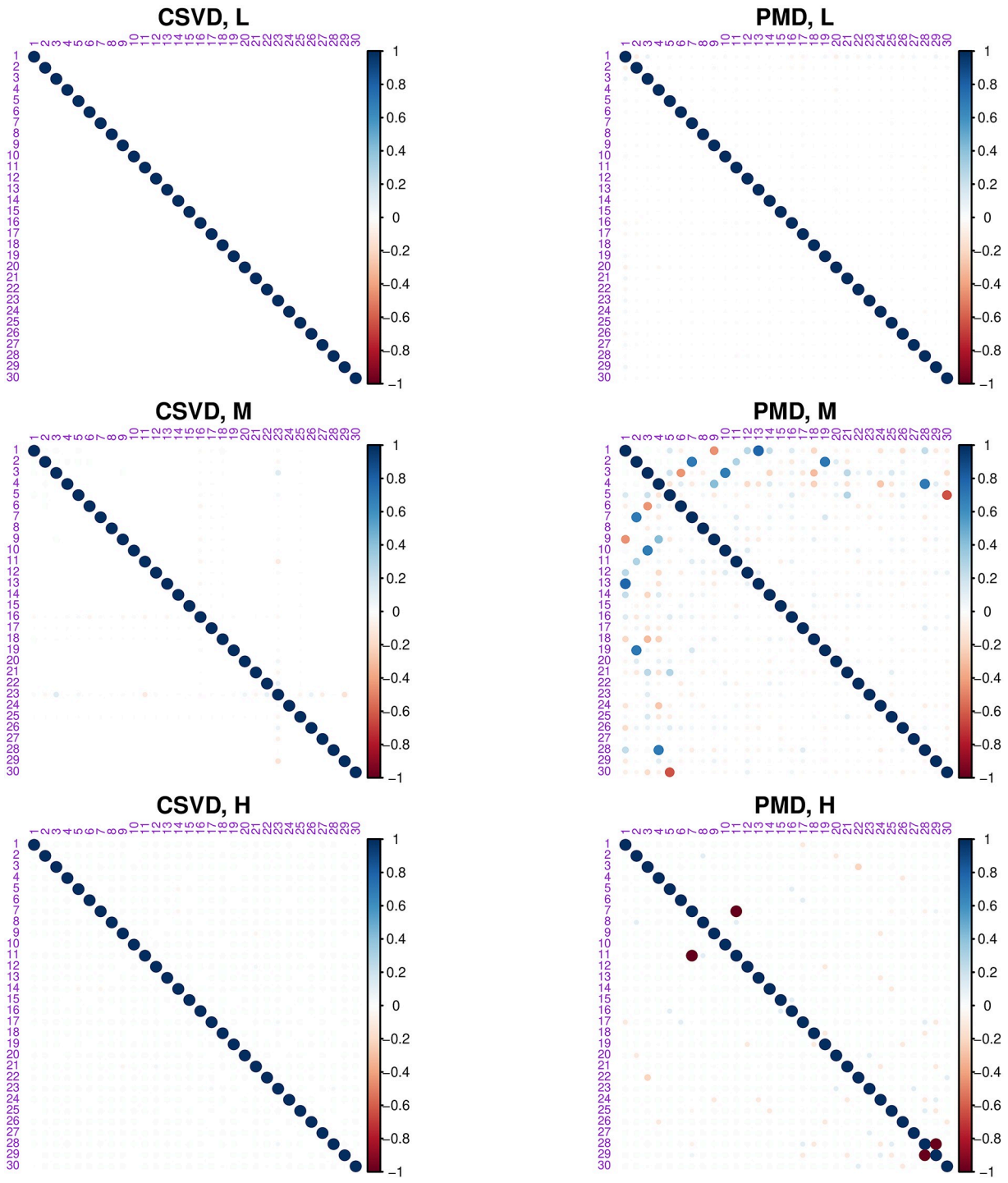

Fig 18. OSIQ data. Plot of the cross-product between the loadings obtained for up to 30 dimensions. Left: CSVD. Right: PMD. Top to bottom: the three different levels of sparsity.

https://doi.org/10.1371/journal.pone.0211463.g018

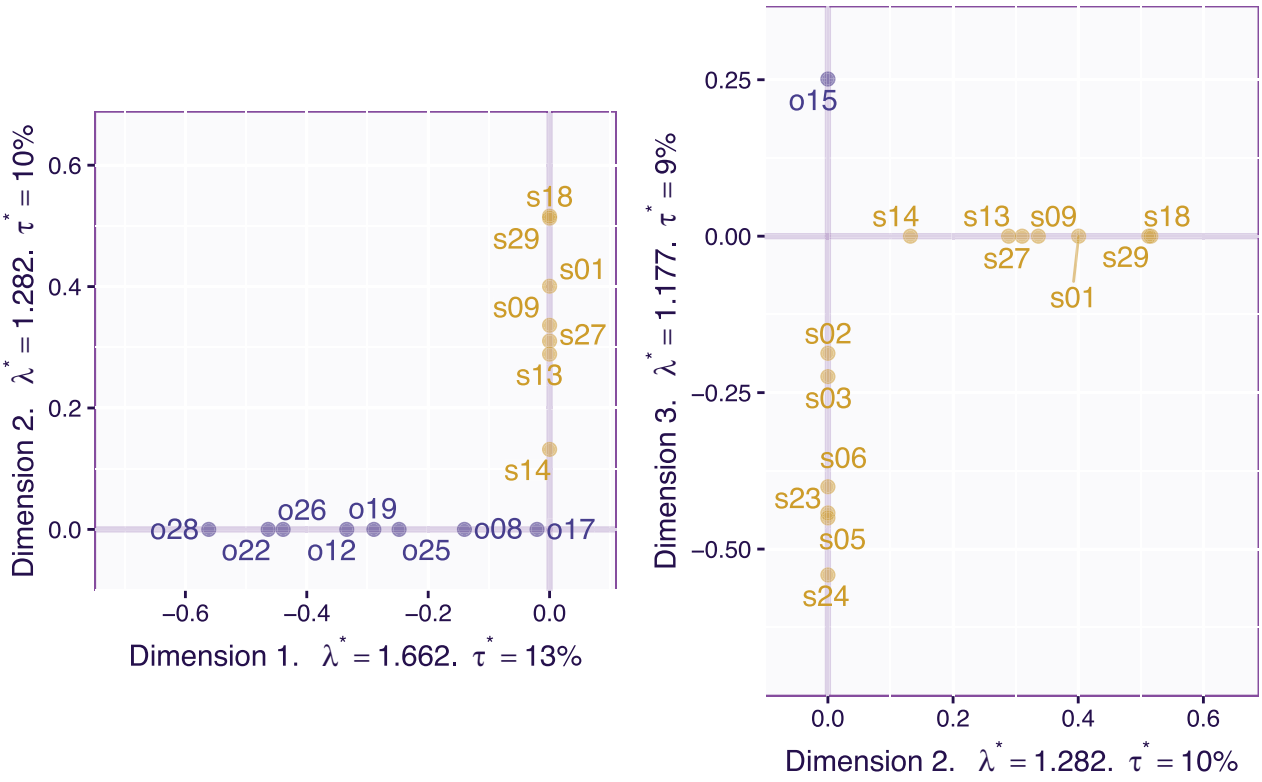

Fig 19. OSIQ data. Loadings for Dimensions 1, 2, and 3 with sparsity parameters set to $c_1 \approx 15.11$ and $c_2 \approx 2.50$. The sparsity parameters were empirically determined visually to create "pure" dimensions.

https://doi.org/10.1371/journal.pone.0211463.g019

CSVD with, however, the caveat, that Varimax required the a priori knowledge of the dimensionality of the space whereas the CSVD did not require this information.

Discussion

The constrained singular value decomposition is a computationally efficient new method that sparsifies the SVD while preserving the orthogonality of the singular vectors. To do so, the CSVD expresses each constraint as a projection onto a convex set and integrates multiple constraints as the projection onto the intersection of the convex sets expressing the constraints (POCS). As shown in Appendix D, the CSVD algorithm is guaranteed to converge to a stable point because it is a member of the more general class of the *block successive upper-bound minimization (BSUM) algorithms*. The CSVD can easily be extended to incorporate additional constraints (e.g., group LASSO, metrics constraints of the rows or columns, spatial constraints) as long as these constraints can be expressed as projections onto convex sets.

To evaluate the relevance of the orthogonality constraints, we compared, on three examples, the plain SVD, the penalized matrix decomposition [11], and the CSVD. We found that, as could be expected, without the orthogonality constraint, higher singular vectors shared information with the earlier singular vectors—a problem likely to hinder the interpretation of these later components. The example using face images shows that the CSVD could extract, from the images, characteristic features defining clusters of observations (e.g., men vs women). The psychometric example illustrates how the CSVD can be used in lieu of rotation (e.g., Varimax)

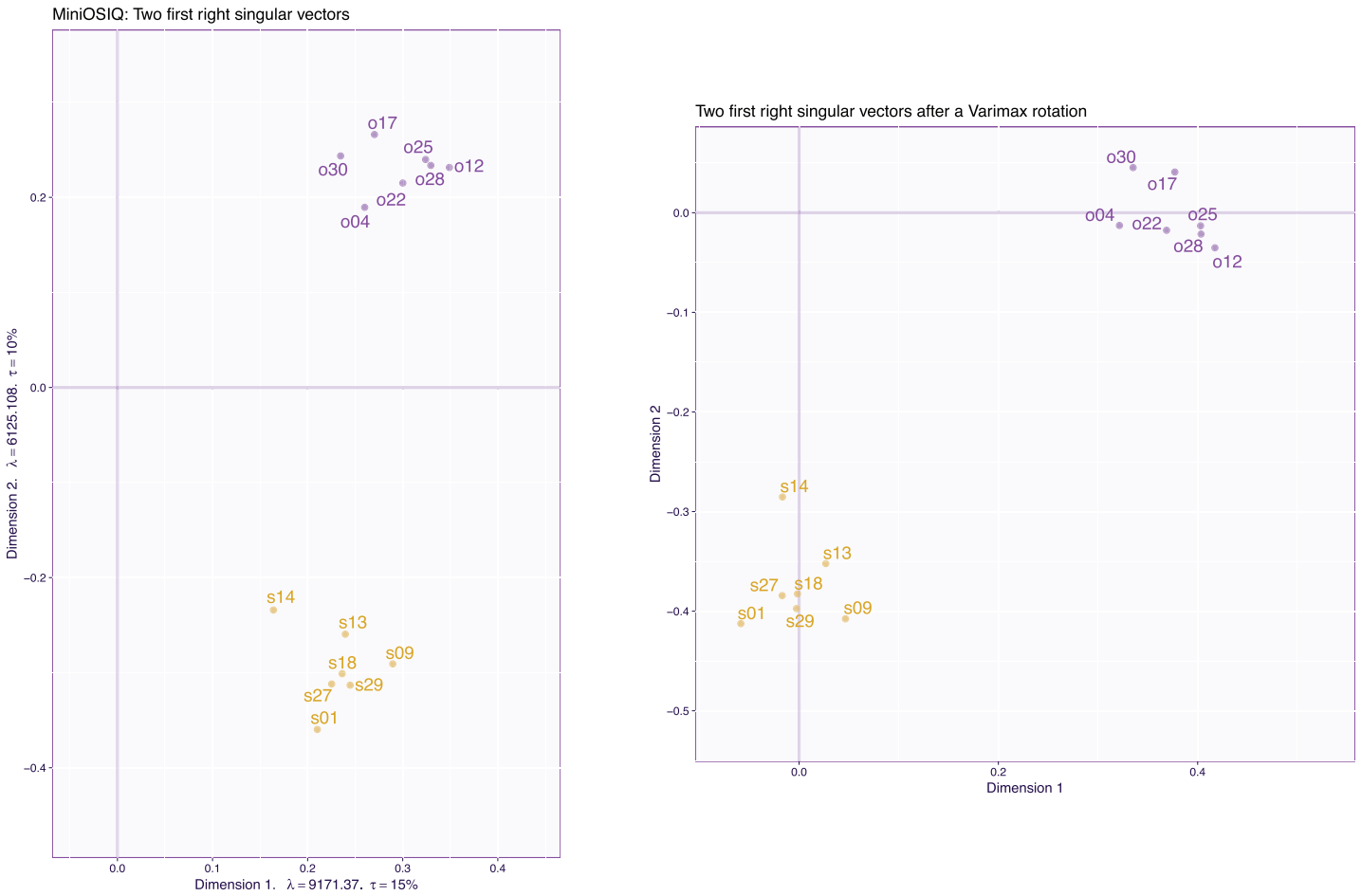

Fig 20. OSIQ data. PCA with the reduced set of 14 items from the OSIQ. Left: Dimensions 1 and 2 for plain PCA; Right: Dimensions 1 and 2 after a two-dimensional Varimax rotation.

https://doi.org/10.1371/journal.pone.0211463.g020

to identify psychometrically "pure" components without having to choose a priori the dimensionality of the space.

Of course, some questions remain open. For example, the choice of the sparsification constant is left to the user but this choice could be helped with some cross-validation schemes such as the one suggested by Witten et al. (see Algorithm 5 in [11]). In practice, this choice is likely to involve some trade-off between the interpretability of the pseudo singular vectors and the number of non-zeros loadings for a few of the first singular vectors. Along the same lines, and just like for the plain SVD, the number of (sparse) singular vectors to examine remains, in part, a subjective decision. Finally, the problem of the reliability of the sparsification, although a topic of great interest for sparse methods of prediction [34], remains open for sparse SVD or PCA and should be a topic for future research. Future directions should also include the integration of the CSVD into other methods that are traditionally based on the regular SVD—such as canonical correlation, or partial least squares correlation—or on the generalized SVD—such as correspondence analysis (see, e.g., [15] for previous relevant work along these lines).

The `R` package `csvd` implementing the constrained singular value decomposition is available for download from https://github.com/vguillemot/csvd.

A The deflation operation generates orthogonal singular vectors

In this section, we show that repeatedly using the deflation operation will generate a set of left (respectively right) orthogonal singular vectors ordered by their singular value.

Theorem 1 (Deflation). *The first singular triplet of the* $(k+1)$*th deflated matrix is the* $(k+1)$*th singular triplet of the original matrix.*

Proof. Assume that the $k \geq 1$, singular values, left and right singular vectors of a given matrix $\mathbf{X}$ have been estimated and stored in matrices $\mathbf{\Delta}_k$, $\mathbf{P}_k$ and $\mathbf{Q}_k$.

Let

$$\mathbf{X}^{(k+1)} = \mathbf{X} - \mathbf{P}_k \mathbf{\Delta}_k \mathbf{Q}_k^\top, \tag{21}$$

be the $(k+1)$th deflated matrix with:

$$\mathbf{P}_k^\top \mathbf{P}_k = \mathbf{I} \text{ and } \mathbf{Q}_k^\top \mathbf{Q}_k = \mathbf{I}. \tag{22}$$

Additionally, let δ, $\mathbf{p}$ and $\mathbf{q}$ be (respectively) the first singular value, left and right singular vectors of $\mathbf{X}^{(k+1)}$.

To prove Theorem 1, we show that δ, $\mathbf{p}$ and $\mathbf{q}$ are the $(k+1)$th singular triplet of $\mathbf{X}$.

First, because δ, $\mathbf{p}$ and $\mathbf{q}$ are a singular triplet of $\mathbf{X}^{(k+1)}$, we have:

$$\delta \mathbf{q} = \mathbf{X}^{(k+1)\top} \mathbf{p} \tag{23}$$

and

$$\delta \mathbf{p} = \mathbf{X}^{(k+1)} \mathbf{q}. \tag{24}$$

Second, to prove that $\mathbf{q}$ is orthogonal to each column of $\mathbf{Q}_k$, we consider the quantity $\mathbf{Q}_k^\top \mathbf{q}$. By multiplying both sides of Eq 23 by $\mathbf{Q}_k^\top$, developing, and simplifying we obtain

$$\begin{aligned} \delta \mathbf{Q}_k^\top \mathbf{q} &= \mathbf{Q}_k^\top \mathbf{X}^{(k+1)\top} \mathbf{p} \\ &= \mathbf{Q}_k^\top (\mathbf{X}^\top - \mathbf{Q}_k \mathbf{\Delta}_k \mathbf{P}_k^\top) \mathbf{p} \qquad \text{(Cf. Equation 21)} \\ &= \underbrace{\mathbf{Q}_k^\top \mathbf{X}^\top}_{\mathbf{\Delta}_k \mathbf{P}_k^\top} \mathbf{p} - \underbrace{\mathbf{Q}_k^\top \mathbf{Q}_k}_{\mathbf{I}} \mathbf{\Delta}_k \mathbf{P}_k^\top \mathbf{p} \\ &= \mathbf{\Delta}_k \mathbf{P}_k^\top \mathbf{p} - \mathbf{\Delta}_k \mathbf{P}_k^\top \mathbf{p} \\ &= \mathbf{0}. \end{aligned} \tag{25}$$

Therefore, when δ is not null, $\mathbf{q}$ is orthogonal to each column of $\mathbf{Q}_k$. A similar proof shows that $\mathbf{p}$ is orthogonal to each column of $\mathbf{P}_k$.

Third, because $\mathbf{p}$ (respectively, $\mathbf{q}$) is orthogonal to each column of $\mathbf{P}_k$ (respectively $\mathbf{Q}_k$), we have

$$\mathbf{X}^{(k+1)\top} \mathbf{p} = \delta \mathbf{q}, \text{ and } \mathbf{X}^{(k+1)} \mathbf{q} = \delta \mathbf{p}, \tag{26}$$

which, combined with Eqs (23) and (24), implies that

$$\mathbf{X}^\top \mathbf{p} = \delta \mathbf{q}, \text{ and } \mathbf{X} \mathbf{q} = \delta \mathbf{p}, \tag{27}$$

which, in turn, shows that δ, $\mathbf{p}$, and $\mathbf{q}$ are a singular triplet of $\mathbf{X}$. This proof also shows (*mutatis mutandis*) that any singular triplet of $\mathbf{X}^{(k+1)}$ is a singular triplet of $\mathbf{X}$.

Finally, from the definition of $\mathbf{X}^{(k+1)}$, and because $\mathbf{X}^{(k+1)}$ is orthogonal to $\mathbf{P}_k \mathbf{\Delta}_k \mathbf{Q}_k$, the rank of $\mathbf{X}$ is equal to the rank of $\mathbf{X}^{(k+1)}$ plus the rank of $\mathbf{P}_k \mathbf{\Delta}_k \mathbf{Q}_k^\top$, which, in run, implies that all the

singular values of $\mathbf{X}^{(k+1)}$ are the remaining singular values of $\mathbf{X}$ and that the first singular value of $\mathbf{X}^{(k+1)}$ is the $(k+1)$th singular value of $\mathbf{X}$.

B Only some values of the constraints lead to solutions

As stated by Witten et al. ([11], page 519 *ff.*) the constraint parameters c_1 and c_2 lead to solutions only when they are in the range:

$$1 \leq c_1 \leq \sqrt{I} \text{ and } 1 \leq c_2 \leq \sqrt{J}. \tag{28}$$

Fig 1 in [11] describes the geometric intuition behind this range in $\mathbb{R}^2$.

In this appendix, we provide a proof of Statement 28. First, we prove Lemma 1 (that directly implies Statement 28).

Lemma 1. Let $\mathbf{x} \in \mathbb{R}^N$, *then*

$$\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{N}\|\mathbf{x}\|_2 \tag{29}$$

Proof. Assume that $\mathbf{x}$ belongs to $\mathbb{R}^N$ and is different from $\mathbf{0}$. The left side of the inequality is a consequence of Hölder's inequality, which states that if $0 < p < +\infty$, and q is a positive real number such that $\frac{1}{p} + \frac{1}{q} = 1$, then

$$\|\mathbf{x}\|_2^2 \leq \|\mathbf{x}\|_p \|\mathbf{x}\|_q. \tag{30}$$

With $p = 1$, this version of Hölder's inequality becomes

$$\|\mathbf{x}\|_2^2 \leq \|\mathbf{x}\|_1 \|\mathbf{x}\|_\infty. \tag{31}$$

Since $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2$, we have

$$\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1. \tag{32}$$

The right hand side of Eq 29 can be seen as a consequence of Cauchy-Schwarz inequality, which, in our case, would be formulated as follows:

$$\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^N, |\langle \mathbf{a}, \mathbf{b} \rangle| \leq \|\mathbf{a}\|_2 \|\mathbf{b}\|_2. \tag{33}$$

If we set $\mathbf{a}$ to $[|x_1|, \ldots, |x_n|]^\top$ and $\mathbf{b}$ to $\mathbf{1}$, we obtain

$$\left| \sum_{i=1}^{N} |x_i| \right| \leq \|\mathbf{x}\|_2 \|\mathbf{1}\|_2, \tag{34}$$

which is equivalent to

$$\|\mathbf{x}\|_1 \leq \sqrt{N}\|\mathbf{x}\|_2. \tag{35}$$

Putting together Eqs 32 and 35 gives:

$$\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{N}\|\mathbf{x}\|_2 \ . \tag{36}$$

Lemma 1 implies that the constraints on the L_2 and L_1 norm of the left and right pseudo-singular vectors can be both active at the same time only if the sparsity parameter is chosen such that: (i) the L_1-ball of radius ρ (i.e., $\mathscr{B}_{L_1}(\rho)$) is entirely included in the L_2-ball of radius 1 (i.e., $\mathscr{B}_{L_2}(1)$) when $\rho \leq 1$, and so that fulfilling the sparsity constraint implies that the L_2 constraint is also fulfilled; and (ii) $\mathscr{B}_{L_2}(1)$ is entirely included in $\mathscr{B}_{L_1}(\rho)$ when $\rho \geq \sqrt{N}$, and so

fulfilling the normalization constraint implies that the sparsity constraint is also fulfilled. To fulfill the constraints on both rows and columns of the CSVD gives the following range for the values of c_1 and c_2:

$$1 \leq c_1 \leq \sqrt{I} \quad \text{and} \quad 1 \leq c_2 \leq \sqrt{J}, \tag{37}$$

which proves the assertion.

C A fast and exact algorithm for the projection onto the intersection of an L_1 and L_2 ball

In this section we describe a fast and exact algorithm for the projection onto the intersection of the L_1-ball of radius c (i.e., $\mathscr{B}_{L_1}(c)$) and the L_2-ball of radius 1 (i.e., $\mathscr{B}_{L_2}(1)$). This projection is defined by the following equation:

$$\text{proj}\,(\mathbf{x}, \mathscr{B}_{L_1}(c) \cap \mathscr{B}_{L_2}(1)) = \begin{cases} \arg\min_{\mathbf{y}\in\mathbb{R}^N} \|\mathbf{y}-\mathbf{x}\|_2^2, \\ \text{s.t. } \mathbf{y} \in \mathscr{B}_{L_1}(c) \cap \mathscr{B}_{L_2}(1) \end{cases} \tag{38}$$

where $\mathbf{x} \in \mathbb{R}^N$, N is the number of variables of the dimension of interest (i.e., I or J), and c is the sparsity parameter (i.e., c_1 or c_2) with $1 \leq c \leq \sqrt{N}$.

In [11], the solution of Eq 38 is computed using a binary search algorithm (BiSe). In the main part of our artcie, we propose to use the more general POCS algorithm. BiSe and POCS are iterative algorithms that give an approximate solution to Eq 38. In the case of the projection on the intersection of the L_1 and L_2 balls, the general POCS algorithm can be replaced by a fast and exact algorithm (see [26, 35]), that we call PL1L2 and detail in this appendix.

Projection onto the L_1-ball

The proposed approach implements an efficient algorithm for projecting a vector onto the L_1-ball [35].

Let $\tilde{\mathbf{x}}$ be the vector containing the absolute value of the components of $\mathbf{x}$ with its elements sorted in decreasing order. Additionally, we define the function $\varphi(\lambda) = \|S(\mathbf{x}, \lambda)\|_1$. This function is continuous, piecewise linear and decreasing from $\varphi(0) = \|\tilde{\mathbf{x}}\|_1$ to $\varphi(\tilde{x}_1) = 0$. Therefore, if $\|\mathbf{x}\|_1 \geq c$, since φ is continuous, there is a positive number λ such that $\varphi(\lambda) = c$. From this, we can deduce the algorithm of the projection onto the L_1-ball of radius c that narrows down to 4 steps.

Algorithm 6: Fast projection onto the L_1-ball.

Data: $\mathbf{x}$, c

Result: $\text{proj}_{\mathscr{B}_{L_1}(c)}(\mathbf{x})$

1. Take the absolute value of the components of $\mathbf{x}$ and sort them in decreasing order into a new vector $\tilde{\mathbf{x}}$;
2. Find i such that $\varphi(\tilde{x}_i) \leq c < \varphi(\tilde{x}_{i+1})$;
3. Find δ such that $\varphi(\tilde{x}_i - \delta) = c$. Since $\varphi(\tilde{x}_i - \delta) = \sum_{k=1}^{i} \tilde{x}_k - i(\tilde{x}_i - \delta) = \varphi(\tilde{x}_i) + i\delta$, then $\delta = \frac{c - \varphi(\tilde{x}_i)}{i}$;
4. Compute $S(\mathbf{x}, \lambda)$ with $\lambda = \tilde{x}_i - \delta$;

At the end of the algorithm, we obtain $S(\mathbf{x}, \lambda)$ which is now the projection of $\mathbf{x}$ onto $\mathscr{B}_{L_1}(c)$. A similar algorithm was proposed in [36], [37], and [38].

Projection onto the intersection of the L_1 and L_2-balls

In order to solve the optimization problem from Eq 38, we extend Algorithm 6 to the function

$$\psi(\lambda) = \frac{\|S(\tilde{\mathbf{x}}, \lambda)\|_1}{\|S(\tilde{\mathbf{x}}, \lambda)\|_2} \ . \tag{39}$$

We have the following Lemma:

Lemma 2. *Let* $\mathbf{x}$ *be a vector of* $\mathbb{R}^N$, *composed of* $n \leq N$ *non-zero elements. Then*

$$\|\mathbf{x}\|_1 \leq \sqrt{n}\|\mathbf{x}\|_2 \ . \tag{40}$$

Proof. The proof of this Lemma is very similar to the proof given in Appendix B. Recall that as a consequence of the Cauchy-Schwarz inequality:

$$\forall \ \mathbf{a}, \mathbf{b} \in \mathbb{R}^N, |\langle \mathbf{a}, \mathbf{b} \rangle| \leq \|\mathbf{a}\|_2 \|\mathbf{b}\|_2. \tag{41}$$

With $\mathbf{a} = [|x_1|, \ldots, |x_N|]$ and $\mathbf{b}$ a vector such that

$$b_i = \begin{cases} 1 \text{ if } x_i \neq 0, \\ 0 \text{ if } x_i = 0, \end{cases} \tag{42}$$

the previous inequality becomes

$$\sum_{i=1}^{N} b_i |x_i| \leq \|\mathbf{x}\|_2 \|\mathbf{b}\|_2, \tag{43}$$

which is equivalent to

$$\|\mathbf{x}\|_1 \leq \sqrt{n}\|\mathbf{x}\|_2. \tag{44}$$

Proposition 1. For $\lambda \in [0; \tilde{x}_1[$,

$$\psi(\lambda) = \frac{\|S(\tilde{\mathbf{x}}, \lambda)\|_1}{\|S(\tilde{\mathbf{x}}, \lambda)\|_2} \tag{45}$$

verifies the 3 following properties:

1. ψ *is continuous and decreasing.*
2. *There exist an integer i and a positive real number* δ, *smaller than* $\tilde{x}_i - \tilde{x}_{i+1}$, *such that* $\psi(\tilde{x}_i - \delta) = c$.
3. δ *is the solution of a second degree polynomial equation.*

Proof. (**i**). The numerator and denominator of ψ are continuous because there are compositions of continuous functions. Moreover, for any λ strictly smaller than $\tilde{x}_1$, $\|S(\tilde{\mathbf{x}}, \lambda)\|_2 \neq 0$. Therefore, ψ is continuous because it is the ratio of a continuous function and a non-zero continuous function.

Assuming $\tilde{x}_{N+1} = 0$, for $\lambda \in [0; \tilde{x}_1[$ there exists $k \in 1, \ldots, N$ such that $\tilde{x}_{k+1} \leq \lambda < \tilde{x}_k$. For this specific λ, we have:

$$\|S(\tilde{\mathbf{x}}, \lambda)\|_1 = \left(\sum_{j=1}^{k} \tilde{x}_j \right) - k\lambda \tag{46}$$

and

$$\|S(\tilde{\mathbf{x}},\lambda)\|_2^2 = \sum_{j=1}^{k}(\tilde{x}_j-\lambda)^2 = \left(\sum_{j=1}^{k}\tilde{x}_j^2\right) - 2\lambda\left(\sum_{j=1}^{k}\tilde{x}_j\right) + k\lambda^2 \ . \tag{47}$$

Together, Eqs 46 and 47 imply that the derivative of ψ has the form:

$$\psi'(\lambda) = \frac{1}{\|S(\tilde{\mathbf{x}},\lambda)\|_2^2}\left(\frac{\|S(\tilde{\mathbf{x}},\lambda)\|_1^2}{\|S(\tilde{\mathbf{x}},\lambda)\|_2} - k\|S(\tilde{\mathbf{x}},\lambda)\|_2\right) = \frac{1}{\|S(\tilde{\mathbf{x}},\lambda)\|_2}(\psi(\lambda)^2-k) \ . \tag{48}$$

Moreover, because the number of non-zero elements of vector $S(\tilde{\mathbf{x}},\lambda)$ is equal to k, Lemma 2 implies that $\|S(\tilde{\mathbf{x}},\lambda)\|_1 \leq \sqrt{k}\|S(\tilde{\mathbf{x}},\lambda)\|_2$, and therefore $\psi(\lambda)^2 \leq k$. As a consequence, $\psi'(\lambda) \leq 0$, which, in turn, implies that ψ, being a continuous function with a negative derivative, is a decreasing function.

(**ii**). Let $N_{\max}$ be the number of elements of $\mathbf{x}$ equal to $\tilde{x}_1$ (the maximum of $\tilde{\mathbf{x}}$) and $v \in [\tilde{x}_2;\tilde{x}_1[$. Then

$$\psi(v) = \frac{N_{\max}(\tilde{x}_1-v)}{\sqrt{N_{\max}}(\tilde{x}_1-v)} = \sqrt{N_{\max}}. \tag{49}$$

Thus, ψ is decreasing from $\psi(0) = \|\mathbf{x}\|_1/\|\mathbf{x}\|_2 \leq \sqrt{N}$ (Lemma 2) to $\psi(v) = \sqrt{N_{\max}}$. This implies that for $c \in [\sqrt{N_{\max}};\sqrt{N}]$, there is an integer $i \in 1,\ldots,N$ such that $\psi(\tilde{x}_i) \leq c < \psi(\tilde{x}_{i+1})$. Finally, because ψ is continuous, there is a real number δ in $[0;\tilde{x}_i-\tilde{x}_{i+1}[$ such that $\psi(\tilde{x}_i-\delta) = c$.

(**iii**). Using the notations $\ell_1 = \|S(\tilde{\mathbf{x}},\tilde{x}_i)\|_1$ and $\ell_2 = \|S(\tilde{\mathbf{x}},\tilde{x}_i)\|_2$, with i (and δ) defined as previously stated in (**ii**), we have:

$$\begin{aligned}\|S(\tilde{\mathbf{x}},\tilde{x}_i-\delta)\|_1 &= \sum_{j=1}^{i}(\tilde{x}_j-(\tilde{x}_i-\delta)) \\ &= \sum_{j=1}^{i}(\tilde{x}_j-\tilde{x}_i)+i\delta \\ &= \|S(\tilde{\mathbf{x}},\tilde{x}_i)\|_1 + i\delta \\ &= \ell_1 + i\delta\end{aligned} \tag{50}$$

and

$$\begin{aligned}\|S(\tilde{\mathbf{x}},\tilde{x}_i-\delta)\|_2^2 &= \sum_{j=1}^{i}(\tilde{x}_j-(\tilde{x}_i-\delta))^2 \\ &= \sum_{j=1}^{i}((\tilde{x}_j-\tilde{x}_i)^2+2\delta(\tilde{x}_j-\tilde{x}_i)+\delta^2) \\ &= \ell_2^2 + 2\delta\ell_1 + i\delta^2.\end{aligned} \tag{51}$$

Moreover, since

$$\psi(\tilde{x}_i-\delta) = c = \frac{\|S(\tilde{\mathbf{x}},\tilde{x}_i-\delta)\|_1}{\|S(\tilde{\mathbf{x}},\tilde{x}_i-\delta)\|_2}, \tag{52}$$

the following equality holds:

$$\|S(\tilde{\mathbf{x}}, \tilde{x}_i - \delta)\|_1^2 = c^2 \|S(\tilde{\mathbf{x}}, \tilde{x}_i - \delta)\|_2^2. \tag{53}$$

Incorporating Eqs 50 and 51 into Eq 53 gives:

$$\delta^2(i^2 - ic^2) + 2\delta\ell_1(i - c^2) + \ell_1^2 - c^2\ell_2^2 = 0 \ . \tag{54}$$

The goal is now to find the positive root of this second degree polynomial equation. The discriminant Δ is equal to $4c^2(c^2 - i)(\ell_1^2 - i\ell_2^2)$. It remains to show that Δ is positive.

First, the number of non-zero elements of $S(\tilde{\mathbf{x}}, \tilde{x}_{i+1})$ is equal to i and Lemma 2 yields $\|S(\tilde{\mathbf{x}}, \tilde{x}_{i+1})\|_1 \leq \sqrt{i}\|S(\tilde{\mathbf{x}}, \tilde{x}_{i+1})\|_2$. Second, $\psi(\tilde{x}_{i+1}) = \dfrac{\|S(\tilde{\mathbf{x}}, \tilde{x}_{i+1})\|_1}{\|S(\tilde{\mathbf{x}}, \tilde{x}_{i+1})\|_2} > c$ so $\|S(\tilde{\mathbf{x}}, \tilde{x}_{i+1})\|_1 > c\|S(\tilde{\mathbf{x}}, \tilde{x}_{i+1})\|_2$. Combining the two previous inequalities yields $(i - c^2)\|S(\tilde{\mathbf{x}}, \tilde{x}_{i+1})\|_2^2 > 0$ which implies that i–$c^2 > 0$. Third, from $\psi(\tilde{x}_i) = \ell_1/\ell_2 \leq c < \sqrt{i}$, we deduce that $\ell_1^2 - i\ell_2^2 \leq 0$ which ensures that Δ is positive.

To conclude, the sign of $\dfrac{\ell_1^2 - c^2\ell_2^2}{i^2 - ic^2}$ corresponds to the sign of the product of the 2 roots. As this term is negative, the 2 roots have opposite signs. The single solution of $\psi(\tilde{x}_i - \delta) = c$ is:

$$\begin{aligned} \delta &= \frac{-2\ell_1(i - c^2) + \sqrt{\Delta}}{2i(i - c^2)} \\ &= \frac{-2\ell_1(i - c^2) + 2c\sqrt{[c^2 - i][\ell_1^2 - i\ell_2^2]}}{2i(i - c^2)} \\ &= -\frac{\ell_1}{i} + \frac{c}{i}\sqrt{\frac{i\ell_2^2 - \ell_1^2}{i - c^2}}. \end{aligned} \tag{55}$$

Using the fact that $\psi(\tilde{x}_i) = \ell_1/\ell_2$, the previous equation can be simplified as

$$\delta = \frac{\|S(\tilde{\mathbf{x}}, \tilde{x}_i)\|_2}{i}\left(c\sqrt{\frac{i - \psi(\tilde{x}_i)^2}{i - c^2}} - \psi(\tilde{x}_i)\right). \tag{56}$$

We deduce from this a four step algorithm, called PL1L2, for the projection onto the intersection of the L_1-ball of radius c and the L_2-ball of radius 1.

Algorithm 7: PL1L2: an algorithm for a fast and exact projection onto $\mathscr{B}_{L_1}(c) \cap \mathscr{B}_{L_2}(1)$.

Data: $\mathbf{x}$, c

Result: $\text{proj}_{\mathscr{B}_{L_1}(c)\cap\mathscr{B}_{L_2}(1)}(\mathbf{x})$

1. Take the absolute value of $\mathbf{x}$ and sort its elements in decreasing order to get $\tilde{\mathbf{x}}$;
2. Find i such that $\psi(\tilde{x}_{i+1}) \leq c < \psi(\tilde{x}_i)$;
3. Let $\delta = \dfrac{\|S(\tilde{\mathbf{x}}, \tilde{x}_i)\|_2}{i}\left(c\sqrt{\dfrac{i - \psi(\tilde{x}_i)^2}{i - c^2}} - \psi(\tilde{x}_i)\right)$;
4. Compute $S(\mathbf{x}, \lambda)$ with $\lambda = \tilde{x}_i - \delta$;

D Convergence of the CSVD algorithm

In this appendix we prove the convergence of the CSVD. To do so, we show that the CSVD is an instance of the *block successive upper-bound minimization (BSUM) algorithm* (introduced in [27]) and, as such, converges to a stationary point.

D.1 Definitions and notations

Define f, which is the negative of the objective function from Eq 15

$$f(\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix}) = -\mathbf{p}^\top \mathbf{X}\mathbf{q}. \tag{57}$$

The functions u_1 and u_2 are two "approximations" of f, defined as

$$u_1(\tilde{\mathbf{p}}; \begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix}) = f(\begin{bmatrix}\tilde{\mathbf{p}}\\ \mathbf{q}\end{bmatrix}) = -\tilde{\mathbf{p}}^\top \mathbf{X}\mathbf{q} \tag{58}$$

and

$$u_2(\tilde{\mathbf{q}}; \begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix}) = f(\begin{bmatrix}\mathbf{p}\\ \tilde{\mathbf{q}}\end{bmatrix}) = -\mathbf{p}^\top \mathbf{X}\tilde{\mathbf{q}} \tag{59}$$

These two functions depend on the fixed given vectors $\mathbf{p}$ and $\mathbf{q}$ and vary according to $\tilde{\mathbf{p}} \in \mathbb{R}^I$ and $\tilde{\mathbf{q}} \in \mathbb{R}^J$.

In the BSUM framework, f is minimized by iteratively minimizing u_1 over a convex set $\mathscr{P} \subseteq \mathbb{R}^I$ and u_2 over a convex set $\mathscr{Q} \subseteq \mathbb{R}^J$.

Definition 1 (BSUM algorithm [27]). *The BSUM algorithm (in the present setting) is defined as*:

1. *Minimize* u_1 *over* $\mathscr{P}$ *with* $\mathbf{q}$ *fixed, and update* $\mathbf{p}$ *with the solution*;
2. *Minimize* u_2 *over* $\mathscr{Q}$ *with* $\mathbf{p}$ *fixed, and update* $\mathbf{q}$ *with the solution*,

and iterate until convergence.

Recall the following definitions.

Definition 2 (Directional derivative). *Let* g *be a function with gradient at* $\mathbf{x}$ *denoted* $\nabla_{\mathbf{x}} g$. *The directional derivative of* g *in a direction* $\mathbf{d}$ *is*

$$g'(\mathbf{x} \,|\, \mathbf{d}) = \langle \nabla_{\mathbf{x}} g \,|\, \mathbf{d} \rangle. \tag{60}$$

Definition 3 (Regularity). *Let* f *be a differentiable function defined over* $\mathscr{P} \times \mathscr{Q}$. *Assume that*

$$f'(\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} || \begin{bmatrix}\mathbf{d}_1\\ \mathbf{0}\end{bmatrix}) \geq 0 \tag{61}$$

and

$$f'(\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} || \begin{bmatrix}\mathbf{0}\\ \mathbf{d}_2\end{bmatrix}) \geq 0\,, \tag{62}$$

with $\mathbf{d}_1 \in \mathbb{R}^I$ *and* $\mathbf{d}_2 \in \mathbb{R}^J$. *If this implies that*

$$f'(\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} || \begin{bmatrix}\mathbf{d}_1\\ \mathbf{d}_2\end{bmatrix}) \geq 0\,, \tag{63}$$

then f *is regular*.

D.2 Equivalence of the BSUM algorithm and the CSVD

Algorithm 5 is equivalent to the BSUM algorithm because:

1. Minimizing f is the same as maximizing the objective function in Eq 15.
2. Minimizing u_1 over $\mathscr{P}$ is equivalent to the left projection step.
3. Similarly, minimizing u_2 is equivalent to the right projection step.

D.3 Convergence of the BSUM algorithm

In order to converge to a stationary point, the BSUM algorithm needs to meet a few key assumptions that are specified in the following theorem (adapted from Theorem 2 in [27]).

Theorem 2 (Convergence). *The BSUM algorithm converges to a stationary point under the following conditions*:

1. *f is regular*,
2. u_1, u_2 *and f coincide (condition (B1) in* [27]*)*

$$u_1(p; \left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right]) = u_2(\mathbf{q}; \left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right]) = f(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right]), \quad \forall \mathbf{p} \in \mathscr{P}, \mathbf{q} \in \mathscr{Q} \tag{64}$$

3. u_1, u_2 *are upper bounds of f (condition (B2) in* [27]*),* $\forall \tilde{\mathbf{p}}, \mathbf{p} \in \mathscr{P}$, *and* $\forall \tilde{\mathbf{q}}, \mathbf{q} \in \mathscr{Q}$

$$\mathbf{u}_1(\tilde{\mathbf{p}}; \left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right]) \geq f(\left[\begin{smallmatrix}\tilde{\mathbf{p}}\\ \mathbf{q}\end{smallmatrix}\right]) \quad \textit{and} \quad \mathbf{u}_2(\tilde{\mathbf{q}}; \left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right]) \geq f(\left[\begin{smallmatrix}\mathbf{p}\\ \tilde{\mathbf{q}}\end{smallmatrix}\right]), \tag{65}$$

4. *the directional derivatives of* u_1, u_2 *and f coincide (condition (B3) in* [27]*)*

$$\mathbf{u}_1'(\tilde{\mathbf{p}}; \left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \mathbf{d}_1)|_{\tilde{\mathbf{p}}=\mathbf{p}} = f'(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \left[\begin{smallmatrix}\mathbf{d}_1\\ \mathbf{0}\end{smallmatrix}\right]), s.t. \mathbf{p} + \mathbf{d}_1 \in \mathscr{P}, \tag{66}$$

and

$$\mathbf{u}_2'(\tilde{\mathbf{q}}; \left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \mathbf{d}_2)|_{\tilde{\mathbf{q}}=\mathbf{p}} = f'(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \left[\begin{smallmatrix}\mathbf{0}\\ \mathbf{d}_2\end{smallmatrix}\right]), s.t. \mathbf{q} + \mathbf{d}_2 \in \mathscr{Q}, \tag{67}$$

5. u_1 *and* u_2 *are continuous functions (condition (B4) in* [27]*).*

D.3.1 Regularity. We show in this section that *f* is regular. The gradient of *f* with respect to its arguments, **p** and **q**, is defined as

$$\nabla f(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right]) = \begin{bmatrix} -\mathbf{x}_{\mathbf{q}} \\ -\mathbf{x}^{\top}{}_{\mathbf{p}} \end{bmatrix}. \tag{68}$$

Thus, the directional derivative of *f* in the direction $\mathbf{d} = \left[\begin{smallmatrix}\mathbf{d}_1\\ \mathbf{d}_2\end{smallmatrix}\right]$, with $\mathbf{d}_1 \in \mathbb{R}^I$ and $d_2 \in \mathbb{R}^J$ is equal to

$$f'(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \mathbf{d}) = \nabla f(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right])^{\top} \mathbf{d} = \underbrace{-\mathbf{d}_1^{\top}\mathbf{X}\mathbf{q}}_{f'(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \left[\begin{smallmatrix}\mathbf{d}_1\\ \mathbf{0}\end{smallmatrix}\right])} + \underbrace{-\mathbf{d}_2^{\top}\mathbf{X}^{\top}\mathbf{p}}_{f'(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \left[\begin{smallmatrix}\mathbf{0}\\ \mathbf{d}_2\end{smallmatrix}\right])}. \tag{69}$$

Hence, if $f'(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \left[\begin{smallmatrix}\mathbf{d}_1\\ \mathbf{0}\end{smallmatrix}\right]) \geq 0$ and $f'(\left[\begin{smallmatrix}\mathbf{p}\\ \mathbf{q}\end{smallmatrix}\right] | \left[\begin{smallmatrix}\mathbf{0}\\ \mathbf{d}_2\end{smallmatrix}\right]) \geq 0$, then the directional derivative of *f* in the direction of **d** is also positive, which proves that *f* is regular.

D.3.2 (B1), (B2), and (B3) in Razaviyayn et al., 2013. Because u_1 and u_2 are equal to the function *f* with either **p** or **q** fixed, u_1 and u_2 coincide with *f*, which proves (B1). Necessarily, so do their directional derivatives, which proves (B3). Finally because they coincide, u_1 and u_2 are upper bounds of *f* [in the sense of the condition (B2) in [27]].

D.3.3 Continuity: (B4) in Razaviyayn et al., 2013. Being compositions of linear operations, the functions u_1 and u_2 are both continuous.

D.4 Conclusion

It follows from these properties that the CSVD method, as described in Algorithm 5 being based on alternating between applying the projected power method with respect to **p** and to **q**, is a particular instance of the *block successive upper-bound minimization (BSUM) algorithm* ([27]). Therefore, any limit point of the CSVD method is a stationary point and so the CSVD converges to a stationary solution of Eq 15.

Supporting information

S1 Table. Simulated data. Additional results on broader settings.
(PDF)

Author Contributions

Conceptualization: Vincent Guillemot, Hervé Abdi.

Data curation: Vincent Guillemot, Derek Beaton, Brian Levine, Hervé Abdi.

Formal analysis: Vincent Guillemot, Arnaud Gloaguen, Tommy Löfstedt, Nicolas Raymond, Hervé Abdi.

Methodology: Vincent Guillemot, Hervé Abdi.

Software: Vincent Guillemot.

Supervision: Vincent Guillemot, Arthur Tenenhaus, Hervé Abdi.

Validation: Hervé Abdi.

Visualization: Vincent Guillemot.

Writing – original draft: Vincent Guillemot, Hervé Abdi.

Writing – review & editing: Vincent Guillemot, Derek Beaton, Arnaud Gloaguen, Tommy Löfstedt, Hervé Abdi.

References

1. Abdi H. Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). In: Salkind NJ, editor. Encyclopedia of Measurement and Statistic. Thousand Oaks (CA): Sage; 2007. p. 907–912.
2. Greenacre M. Correspondence analysis. New-York: Academic Press; 1984.
3. Lebart L, Morineau A, Warwick KM. Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices. New-York: Wiley; 1984.
4. Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. 2016; 374 (2065). https://doi.org/10.1098/rsta.2015.0202
5. Hotelling H. Relations between two sets of variates. Biometrika. 1936 dec; 28(3-4):321–377. https://doi.org/10.1093/biomet/28.3-4.321
6. Greenacre M. Correspondence analysis in practice. Boca Raton: Chapman and Hall; 2010.
7. Abdi H, Williams LJ. Partial least squares methods: Partial least squares correlation and partial least square regression. In: Reisfeld B, Mayeno A, editors. Methods in Molecular Biology: Computational Toxicology. New-York: Springer Verlag; 2013. p. 549–579.
8. Abdi H, Williams LJ. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistic. 2010; 2:97–106. https://doi.org/10.1002/wics.101
9. Efron B, Hastie T. Computer Age Statistical Inference. Cambridge: Cambridge University Press; 2016.
10. Trendafilov NT. From simple structure to sparse components: a review. Computational Statistics. 2014 jun; 29(3-4):431–454. Available from: http://link.springer.com/10.1007/s00180-013-0434-5.

11. Witten DM, Tibshirani R, Hastie T. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics. 2009; 10(3):515–534. https://doi.org/10.1093/biostatistics/kxp008 PMID: 19377034
12. Lu Z, Zhang Y. An augmented Lagrangian approach for sparse principal component analysis. Mathematical Programming. 2012 oct; 135(1-2):149–193. Available from: http://link.springer.com/10.1007/s10107-011-0452-4.
13. Genicot M, Huang W, Trendafilov NT. Weakly Correlated Sparse Components with Nearly Orthonormal Loadings. In: GSI: International Conference on Geometric Science of Information. Palaiseau, France: Springer, Cham; 2015. p. 484–490. Available from: http://link.springer.com/10.1007/978-3-319-25040-3_52.
14. Benidis K, Sun Y, Babu P, Palomar DP. Orthogonal Sparse PCA and Covariance Estimation via Procrustes Reformulation. IEEE Transactions on Signal Processing. 2016 dec; 64(23):6211–6226. Available from: http://ieeexplore.ieee.org/document/7558183/.
15. Allen GI, Grosenick L, Taylor J. A Generalized Least-Square Matrix Decomposition. Journal of the American Statistical Association. 2014 jan; 109(505):145–159. https://doi.org/10.1080/01621459.2013.852978
16. Combettes PL. The foundations of set theoretic estimation. Proceedings of the IEEE. 1993; 81(2):182–208. https://doi.org/10.1109/5.214546
17. Bauschke HH, Combettes PL. Convex Analysis and Monotone Operator Theory in Hilbert Spaces. 2nd ed. New-York, NY: Springer Verlag; 2017.
18. Zou H, Hastie T, Tibshirani R. Sparse principal component analysis. Journal of computational and graphical statistics. 2006; 15(2):265–286. https://doi.org/10.1198/106186006X113430
19. Hastie T, Tibshirani R, Wainwright M. Statistical Learning with Sparsity: The Lasso and Generalizations. Boca Raton: CRC Press; 2015.
20. Mattei PA, Bouveyron C, Latouche P. Globally Sparse Probabilistic PCA. In: Gretton A, Robert CC, editors. Proceedings of the 19th International Conference on Artificial Intelligence and Statistics. vol. 51 of Proceedings of Machine Learning Research. Cadiz, Spain: PMLR; 2016. p. 976–984. Available from: http://proceedings.mlr.press/v51/mattei16.html.
21. Mackey L. Deflation Methods for Sparse PCA. In: Advances in Neural Information Processing Systems; 2009. p. 1017–1024.
22. Jenatton R, Audibert JY, Bach F. Structured Variable Selection with Sparsity-Inducing Norms. The Journal of Machine Learning Research. 2011; 12:2777–2824.
23. Chun H, Keleş S. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. Journal of the Royal Statistical Society Series B, Statistical methodology. 2010 jan; 72(1):3–25. https://doi.org/10.1111/j.1467-9868.2009.00723.x PMID: 20107611
24. Le Floch E, Guillemot V, Frouin V, Pinel P, Lalanne C, Trinchera L, et al. Significant correlation between a set of genetic polymorphisms and a functional brain network revealed by feature selection and sparse Partial Least Squares. NeuroImage. 2012 oct; 63(1):11–24. https://doi.org/10.1016/j.neuroimage.2012.06.061 PMID: 22781162
25. Silver M, Janousova E, Hua X, Thompson PM, Montana G, Alzheimer's Disease Neuroimaging Initiative TADN. Identification of gene pathways implicated in Alzheimer's disease using longitudinal imaging phenotypes with sparse regression. NeuroImage. 2012 nov; 63(3):1681–94. https://doi.org/10.1016/j.neuroimage.2012.08.002 PMID: 22982105
26. Gloaguen A, Guillemot V, Tenenhaus A. An efficient algorithm to satisfy ℓ_1 and ℓ_2 constraints. In: 49èmes Journées de statistique. Avignon, France; 2017. p. 1–6. Available from: http://jds2017.sfds.asso.fr/program/Soumissions/subm306.pdf.
27. Razaviyayn M, Hong M, Luo ZQ. A Unified Convergence Analysis of Block Successive Minimization Methods for Nonsmooth Optimization. SIAM Journal on Optimization. 2013 Jun; 23(2):1126–1153. https://doi.org/10.1137/120891009
28. Boyd SP, Vandenberghe L. Convex optimization. 1st ed. Cambridge: Cambridge University Press; 2004.
29. Blajenkova O, Kozhevnikov M, Motes MA. Object-spatial imagery: a new self-report imagery questionnaire. Applied Cognitive Psychology. 2006 mar; 20(2):239–263. https://doi.org/10.1002/acp.1182
30. Valentin D, Abdi H, Edelman B. From rotation to disfiguration: Testing a dual-strategy model for recognition of faces across view angles. Perception. 1999; 28:817–824. https://doi.org/10.1068/p2932 PMID: 10664774
31. Abdi H, Beaton D. Principal Component and Correspondence Analyses Using R. New York: Springer Verlag; 2019.

32. Turk MA, Pentland AP. Face recognition using eigenfaces. In: Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Comput. Sco. Press; 1991. p. 586–591. Available from: http://ieeexplore.ieee.org/document/139758/.

33. Abdi H. Factor rotations in factor analysis. In: Salkind NJ, editor. Encyclopedia for Research Methods for the Social Sciences. Thousand Oaks (CA): Sage; 2003. p. 792–795.

34. Meinshausen N, Bühlmann P. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2010 jul; 72(4):417–473. https://doi.org/10.1111/j.1467-9868.2010.00740.x

35. Van Den Berg E, Schmidt M, Friedlander MP, Murphy K. Group Sparsity via Linear-Time Projection; 2008. Available from: http://www.optimization-online.org/DB_FILE/2008/07/2056.pdf.

36. Candes EJ, Romberg JK. Signal recovery from random projections. In: Bouman CA, Miller EL, editors. Computational Imaging III, Proceedings of Electronic Imaging 2005. vol. 5674. International Society for Optics and Photonics; 2005. p. 76. Available from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.600722.

37. Daubechies I, Fornasier M, Loris I. Accelerated Projected Gradient Method for Linear Inverse Problems with Sparsity Constraints. Journal of Fourier Analysis and Applications. 2008 dec; 14(5-6):764–792. Available from: http://link.springer.com/10.1007/s00041-008-9039-8.

38. Duchi J, Shalev-Shwartz S, Singer Y, Chandra T. Efficient projections onto the l1-ball for learning in high dimensions. In: Proceedings of the 25th international conference on Machine learning—ICML'08. New York, New York, USA: ACM Press; 2008. p. 272–279. Available from: http://portal.acm.org/citation.cfm?doid=1390156.1390191.