Structured Sparse Principal Components Analysis with the TV-Elastic Net Penalty

Amicie de Pierrefeu, Tommy Löfstedt, Fouad Hadj-Selem, Mathieu Dubois, Renaud Jardri, Thomas Fovet, Philippe Ciuciu *Senior Member*, Vincent Frouin and Edouard Duchesnay

***Abstract*—Principal component analysis (PCA) is an exploratory tool widely used in data analysis to uncover dominant patterns of variability within a population. Despite its ability to represent a data set in a low-dimensional space, PCA's interpretability remains limited. Indeed, the components produced by PCA are often noisy or exhibit no visually meaningful patterns. Furthermore, the fact that the components are usually non-sparse may also impede interpretation, unless arbitrary thresholding is applied. However, in neuroimaging, it is essential to uncover clinically interpretable phenotypic markers that would account for the main variability in the brain images of a population. Recently, some alternatives to the standard PCA approach, such as Sparse PCA, have been proposed, their aim being to limit the density of the components. Nonetheless, sparsity alone does not entirely solve the interpretability problem in neuroimaging, since it may yield scattered and unstable components. We hypothesized that the incorporation of prior information regarding the structure of the data may lead to improved relevance and interpretability of brain patterns. We therefore present a simple extension of the popular PCA framework that adds structured sparsity penalties on the loading vectors in order to identify the few stable regions in the brain images that capture most of the variability. Such structured sparsity can be obtained by combining *e.g.*, ℓ_1 and total variation (TV) penalties, where the TV regularization encodes information on the underlying structure of the data. This paper presents the structured sparse PCA (denoted SPCA-TV) optimization framework and its resolution. We demonstrate SPCA-TV's effectiveness and versatility on three different data sets. It can be applied to any kind of structured data, such as *e.g.*, N-dimensional array images or meshes of cortical surfaces. The gains of SPCA-TV over unstructured approaches (such as Sparse PCA and ElasticNet PCA) or structured approach (such as GraphNet PCA) are significant, since SPCA-TV reveals the variability within a data set in the form of intelligible brain patterns that are easier to interpret and more stable across different samples.**

***Keywords*—MRI, unsupervised machine learning, PCA, total variation.**

I. Introduction

Principal components analysis (PCA) is an unsupervised statistical procedure whose aim is to capture dominant patterns of variability in order to provide an optimal representation of a data set in a lower-dimensional space defined by the principal components (PCs). Given a data set $\mathbf{X} \in \mathbb{R}^{N \times P}$ of N samples and P centered variables, PCA aims to find the most accurate rank-K approximation of the data:

$$\min_{\mathbf{U},\mathbf{D},\mathbf{V}} \left\|\mathbf{X} - \mathbf{U}\mathbf{D}\mathbf{V}^T\right\|_F^2, \quad \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{V}^T\mathbf{V} = \mathbf{I}, d_1 \geq \cdots \geq d_K > 0 \tag{1}$$

where $\|.\|_F$ is the Frobenius norm of a matrix, $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_K] \in \mathbb{R}^{P \times K}$ are the K loading vectors (right singular vectors) that define the new coordinate system where the original features are uncorrelated, $\mathbf{D}$ is the diagonal matrix of the K singular values, and $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_K] \in \mathbb{R}^{N \times K}$ are the K projections of the original samples in the new coordinate system (called principal components (PCs) or left singular vector). Using $K = \text{rank}(\mathbf{X})$ components leads to the singular value decomposition (SVD). A vast majority of neuroimaging problems involve high-dimensional feature spaces ($\approx 10^5$ features *i.e.* voxels or mesh (nodes over the cortical surface) with a relatively limited sample size ($\approx 10^2$ participants. With such "large P, small N" problems, the SVD formulation, based on the data matrix, is much more efficient than an eigenvalue decomposition of the large $P \times P$ covariance matrix.

In a neuroimaging context, our goal is to discover the phenotypic markers accounting for the main variability in a population's brain images. For example, when considering structural images of patients that will convert to Alzheimer disease (AD), we are interested in revealing the brain patterns of atrophy explaining the variability in this population. This provides indications of possible stratification of the cohort into homogeneous sub-groups that may be clinically similar but with a different pattern of atrophy. This could suggest different sub-types of patients with AD or some other etiologies such as dementia with Lewy bodies. Clustering methods might be natural approaches to address such situations, however, they can not reveal subtle differences that go beyond a global and trivial pattern of atrophy. Such patterns are usually captured by the first component of PCA which, after being removed, offers the possibility to identify spatial patterns on the subsequent components.

However, PCA provides dense loading vectors (patterns), that cannot be used to identify brain markers without arbitrary thresholding.

Recently, some alternatives propose to add sparsity in this matrix factorization problem [33], [36], [43]. The sparse dictionary learning framework proposed by [36] provides a sparse coding (rows of $\mathbf{U}$) of samples through a sparse

A. de Pierrefeu, E. Duchesnay, P. Ciuciu, M. Dubois and V. Frouin are with NeuroSpin, CEA, Paris-Saclay, Gif-sur-Yvette - France

F. Hadj-Selem is with the Energy Transition Institute: VeDeCoM - France.

T. Löfstedt is with Department of Radiation Sciences, Umeå University, Umeå - Sweden.

R. Jardri and T. Fovet are with Univ Lille, CNRS UMR 9193, SCALab, CHU Lille, Pôle de Psychiatrie (unit CURE), Lille, France

linear combination of dense basis elements (columns of $\mathbf{V}$). However, the identification of biomarkers requires a sparse dictionary (columns of $\mathbf{V}$). This is precisely the objective of Sparse PCA (SPCA) proposed in [30], [51], [14], [49], [31] which adds a sparsity-inducing penalty on the columns of $\mathbf{V}$. Imposing such sparsity constraints on the loading coefficients is a procedure that has been used in fMRI to produce sparse representation of brain functional networks [20],[45].

However, sparse PCA is limited by the fact that it ignores the inherent spatial correlation in the data. It leads to scattered patterns that are difficult to interpret. Furthermore, constraining only the number of features included in the PCs might not always be fully relevant since most data sets are expected to have a spatial structure. For instance, MRI data is naturally encoded on a grid; some voxels are neighbors, while others are not.

We hypothesize that brain patterns are organized into distributed regions across the brain([11],[21],[41]). Recent studies tried to overcome this limitation by encoding prior information concerning the spatial structure of the data (see [29], [24], [48]). However, they used methods that are difficult to plug into the optimization scheme (*e.g.*, spline smoothing, wavelet smoothing) and incorporated prior information that sometimes may be difficult to define. One simple solution is the use of a GraphNet penalty ([23], [32], [40], [18], [38]). It promotes local smoothness of the weight map by simply forcing adjacent voxels to have similar weights using an λ_2 penalty on the gradient of the weight map. Nonetheless, we hypothesized that Graph-net provided smooth solution rather than clearly identified regions. In data classification problems, when extracting structured and sparse predictive maps, the goals are largely aligned with those of PCA. Some classification studies have revealed stable and interpretable results by adding a total variation (TV) penalty to the sparsity constraint (see [19]). TV is widely used as a tool in image denoising and restoration. It accounts for the spatial structure of images by encoding piecewise smoothness and enabling the recovery of homogeneous regions separated by sharp boundaries.

For simplicity, rather than solving Eq. (2), we solve a slightly different criterion which results from using the Lagrange form, rather than the bound form, of the constraints on $\mathbf{V}$. Then, we extend the Lagrangian form by adding penalties (ℓ_1, ℓ_2 and TV) to the minimization problem:

$$\min_{\mathbf{U},\mathbf{D},\mathbf{V}} \frac{1}{N}\|\mathbf{X}-\mathbf{U}\mathbf{D}\mathbf{V}^\top\|_F^2 + \sum_{k=1}^{K}\left\{\lambda_2\|\mathbf{v}_k\|_2^2+\lambda_1\|\mathbf{v}_k\|_1+\lambda\sum_{g\in\mathcal{G}}\|\mathbf{A}_g\mathbf{v}\|_2\right\}, \quad (2)$$
$$\text{s. t. } \|\mathbf{u}_k\|_2^2=1, \forall k=1,\cdots,K,$$

where λ_1, λ_2 and λ are hyper-parameters controlling the relative strength of each penalty. We further propose a generic optimization framework that can combine any differentiable convex (penalized) loss function with: (i) penalties whose proximal operator is known (here $\|\cdot\|_1$) and (ii) a large range of complex, non-smooth convex structured penalties that can be formulated as a $\|\cdot\|_{2,1}$-norm defined over a set of groups $\mathcal{G}$. Such group-penalties cover *e.g.*, total variation and overlapping group lasso.

This new problem aims at finding a linear combination of original variables that points in directions explaining as much variance as possible in data while enforcing sparsity and structure (piecewise smoothness for TV) of the loadings.

To achieve this, it is necessary to sacrifice some of the explained variance as well as the orthogonality of both the loading and the principal components. Most existing SPCA algorithms [51], [14], [49], [31], do not impose orthogonal loading directions either. While we forced the components to have unit norm for visualization purposes, we do not, in this formulation, enforce $\|\mathbf{v}_k\|_2 = 1$. Instead, the value of $\|\mathbf{v}\|_2$ is controlled by the hyper-parameter λ_2. This penalty on the loading, together with the unit norm constraint on the component, prevents us from obtaining trivial solutions. The optional $\frac{1}{N}$ factor acts on and conveniently normalizes the loss to account for the number of samples in order to simplify the settings of the hyper-parameters: $\lambda_1, \lambda_2, \lambda$.

This paper presents an extension of the popular PCA framework by adding structured sparsity-inducing penalties on the loading vectors in order to identify the few stable regions in the brain images accounting for most of the variability. The addition of a prior that reflects the data's structure within the learning process gives the paper a scope that goes beyond Sparse PCA. To our knowledge, very few papers ([1], [24], [29], [48]) addressed the use of structural constraint in PCA. The study [29] proposes a norm that induces structured sparsity (called SSPCA) by restraining the support of the solution to be sparse with a certain set of group of variables. Possible supports include set of variables forming rectangles when arranged on a grid. Only one study, recently used the total variation prior [1], in a context of multi-subject dictionary learning, based on a different optimization scheme [5].

Section II presents our main contribution: a simple optimization algorithm that combines well known methods (deflation scheme and alternate minimization) with an original continuation algorithm based on Nesterov's smoothing technique. Our proposed algorithm has the ability to include the TV penalty, but many other non-smooth penalties, such as *e.g.* overlapping group lasso, could also be used. This versatile mathematical framework is an essential feature in neuroimaging. Indeed, it enables a straightforward application to all kinds of data with known structure such as N-dimensional images (of voxels) or meshes of (cortical) surfaces. Section III demonstrates the relevance of structured sparsity on both simulated and experimental data, for structural and functional MRI (fMRI) acquisitions. SPCA-TV achieved a higher reconstruction accuracy and more stable solutions than ElasticNet PCA, Sparse PCA, GraphNet PCA and SSPCA (from [29]) . More importantly, SPCA-TV yields more interpretable loading vectors than other methods.

II. METHOD

A common approach to solve the PCA problem, see [14], [31], [49]), is to compute a rank-1 approximation of the data matrix, and then repeat this on the deflated matrix [34], where the influence of the PCs are successively extracted and

discarded. We first detail the notation for estimating a single component (Section II-A), and its solution using an alternating minimization pipeline (Section II-B). Then, we develop the TV regularization framework (Section II-C and Section II-D). Last, we discuss the algorithm used to solve the minimization problem and its ability to converge toward stable pairs of components/loading vectors (Section II-E) and (Section II-F).

A. Single component computation

Given a pair of loading/component vectors, $\mathbf{u} \in \mathbb{R}^N, \mathbf{v} \in \mathbb{R}^P$, the best rank-1 approximation of the problem given in Eq. (2) is equivalent [49] to:

$$\min_{\mathbf{u},\mathbf{v}} f \equiv \overbrace{\underbrace{-\frac{1}{N}\mathbf{u}^\top \mathbf{X}\mathbf{v} + \lambda_2\|\mathbf{v}\|_2^2}_{l(\mathbf{v})}}^{\text{smooth}} + \overbrace{\underbrace{\lambda_1 \|\mathbf{v}\|_1}_{h(\mathbf{v})} + \underbrace{\lambda \sum_{g\in\mathcal{G}} \|\mathbf{A}_g\mathbf{v}\|_2}_{s(\mathbf{v})}}^{\text{non-smooth}} \quad \text{s. t. } \|\mathbf{u}\|_2^2 \leq 1, \tag{3}$$

where $l(\mathbf{v})$ is the penalized smooth (*i.e.* differentiable) loss, $h(\mathbf{v})$ is a sparsity-inducing penalty whose proximal operator is known and $s(\mathbf{v})$ is a complex penalty on the structure of the input variables with an unknown proximal operator.

This problem is convex in $\mathbf{u}$ and in $\mathbf{v}$ but not in $(\mathbf{u}, \mathbf{v})$.

B. Alternating minimization of the bi-convex problem

The objective function to minimize is bi-convex [9]. The most common approach to solve a bi-convex optimization problem (which does not guarantee global optimality of the solution) is to alternatively update $\mathbf{u}$ and $\mathbf{v}$ by fixing one of them at the time and solving the corresponding convex optimization problem on the other parameter vector.

On the one hand, when $\mathbf{v}$ is fixed, the problem to solve is

$$\min_{\mathbf{u}\in\mathbb{R}^N} -\frac{1}{N}\mathbf{u}^\top \mathbf{X}\mathbf{v} \quad \text{s. t. } \|\mathbf{u}\|_2^2 \leq 1, \tag{4}$$

with the associated explicit solution

$$\mathbf{u}^*(\mathbf{v}) = \frac{\mathbf{X}\mathbf{v}}{\|\mathbf{X}\mathbf{v}\|_2}. \tag{5}$$

On the other hand, solving the equation with respect to $\mathbf{v}$ with a fixed $\mathbf{u}$ presents a higher level of difficulty that will be discussed in Section II-E.

C. Reformulating TV as a linear operator

Before discussing the minimization with respect to $\mathbf{v}$, we provide details on the encoding of the spatial structure within the $s(\mathbf{v})$ penalty.

It is essential to note that the algorithm is independent of the spatial structure of the data. All the structural information is encoded in a linear operator, $\mathbf{A}$, that is computed outside of the algorithm. Thus the algorithm has the ability to address various structured data and, most importantly, other penalties than just the TV penalty. The algorithm requires the setting of two parameters: (i) the linear operator $\mathbf{A}$; (ii) a projection function detailed in Eq. (12).

This section presents the formulation and the design of $\mathbf{A}$ in the specific case of a TV penalty applied to the loading vector $\mathbf{v}$ measured on a 3-dimensional (3D) image or a 2D mesh of the cortical surface.

1) 3D image: The brain mask is used to establish a mapping $g(i,j,k)$ between the coordinates (i,j,k) in the 3D grid, and an index $g \in [\![1;P]\!]$ in the collapsed image. We extract the spatial neighborhood of g, of size ≤ 4, corresponding to voxel g and its 3 neighboring voxels, within the mask, in the i,j and k directions. By definition, we have

$$\mathrm{TV}(\mathbf{v}) \equiv \sum_{g=1}^{P} \left\| \nabla\left(\mathbf{v}_{g(i,j,k)}\right) \right\|_2. \tag{6}$$

The first order approximation of the spatial gradient $\nabla(\mathbf{v}_{g(i,j,k)})$ is computed by applying the linear operator $\mathbf{A}_g^{'} \in \mathbb{R}^{3\times 4}$ to the loading vector $\mathbf{v}_g$ in the spatial neighborhood of g, *i.e.*

$$\nabla\left(\mathbf{v}_{g(i,j,k)}\right) = \underbrace{\begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_g'} \underbrace{\begin{bmatrix} v_{g(i,j,k)} \\ v_{g(i+1,j,k)} \\ v_{g(i,j+1,k)} \\ v_{g(i,j,k+1)} \end{bmatrix}}_{\mathbf{v}_g}, \tag{7}$$

where $v_{g(i,j,k)}$ is the loading coefficient at index g in the collapsed image corresponding to voxel (i,j,k) in the 3D image. Then $\mathbf{A}_g^{'}$ is extended, using zeros, to a large but very sparse matrix $\mathbf{A}_g \in \mathbb{R}^{3\times P}$ in order to be directly applied on the full vector $\mathbf{v}$. If some neighbors lie outside the mask, the corresponding rows in $\mathbf{A}_g$ are removed. Noticing that for TV there is one group per voxel in the mask ($\mathcal{G} = [\![1;P]\!]$), we can reformulate TV from Eq. (6) using a general expression:

$$\mathrm{TV}(\mathbf{v}) = \sum_{g\in\mathcal{G}} \|\mathbf{A}_g\mathbf{v}\|_2. \tag{8}$$

Finally, with a vertical concatenation of all the $\mathbf{A}_g$ matrices, we obtain the full linear operator $\mathbf{A} \in \mathbb{R}^{3P\times P}$ that will be used in Section II-E.

2) Mesh of cortical surface: The linear operator $\mathbf{A}_g^{'}$ used to compute a first order approximation of the spatial gradient can be obtained by examining the neighboring vertices of each vertex g. With common triangle-tessellated surfaces, the neighborhood size is ≤ 7 (including g). In this setting, we have $\mathbf{A}_g^{'} \in \mathbb{R}^{3\times 7}$, which can be extended and concatenated to obtain the full linear operator $\mathbf{A}$.

D. Nesterov's smoothing of the structured penalty

We consider the convex non-smooth minimization of Eq. (3) with respect to $\mathbf{v}$, where thus $\mathbf{u}$ is fixed. This problem includes a general structured penalty, $s(\cdot)$, that covers the specific case of TV. A widely used approach when dealing with non-smooth problems is to use methods based on the proximal operator of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2017.2749140, IEEE Transactions on Medical Imaging

the penalties. For the ℓ_1 penalty alone, the proximal operator is analytically known and efficient iterative algorithms such as ISTA and FISTA are available (see [4]). However, since the proximal operator of the TV+ℓ_1 penalty is not closed form, standard implementation of those algorithms is not suitable. In order to overcome this barrier, we used Nesterov's smoothing technique [39]. It consists of approximating the non-smooth penalties for which the proximal operator is unknown (*e.g.*, TV) with a smooth function (of which the gradient is known). Non-smooth penalties with known proximal operators (*e.g.*, ℓ_1) are not affected. Hence, as described in [50], it allows to use an exact accelerated proximal gradient algorithm. Thus we can solve the PCA problem penalized by TV and elastic net, where an exact ℓ_1 penalty is used.

Using the dual norm of the ℓ_2-norm (which happens to be the ℓ_2-norm too), Eq. (8) can be reformulated as

$$s(\mathbf{v}) = \sum_{g\in\mathcal{G}} \|\mathbf{A}_g\mathbf{v}\|_2 = \sum_{g\in\mathcal{G}} \max_{\|\boldsymbol{\alpha}_g\|_2\leq 1} \boldsymbol{\alpha}_g^\top \mathbf{A}_g\mathbf{v}, \tag{9}$$

where $\boldsymbol{\alpha}_g \in \mathcal{K}_g = \{\boldsymbol{\alpha}_g \in \mathbb{R}^3 : \|\boldsymbol{\alpha}_g\|_2 \leq 1\}$ is a vector of auxiliary variables, in the ℓ_2 unit ball, associated with $\mathbf{A}_g\mathbf{v}$. As with $\mathbf{A} \in \mathbb{R}^{3P\times P}$ which is the vertical concatenation of all the $\mathbf{A}_g$, we concatenate all the $\boldsymbol{\alpha}_g$ to form the $\boldsymbol{\alpha} \in \mathcal{K} = \{[\boldsymbol{\alpha}_1^\top,\ldots,\boldsymbol{\alpha}_P^\top]^\top : \boldsymbol{\alpha}_g \in \mathcal{K}_g\} \in \mathbb{R}^{3P}$. $\mathcal{K}$ is the Cartesian product of 3D unit balls in Euclidean space and, therefore, a compact convex set. Eq. (9) can further be written as

$$s(\mathbf{v}) = \max_{\boldsymbol{\alpha}\in K} \boldsymbol{\alpha}^\top \mathbf{A}\mathbf{v}. \tag{10}$$

Given this formulation of $s(\mathbf{v})$, we can apply Nesterov's smoothing. For a given smoothing parameter, $\mu > 0$, the $s(\mathbf{v})$ function is approximated by the smooth function

$$s_\mu(\mathbf{v}) = \max_{\boldsymbol{\alpha}\in K} \left\{\boldsymbol{\alpha}^\top \mathbf{A}\mathbf{v} - \frac{\mu}{2}\|\boldsymbol{\alpha}\|_2^2\right\}, \tag{11}$$

for which $\lim_{\mu\to 0} s_\mu(\mathbf{v}) = s(\mathbf{v})$. Nesterov [39] demonstrates this convergence using the inequality in Eq. (15). The value of $\boldsymbol{\alpha}_\mu^*(\mathbf{v}) = [\boldsymbol{\alpha}_{\mu,1}^{*\top},\ldots,\boldsymbol{\alpha}_{\mu,g}^{*\top},\ldots,\boldsymbol{\alpha}_{\mu,P}^{*\top}]^\top$ that maximizes Eq. (11) is the concatenation of projections of vectors $A_g\mathbf{v} \in \mathbb{R}^3$ to the ℓ_2 ball ($\mathcal{K}_g$): $\boldsymbol{\alpha}_{\mu,g}^*(\mathbf{v}) = \mathrm{proj}_{\mathcal{K}_g}\left(\frac{A_g\mathbf{v}}{\mu}\right)$, where

$$\mathrm{proj}_{\mathcal{K}_g}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\|_2 \leq 1 \\ \frac{\mathbf{x}}{\|\mathbf{x}\|_2} & \text{otherwise.} \end{cases}. \tag{12}$$

The function s_μ, *i.e.* the Nesterov's smooth transform of s, is convex and differentiable. Its gradient given by [39]

$$\nabla(s_\mu)(\mathbf{v}) = \mathbf{A}^\top \boldsymbol{\alpha}_\mu^*(\mathbf{v}), \tag{13}$$

is Lipschitz-continuous with constant

$$L\big(\nabla(s_\mu)\big) = \frac{\|\mathbf{A}\|_2^2}{\mu}, \tag{14}$$

where $\|\mathbf{A}\|_2$ is the matrix spectral norm of $\mathbf{A}$. Moreover, Nesterov [39] provides the following inequality relating s_μ and s

$$s_\mu(\mathbf{v}) \leq s(\mathbf{v}) \leq s_\mu(\mathbf{v}) + \mu M, \quad \forall \mathbf{v} \in \mathbb{R}^p, \tag{15}$$

where $M = \max_{\boldsymbol{\alpha}\in\mathcal{K}} \frac{\|\boldsymbol{\alpha}\|_2^2}{2} = \frac{P}{2}$.

Thus, a new (smoothed) optimization problem, closely related to Eq. (3) (with fixed $\mathbf{u}$), arises from this regularization as

$$\min_{\mathbf{v}} \overbrace{\underbrace{-\frac{1}{n}\mathbf{u}^\top\mathbf{X}\mathbf{v} + \lambda_2\|\mathbf{v}\|_2^2}_{l(\mathbf{v})} + \underbrace{\lambda\left\{\boldsymbol{\alpha}_\mu^*(\mathbf{v})^\top\mathbf{A}\mathbf{v} - \frac{\mu}{2}\|\boldsymbol{\alpha}^*\|_2^2\right\}}_{s_\mu(\mathbf{v})}}^{\text{smooth}} + \overbrace{\lambda_1\underbrace{\|\mathbf{v}\|_1}_{h(\mathbf{v})}}^{\text{non-smooth}}. \tag{16}$$

Since we are now able to explicitly compute the gradient of the smooth part $\nabla(l + \lambda s_\mu)$ (Eq. (18)), its Lipschitz constant (Eq. (19)) and also the proximal operator of the non-smooth part, we have all the ingredients necessary to solve this minimization function using an accelerated proximal gradient methods [4]. Given a starting point $\mathbf{v}^0$ and a smoothing parameters μ, FISTA (Algorithm 1) minimizes the smoothed problem and reaches a prescribed precision ε_μ.

However, in order to control the convergence of the algorithm (presented in Section II-E1), we introduce the Fenchel dual function and the corresponding dual gap of the objective function. The Fenchel duality requires the loss to be strongly convex, which is why we further reformulate Eq. (16) slightly: All penalty terms are divided by λ_2 and by using the following equivalent formulation for the loss, we obtain the minimization problem

$$\min_{\mathbf{v}} f_\mu \equiv \overbrace{\underbrace{\frac{1}{2}\left\|\mathbf{v} - \frac{\mathbf{X}^\top\mathbf{u}}{n\lambda_2}\right\|_2^2}_{\mathcal{L}(\mathbf{v})} + \underbrace{\frac{1}{2}\|\mathbf{v}\|_2^2 + \frac{\lambda}{\lambda_2}\left\{\boldsymbol{\alpha}_\mu^*(\mathbf{v})^\top\mathbf{A}\mathbf{v} - \frac{\mu}{2}\|\boldsymbol{\alpha}^*\|_2^2\right\}}_{\psi_\mu(\mathbf{v})} + \frac{\lambda_1}{\lambda_2}\|\mathbf{v}\|_1}^{l(\mathbf{v}) \quad s_\mu(\mathbf{v}) \quad h(\mathbf{v})}. \tag{17}$$

This new formulation of the smoothed objective function (noted f_μ) preserves the decomposition of f_μ into a sum of a smooth term $l + \frac{\lambda}{\lambda_2}s_\mu$ and a non-smooth term h. Such decomposition is required for the application of FISTA with Nesterov's smoothing. Moreover, this formulation provides a decomposition of f_μ into a sum of a smooth loss $\mathcal{L}$ and a penalty term ψ_μ required for the calculation of the gap presented in Section II-E1).

We provide all the required quantities to minimize Eq. (17) using Algorithm 1. Using Eq. (13) we compute the gradient of the smooth part as

$$\begin{aligned}\nabla\left(l + \frac{\lambda}{\lambda_2}s_\mu\right) &= \nabla(l) + \frac{\lambda}{\lambda_2}\nabla(s_\mu) \\ &= (2\mathbf{v} - \frac{\mathbf{X}^\top\mathbf{u}}{n\lambda_2}) + \frac{\lambda}{\lambda_2}\mathbf{A}^\top\boldsymbol{\alpha}_\mu^*(\mathbf{v}^k),\end{aligned} \tag{18}$$

and its Lipschitz constant (using Eq. (14))

$$L\left(\nabla\left(l + \frac{\lambda}{\lambda_2}s_\mu\right)\right) = 2 + \frac{\lambda}{\lambda_2}\frac{\|\mathbf{A}\|_2^2}{\mu}. \tag{19}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2017.2749140, IEEE Transactions on Medical Imaging

Algorithm 1 FISTA$\big(\mathbf{X}^\top\mathbf{u},\ \mathbf{v}^0,\ \varepsilon_\mu,\ \mu,\ \mathbf{A},\ \lambda,\ L(\nabla(g))\big)$

1: $\mathbf{v}^1 = \mathbf{v}^0$; $k = 2$
2: Compute the gradient of the smooth part $\nabla(g + \lambda s_\mu)$ (Eq. (18)) and its Lipschitz constant L_μ (Eq. (19)).
3: Compute the size $t_\mu = L_\mu^{-1}$
4: **repeat**
5: $\quad \mathbf{z} = \mathbf{v}^{k-1} + \frac{k-2}{k+1}\left(\mathbf{v}^{k-1} - \mathbf{v}^{k-2}\right)$
6: $\quad \mathbf{v}^k = \mathrm{prox}_h\big(\mathbf{z} - t_\mu \nabla(g + \lambda s_\mu)(\mathbf{z})\big)$
7: **until** $\textsc{Gap}_\mu(\mathbf{v}^k) \leq \varepsilon_\mu$
8: **return** $\mathbf{v}^k$

E. Minimization of the loading vectors with CONESTA

The step size t_μ computed in Line 3 of Algorithm 1, depends on the smoothing parameter μ (see Eq. (19)). Hence, there is a trade-off between speed and precision. Indeed, high precision, with a small μ, will lead to a slow convergence (small t_μ). Conversely, poor precision (large μ) will lead to rapid convergence (large t_μ). Thus we propose a continuation approach (Algorithm 2) which decreases the smoothing parameter with respect to the distance to the minimum. On the one hand, when we are far from $\mathbf{v}^*$ (the minimum of Eq. (17)), we can use a large μ to rapidly decrease the objective function. On the other hand, when we are close to $\mathbf{v}^*$, we need a small μ in order to obtain an accurate approximation of the original objective function.

1) Duality gap: The distance to the unknown $f(\mathbf{v}^*)$ is estimated using the duality gap. Duality formulations are often used to control the achieved precision level when minimizing convex functions. They provide an estimation of the error $f(\mathbf{v}^k) - f(\mathbf{v}^*)$, for any $\mathbf{v}$, when the minimum is unknown. The duality gap is the cornerstone of the CONESTA algorithm. Indeed, it is used three times:

i As the stopping criterion in the inner FISTA loop (Line 7 in Algorithm 1). FISTA will stop as soon as the current precision is achieved using the current smoothing parameter, μ. This prevents unnecessary convergence toward the approximated (smoothed) objective function.

ii In the ith CONESTA iteration, as a way to estimate the current error $f(\mathbf{v}^i) - f(\mathbf{v}^*)$ (Line 7 in Algorithm 2). The error is estimated using the gap of the smoothed problem $\textsc{Gap}_{\mu=\mu^i}(\mathbf{v}^{i+1})$ which avoid unnecessary computation since it has already been computed during the last iteration of FISTA. The inequality in Eq. (15) is used to obtain the gap ε^i to the original non-smoothed problem. The next desired precision ε^{i+1} and the smoothing parameter, μ^{i+1}, are derived from this value.

iii Finally, as the global stopping criterion within CONESTA (Line 10 in Algorithm 2). This will guarantee that the obtained approximation of the minimum, $\mathbf{v}^i$, at convergence, satisfies $f(\mathbf{v}^i) - f(\mathbf{v}^*) < \varepsilon$.

Based on Eq. (17), which decomposes the smoothed objective function as a sum of a strongly convex loss and the penalty,

$$f_\mu(\mathbf{v}) = \mathcal{L}(\mathbf{v}) + \psi_\mu(\mathbf{v}),$$

we compute the duality gap that provides an upper bound estimation of the error to the optimum. At any step k of the algorithm, given the current primal $\mathbf{v}^k$ and the dual $\sigma(\mathbf{v}^k) \equiv \nabla\mathcal{L}(\mathbf{v}^k)$ variables [8], we can compute the duality gap using the Fenchel duality rules [35]:

$$\textsc{Gap}(\mathbf{v}^k) \equiv f_\mu(\mathbf{v}^k) + \mathcal{L}^*\big(\sigma(\mathbf{v}^k)\big) + \psi_\mu^*\big(-\sigma(\mathbf{v}^k)\big), \quad (20)$$

where $\mathcal{L}^*$ and ψ_μ^* are respectively the Fenchel conjugates of $\mathcal{L}$ and ψ_μ. Denoting by $\mathbf{v}^*$ the minimum of f_μ (solution of Eq. (17)), the interest of the duality gap is that it provides an upper bound for the difference with the optimal value of the function. Moreover, it vanishes at the minimum:

$$\begin{aligned} \textsc{Gap}(\mathbf{v}^k) \geq f(\mathbf{v}^k) - f(\mathbf{v}^*) &\geq 0 \\ \textsc{Gap}(\mathbf{v}^*) &= 0. \end{aligned} \quad (21)$$

The dual variable is

$$\sigma(\mathbf{v}^k) \equiv \nabla\mathcal{L}(\mathbf{v}^k) = \mathbf{v} - \frac{\mathbf{X}^\top\mathbf{u}}{n\lambda_2}, \quad (22)$$

the Fenchel conjugate of the squared loss $\mathcal{L}(\mathbf{v}^k)$ is

$$\mathcal{L}^*(\sigma(\mathbf{v}^k)) = \frac{1}{2}\|\sigma(\mathbf{v}^k)\|_2^2 + \sigma(\mathbf{v}^k)^\top\frac{\mathbf{X}^\top\mathbf{u}}{n\lambda_2}. \quad (23)$$

In [25] the authors provide the expression of the Fenchel conjugate of the penalty $\psi_\mu(\mathbf{v}^k)$:

$$\begin{aligned} \psi_\mu^*(-\sigma(\mathbf{v}^k)) =& \frac{1}{2}\sum_{j=1}^{P}\left(\left[\left|-\sigma(\mathbf{v}^k)_j - \frac{\lambda}{\lambda_2}\big(\mathbf{A}^\top\boldsymbol{\alpha}_\mu^*(\mathbf{v}^k)\big)_j\right| - \frac{\lambda_1}{\lambda_2}\right]_+^2\right) \\ &+ \frac{\lambda\mu}{2\lambda_2}\left\|\boldsymbol{\alpha}_\mu^*(\mathbf{v}^k)\right\|_2^2 \end{aligned} \quad (24)$$

where $[\cdot]_+ = \max(0, \cdot)$.

The expression of the duality gap in Eq. (20) provides an estimation of the distance to the minimum. This distance is geometrically decreased by a factor $\tau = 0.5$ at the end of each continuation, and the decreased value defines the precision that should be reached by the next iteration (Line 8 of Algorithm 2). Thus, the algorithm dynamically generates a sequence of decreasing prescribed precisions ε^i. Such a scheme ensures the convergence [25] towards a globally desired final precision, ε, which is the only parameter that the user needs to provide.

2) Determining the optimal smoothing parameter: Given the current prescribed precision ε^i, we need to compute an optimal smoothing parameter $\mu_{opt}(\varepsilon^i)$ (Line 9 in Algorithm 2) that minimizes the number of FISTA iterations needed to achieve such precision when minimizing Eq. (3) (with fixed $\mathbf{u}$) via Eq. (17) (*i.e.*, such that $f(\mathbf{v}^{(k)}) - f(\mathbf{v}^*) < \varepsilon^i$).

In [25], the authors provide the expression of this optimal smoothing parameter:

$$\mu_{opt}(\varepsilon^i) = \frac{-\lambda M\|\mathbf{A}\|_2^2 + \sqrt{(\lambda M\|\mathbf{A}\|_2^2)^2 + ML(\nabla(l))\|\mathbf{A}\|_2^2\varepsilon^i}}{ML(\nabla(l))}, \quad (25)$$

where $M = P/2$ (Eq. (15)) and $L(\nabla(l)) = 2$ is the Lipschitz constant of the gradient of l as defined in Eq. (17).

We call the resulting algorithm CONESTA (short for **CO**ntinuation with **NE**sterov smoothing in a **S**hrinkage-**T**hresholding **A**lgorithm). It is presented in detail, with convergence proofs in [25].

Let K be the total number of FISTA loops used in CONESTA, then we have experimentally verified that the convergence rate to the solution of Eq. (16) is $O\big(1/K^2\big)$ (which is the optimal convergence rate for first-order methods). Also, the algorithm works even if some of the weights λ_1 or λ are zero, which thus allows us to solve *e.g.*, the elastic net using CONESTA. Note that it has been rigorously proved that the continuation technique improves the convergence rate compared to the simple smoothing using a single value of μ. Indeed, it has been demonstrated in [6] (see also [50]) that the convergence rate obtained with single value of μ, even optimised, is $O\big(1/K^2\big) + O(1/K)$. However, it has recently been proved in [25] that the CONESTA algorithm achieves a $O(1/K)$ for general convex functions.

We note that CONESTA could easily be adapted to many other penalties. For example, to add the group lasso (GL) constraint to our structure, we just have to design a specific linear operator $\mathbf{A}_{GL}$ and concatenate it to the actual linear operator $\mathbf{A}$.

Algorithm 2 CONESTA$\big(\mathbf{X}^\top\mathbf{u},\, \varepsilon\big)$

Initialize $\mathbf{v}^0 \in \mathbb{R}^P$
$\varepsilon^0 = \tau \cdot \text{GAP}_{\mu=10^{-8}}(\mathbf{v}^0)$
$\mu^0 = \mu_{opt}\left(\varepsilon^0\right)$
repeat
  $\varepsilon_\mu^i = \varepsilon^i - \mu^i \gamma M$
  $\mathbf{v}^{i+1} = \text{FISTA}(\mathbf{X}^\top\mathbf{u},\, \mathbf{v}^i,\, \varepsilon_\mu^i,\, \ldots)$
  $\varepsilon^i = \text{GAP}_{\mu=\mu^i}(\mathbf{v}^{i+1}) + \mu^i \gamma M$
  $\varepsilon^{i+1} = \tau \cdot \varepsilon^i$
  $\mu^{i+1} = \mu_{opt}\big(\varepsilon^{i+1}\big)$
until $\varepsilon^i \leq \varepsilon$
return $\mathbf{v}^{i+1}$

F. The algorithm for the SPCA-TV problem

The computation of a single component through SPCA-TV can be achieved by combining CONESTA and Eq. (5) within an alternating minimization loop. Mackey [34] demonstrated that further components can be efficiently obtained by incorporating this single-unit procedure in a deflation scheme as done in *e.g.* [14], [31]. The stopping criterion is defined as

$$\text{STOPPINGCRITERION} = \frac{\left\|\mathbf{X}^k - \mathbf{u}^{i+1}\mathbf{v}^{i+1\top}\right\|_F - \left\|\mathbf{X}^k - \mathbf{u}^i\mathbf{v}^{i\top}\right\|_F}{\left\|\mathbf{X}^k - \mathbf{u}^{i+1}\mathbf{v}^{i+1\top}\right\|_F}. \tag{26}$$

All the presented building blocks were combined into Algorithm 3 to solve the SPCA-TV problem.

Algorithm 3 SPCA-TV$(\mathbf{X},\, \varepsilon\big)$

$\mathbf{X}_0 = \mathbf{X}$
for all $k = 0, \ldots, K$ **do** ▷ Components
  Initialize $\mathbf{u}^0 \in \mathbb{R}^N$
  repeat ▷ Alternating minimization
    $\mathbf{v}^{i+1} = \text{CONESTA}(\mathbf{X}_k^\top\mathbf{u}^i, \varepsilon)$
    $\mathbf{u}^{i+1} = \frac{\mathbf{X}_k\mathbf{v}^{i+1}}{\|\mathbf{X}_k\mathbf{v}^{i+1}\|_2}$
  until STOPPINGCRITERION $\leq \varepsilon$
  $\mathbf{v}_{k+1} = \mathbf{v}^{i+1}$
  $\mathbf{u}_{k+1} = \mathbf{u}^{i+1}$
  $\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{u}^{k+1}\mathbf{v}^{k+1\top}$ ▷ Deflation
end for
return $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_K], \mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_K]$

III. EXPERIMENTS

We evaluated the performance of SPCA-TV using three experiments: One simulation study carried out on a synthetic data set and two on neuroimaging data sets. In order to compare the performance of SPCA-TV with existing sparse PCA models, we also included results obtained with Sparse PCA, ElasticNet PCA, GraphNet PCA and SSPCA from [29]. We used the scikit-learn implementation [42] for the Sparse PCA while we used the Parsimony package (https://github.com/neurospin/pylearn-parsimony) for the ElasticNet, GraphNet PCA and SPCA-TV methods. Concerning SSPCA, we used the MATLAB implementation provided in [29].

The number of parameters to set for each method is different: For Sparse PCA, the λ_1 parameter selects its optimal value from the range $\{0.1\ ,\ 1.0\ ,\ 5.0\ ,\ 10.0\}$. ElasticNet PCA requires the setting of the λ_1 and the λ_2 penalties weights. Meanwhile, GraphNet PCA and SPCA-TV requires the settings of an additional parameter, namely the spatial constraint penalty λ. We operated a re-parametrization of these penalty weights in ratios. A global parameter $\alpha \in \{0.01,\ 0.1,\ 1.0\}$ controls the weight attributed to the whole penalty term, including the spatial and the ℓ_1 regularization. Individual constraints are expressed in terms of ratios: the ℓ_1 ratio: $\lambda_1/(\lambda_1 + \lambda_2 + \lambda)$, $\in \{0.1,\ 0.5,\ 0.8\}$ and the ℓ_{TV} (or ℓ_{GN} for GraphNet) : $\lambda/(\lambda_1 + \lambda_2 + \lambda)$, $\in \{0.1,\ 0.5,\ 0.8\}$. For ElasticNet, we explore the grid of parameters composed of the Cartesian product of α and ℓ_1 ratio subsets. For GraphNet PCA and SPCA-TV, we perform a parameter search on a grid of parameters given by the Cartesian product of respectively $(\alpha,\ \ell_1\ \ell_{GN})$ subsets and $(\alpha,\ \ell_1\ \ell_{TV})$ subsets . Concerning SSPCA method, the regularization parameter selects its optimal value in the range $\{10^{-8}, ..., 10^8\}$

However, in order to ensure that the components extracted have a minimum amount of sparsity, we also included a criteria controlling sparsity: At least half of the features of the components have to be zero. For both real neuroimaging experiments, performance was evaluated through a 5-fold x 5-fold double cross validation pipeline. The double cross-validation process consists of two nested cross-validation loops which are referred to as internal and external cross-validation loops. In the outer (external) loop, all samples are randomly split into subsets referred to as training and test sets. The test sets are exclusively used for model assessment while the train sets are used in the inner (internal) loop for model fitting

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2017.2749140, IEEE Transactions on Medical Imaging

and selection. The inner folds select the set of parameters minimizing the reconstruction error on the outer fold. For the synthetic data, we used 50 different purposely-generated data sets and 5 inner folds for parameters selection. In order to evaluate the reconstruction accuracy of the methods, we reported the mean Frobenius norm of the reconstruction error across the folds/data sets, on independent test data. The hypothesis we wanted to test was whether there was a substantial decrease in the reconstruction error of independent data when using SPCA-TV compared to when using Sparse PCA, ElasticNet PCA, GraphNet PCA and SSPCA . It was tested through a related two samples t-test. This choice to compare methods performance on independent test data was motivated by the fact that the optimal reconstruction of the training set is necessarily hindered by spatial and sparsity constraints. We therefore expect SPCA-TV to perform worse on train data than other less constrained methods. However, the TV penalty has a more important purpose than just to minimize the reconstruction error: the estimation of coherent and reproducible loadings. Indeed, clinicians expect that, if images from other patients with comparable clinical conditions had been used, the extracted loading vectors would have turned out to be similar. Therefore, since the ultimate goal of SPCA-TV is to yield stable and reproducible weight maps, it is more relevant to evaluate methods on independent test data.

The stability of the loading vectors obtained across various training data sets (variation in the learning samples) was assessed through a similarity measure: the pairwise Dice index between loading vectors obtained with different folds/data sets [16]. We tested whether pairwise Dice indices are significantly higher in SPCA-TV compared other methods. Testing this hypothesis is equivalent to testing the sign of the difference of pairwise Dice indices between methods. However, since the pairwise Dice indices are not independent from one another (the folds share many of their learning samples), the direct significance measures are biased. We therefore used permutation testing to estimate empirical p-values. The null hypothesis was tested by simulating samples from the null distribution. We generated $1\,000$ random permutations of the sign of the difference of pairwise Dice index between the PCA methods under comparisons, and then the statistics on the true data were compared with the ones obtained on the reshuffled data to obtain empirical p-values.

For each experiment, we made the initial choice to retrieve the first ten components. However, given the length constraint, we only present the weights maps associated to the top three components for Sparse PCA and SPCA-TV in this paper. ElasticNet PCA, GraphNet PCA and SSPCA 's weights maps of experiments are presented in the supplementary materials (supplementary materials are available in the supplementary files /multimedia tab).

A. Simulation study

We generated 50 sets of synthetic data, each composed of 500 images of size 100×100 pixels. Images are generated using the following noisy linear system :

$$u_1 V^1 + u_2 V^2 + u_3 V^3 + \epsilon \in \mathbb{R}^{10\,000}, \tag{27}$$

where $V = [V^1, V^2, V^3] \in \mathbb{R}^{10\,000 \times 3}$ are sparse and structured loading vectors, illustrated in Fig. 1. The support of V^1 defines the two upper dots, the support of V^2 defines the two lower dots, while V^3 's support delineates the middle dot. The coefficients $u = [u_1, u_2, u_3]$ that linearly combine the components of V are generated according to a centered Gaussian distribution. The elements of the noise vector ϵ are independent and identically distributed according to a centered Gaussian distribution with a 0.1 signal-to-noise ratio (SNR). This SNR was selected by a previous calibration pipeline, where we tested the efficiency of data reconstruction at multiple SNR ranges running from 0 to 0.5. We decided to work with a 0.1 SNR because it is located in the range of values where standard PCA starts being less efficient in the recovery process.

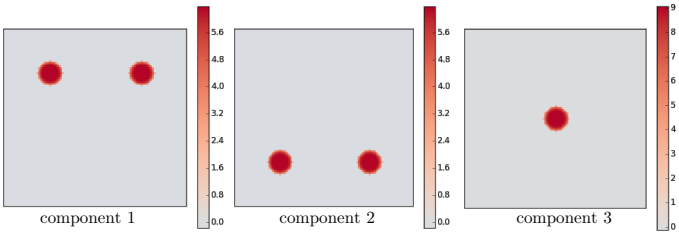

Fig. 1: Loading vectors $V = [V^1, V^2, V^3] \in \mathbb{R}^{10\,000 \times 3}$ used to generate the images

We splitted the 500 artificial images into a test and a training set, with 250 images in each set and learned the decomposition on the training set.

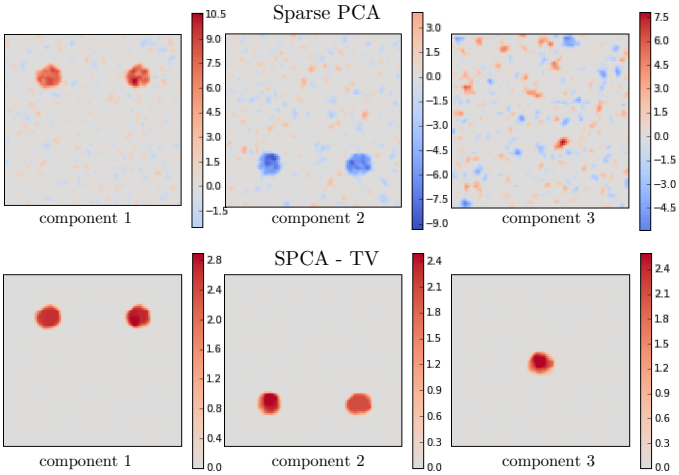

Fig. 2: Loading vectors recovered from 250 images using Sparse PCA and SPCA-TV.

Fig. 2 represents the loading vectors extracted with one data set. Please note that the sign is arbitrary. Indeed, if we consider the loss of Eq. (3), $\mathbf{u}^\top$ and $\mathbf{v}$ can be both multiply by -1

TABLE I: Scores are averaged across the 50 independent data sets. We tested whether the scores obtained with existing PCA methods are significantly different from scores obtained with SPCA-TV. Significance notations: ***: $p \leq 10^{-3}$

	Scores		
Methods	Test Data Reconstruction Error	MSE	Dice Index
Sparse PCA	1576.0***	0.91***	0.28***
ElasticNet PCA	1572.4***	0.83***	0.43***
GraphNet PCA	1570.8***	0.83***	0.30***
SSPCA	1571.9***	1.54***	0.07***
SPCA-TV	1570.1	0.64	0.52

without changing anything. We observe that Sparse PCA yields very scattered loading vectors. The loading vectors of SPCA-TV, on the other hand, are sparse; but also organized in clear regions. SPCA-TV provides loading vectors that closely match the ground truth. The reconstruction error is evaluated on the test sets (Tab. I), with its value over the 50 data sets being significantly lower in SPCA-TV than in Sparse PCA ($T = 94.5$, $p = 3.9 \cdot 10^{-57}$), ElasticNet PCA ($T = 33.2$, $p = 2.7 \cdot 10^{-35}$, GraphNet PCA ($T = 12.7$, $p = 3.6 \cdot 10^{-17}$ and SSPCA from [29] ($T = 18.9$, $p = 3.9 \cdot 10^{-24}$)) methods. Additional details concerning the reconstruction accuracy of both the train and test data is presented in Figure 1 of supplementary materials (supplementary materials are available in the supplementary files /multimedia tab).

A different way of quantifying the reconstruction accuracy for each method is to evaluate how closely the extracted loadings match the known ground truth of simulated data set. We computed the mean squared error (MSE) between the ground truth and the estimated loadings. The results are presented in Tab. I. We note that the MSE is significantly lower with SPCA-TV than with Sparse PCA ($T = 6.9$, $p = 8.0 \cdot 10^{-9}$), ElasticNet PCA ($T = 6.2$, $p = 1.1 \cdot 10^{-07}$), GraphNet-PCA ($T = 4.1$, $p = 1.4 \cdot 10^{-04}$) and SSPCA ($T = 22.6$, $p = 1.5 \cdot 10^{-27}$).

Moreover, when evaluating the stability of the loading vectors across resampling, we found a higher statistically significant mean Dice index when using SPCA-TV compared to the other methods ($p < 0.001$). The results are presented in Tab. I. They indicate that SPCA-TV is more robust to variation in the learning samples than the other sparse methods. SPCA-TV yields reproducible loading vectors across data sets.

These results indicate that the SPCA-TV loadings are not only more stable across resampling but also achieve a better recovery of the underlying variability in independent data than the Sparse PCA, ElasticNet PCA,GraphNet PCA and SSPCA methods.

One of the issues linked to biconvex optimization is the risk of falling into locals minima. Conscious of this potential risk, we set up an experiment in which we ran 50 times the optimization of the same problem, with a different starting point at each run. We then compare the resulting loading vectors obtained at each run, and computed a similarity measure, the Dice index. It quantifies the proximity between each independently-run solution with a different starting point. We obtained a Dice index of 0.99 on the 1st component, 0.99 on the 2nd component, and 0.72 on the 3rd component. Off the strength of this indices, we are confident of this algorithm robustness and ability to converge toward the same stable solution independently from the choice of the starting point.

B. 3D images of functional MRI of patients with schizophrenia

We then applied the methods on 3D images of BOLD functional MRI (fMRI) acquired with the same scanner and pulse sequence. Imaging was performed on a 1.5 T scanner using a standard head-coil. For all functional scans the field-of-view was $206 * 206 * 153$ mm, with a resolution close to 3.5 mm in all directions. The parameters of the PRESTO sequence were TE = 9.6 ms, TR = 19.25 ms, EPI-factor = 15, flip angle = 9. Each fMRI run consisted of 900 volumes collected. The cohort is composed of 23 patients with schizophrenia (average age = 34.96 years, 8 Females/15 Males). Brain activation was measured while subjects experienced multimodal hallucinations. The fMRI data was pre-processed using SPM12, (WELLCOME Department of Imaging Neuroscience, London, UK). Data preprocessing consisted of motion correction (realignment), coregistration of the individual anatomical T1 image to the functional images, spatial normalization to MNI space using DARTEL based on segmented T1 scans.

We considered each set of consecutive images under pre-hallucinations state as a block. Since, most of the patients hallucinate more than once during the scanning session, we have more blocks than patients (83 blocks). The activation maps are computed from these blocks. Based on the general linear model approach, we regressed for each block, the fMRI signal time course on a linear ramp function: Indeed, we hypothesized that activation in some regions presents a ramp-like increase during the time preceding the onset of hallucinations. (See example of regression in figure 3 in the supplementary materials,available in the supplementary files /multimedia tab.). The activation maps that we used as an input to the SPCA-TV method are the statistical parametric maps associated with the coefficients of the block regression. (See one example in Figure 4 of supplementary materials, available in the supplementary files /multimedia tab). We obtained a data set of $n = 83$ maps and $p = 63\,966$ features. We hypothesized that the principal components extracted with SPCA-TV from these activation maps could uncover major trends of variability within pre-hallucination patterns. Thus, they might reveal the existence of subgroups of patients, according to the sensory modality (e.g., vision or audition) involved during hallucinations.

We applied all PCA methods under study to this data set except SSPCA. Indeed, the SSPCA method ([29]) could not be applied to this specific example since datasets have to be constituted from closed cubic forms without any holes to be eligible for SSPCA method application. It does not support masked data such as the one used here.

The loading vectors extracted from the activation maps of pre-hallucinations scans with Sparse PCA and SPCA-TV are presented in Fig. 3. We observe a similar behavior as in the synthetic example, namely that the loading vectors of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMI.2017.2749140, IEEE Transactions on Medical Imaging

Sparse PCA tend to be scattered and produce irregular patterns. However, SPCA-TV seems to yield structured and smooth sources of variability, which can be interpreted clinically. Furthermore, the SPCA-TV loading vectors are not redundant and revealed different patterns.

Indeed, the loading vectors obtained by SPCA-TV are of great interest because they revealed insightful patterns of variability in the data: the second loading is composed of interesting areas such as the precuneus cortex and the cingulate gyrus, but also areas related to vision-processing areas such as the occipital fusiform gyrus and the parietal operculum cortex regions. The third loading reveals important weights in the middle temporal gyrus, the parietal operculum cortex and the frontal pole. The first loading vector encompasses all features of the brain. One might see this first component as a global variability affecting the whole brain, such as the overarching effect of age. SPCA-TV selects this dense configuration in spite of the sparsity constraint: It is highly desirable to remove any sort of global effect at first, in order to start identifying local patterns in next components, that are not impacted by a global variability of this kind. We can identified a widespread set of dysfunctional language-related or vision-related areas that present increasing activity during the time preceding the onset of hallucinations. The regions extracted by SPCA-TV are found to be pertinent according to the existing literature on the topic. ([28], [27], [7]]).

These results seem to indicate the possible existence of subgroups of patients according to the hallucination modalities involved. An interesting application would be to use the score of the second component extracted by SPCA-TV in order to distinguish patients with visual hallucinations from those suffering mainly from auditory hallucinations.

The reconstruction error is significantly lower in SPCA-TV than in Sparse PCA ($T = 13.9$, $p = 1.5 \cdot 10^{-4}$), ElasticNet PCA ($T = 7.1$, $p = 2.1 \cdot 10^{-3}$) and GraphNet PCA ($T = 4.6$, $p = 1.0 \cdot 10^{-2}$). Moreover, when assessing the stability of the loading vectors across the folds, we found a higher statistically significant mean Dice index in SPCA-TV compared to: Sparse PCA ($p = 4.0 \cdot 10^{-3}$), ElasticNet PCA ($p = 4.0 \cdot 10^{-3}$) and GraphNet PCA ($p = 2.0 \cdot 10^{-3}$) as presented in Tab. II. Additional details regarding the reconstruction accuracy on both the train and test sets, and the Dice index, is presented in Figure 5 of supplementary materials (available in the supplementary files /multimedia tab).

TABLE II: Scores of the fMRI data are averaged across the 5 folds. We tested whether the averaged scores obtained with existing PCA methods are significantly different from scores obtained with SPCA-TV. Significance notations: ***: $p \leq 10^{-3}$, **: $p \leq 10^{-2}$

	Scores	
Methods	Test Data Reconstruction Error	Dice Index
Sparse PCA	1515.2***	0.34**
ElasticNet PCA	1482.7**	0.32**
GraphNet PCA	1428.1**	0.58**
SPCA-TV	1414.0	0.63

In conclusion, SPCA-TV significantly outperforms Sparse, ElasticNet and GraphNet PCA in terms of the reconstruction error on independent test data, and in the sense that loading vectors are both more clinically interpretable and more stable.

We also evaluated the convergence speed of Sparse PCA, Mini-Batch Sparse PCA (a variant of Sparse PCA that is faster but less accurate), ElasticNet PCA, GraphNet PCA and SPCA-TV for this functional MRI data set of $n = 83$ samples and $p = 63\,966$ features. We compared the time of execution required for each algorithm to achieve a given level of precision in Tab. III. Sparse PCA and ElasticNet PCA are similar in terms of convergence time, while mini-batch sparse PCA is much faster but does not converge to high precision. As expected, structured methods (GraphNet PCA and SPCA-TV) take longer than other sparse methods because of the inclusion

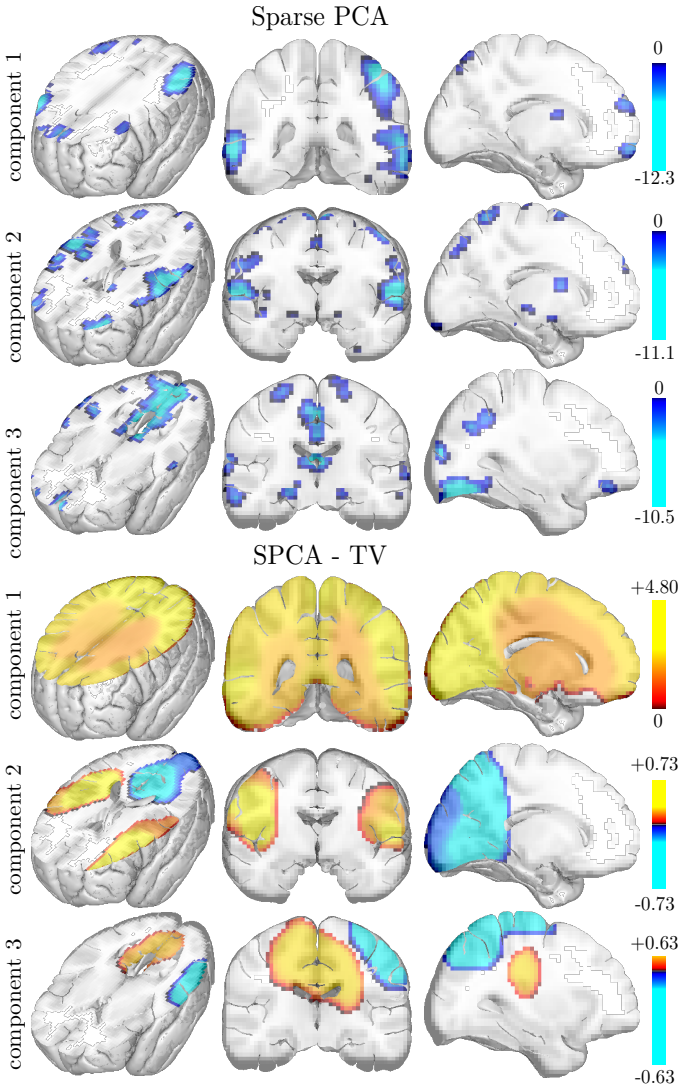

Fig. 3: Loading vectors recovered from the 83 activation maps using Sparse PCA and SPCA-TV.

TABLE III: Comparison of the execution time required for each sparse method to reach the same precision. Times reported in seconds.

	Time to reach a given precision in seconds				
Methods	10	1	10^{-1}	10^{-2}	10^{-3}
Mini-batch Sparse PCA	5.32	-	-	-	-
Sparse PCA	158.0	231.2	344.3	386.8	450.1
ElasticNet PCA	123.7	138.1	302.7	396.4	406.3
GraphNet PCA	301.9	521.6	813.1	881.4	888.4
SPCA-TV	427.7	2958.6	8093.0	13813.4	14459.9

of spatial constraints. Especially, SPCA-TV is much longer than other methods but the convergence time is still reasonable for an fMRI data set with $65\,000$ voxels.

C. Surfaces meshes of cortical thickness in Alzheimer disease

Finally, SPCA-TV was applied to the whole brain anatomical MRI from the ADNI database, the Alzheimer's Disease Neuroimaging Initiative, (http://adni.loni.usc.edu/). The MR scans are T1-weighted MR images acquired at $1.5\ T$ according to the ADNI acquisition protocol. We selected 133 patients with a diagnosis of mild cognitive impairments (MCI) from the ADNI database who converted to AD within two years during the follow-up period. We used PCA to reveal patterns of atrophy explaining the variability in this population. This could provide indication of possible stratification of the population into more homogeneous subgroups, that may be clinically similar, but with different brain patterns. In order to demonstrate the relevance of using SPCA-TV to reveal variability in any kind of imaging data, we worked on meshes of cortical thickness. The $317\,379$ features are the cortical thickness values at each vertex of the cortical surface. Cortical thickness represents a direct index of atrophy and thus is a potentially powerful candidate to assist in the diagnosis of Alzheimer's disease ([3], [17]). Therefore, we hypothesized that applying SPCA-TV to the ADNI data set would reveal important sources of variability in cortical thickness measurements. Cortical thickness measures were performed with the FreeSurfer image analysis suite (Massachusetts General Hospital, Boston, MA, USA), which is documented and freely available for download online (http://surfer.nmr.mgh.harvard.edu/). The technical details of this procedure are described in [46], [13] and [2]. All the cortical thickness maps were registered onto the FreeSurfer common template (fsaverage).

We applied all PCA methods under study to this data set except SSPCA. Indeed, we could not applied SSPCA method to this data set due to some intrinsic limitations of the method. SSPCA 's application is restricted to N-dimensional array images. It does not support meshes of cortical surfaces such as the data set used here.

The loading vectors obtained from the data set with sparse PCA and SPCA-TV are presented in Fig. 4. As expected, Sparse PCA loadings are not easily interpretable because the patterns are irregular and dispersed throughout the brain surface. In contrast, SPCA-TV reveals structured and smooth clusters in relevant regions.

The first loading vector, which maps the whole surface of the brain, can be interpreted as the variability between patients, resulting from a global cortical atrophy, as often observed in AD patients. The second loading vector includes variability in the entorhinal cortex, hippocampus and in temporal regions. Last, the third loading vector might be related to the atrophy of the frontal lobe and captures variability in the precuneus too. Thus, SPCA-TV provides a smooth map that closely matches the well-known brain regions involved in Alzheimer's disease.[22]

Indeed, it is well-documented that cortical atrophy progresses over three main stages in Alzheimer disease.([10], [15]) The cortical structures are sequentially being affected because of the accumulation of amyloid plaques. Cortical atrophy is first observed, in the mild stage of the disease, in regions surrounding the hippocampus ([26], [44], [47]) and the enthorinal cortex ([12]), as seen in the second component. This is consistent with early memory deficits. Then, the disease progresses to a moderate stage; where atrophy gradually extends to the prefrontal association cortex as revealed in the third component ([37]). In the severe stage of the disease, the whole cortex is affected by atrophy ([15]) (as revealed in the first component). In order to assess the clinical significance of these weight maps; we tested the correlation between the scores corresponding to the three components and performance on a clinical test: ADAS. The Alzheimer's Disease Assessment Scale-Cognitive subscale, is the most widely used general cognitive measure in AD. ADAS is scored in terms of errors, so a high score indicates poor performance. We obtained significant correlations between ADAS test performance and components 'scores in Fig. 5. $r = -0.34, p = 4.2 \cdot 10^{-11}$ for the first component, $r = -0.26, p = 3.6 \cdot 10^{-7}$ for the second component and $r = -0.35, p = 4.0 \cdot 4.5^{-12}$ for the third component) The same behavior is observable for all three components: The ADAS score grows proportionately to the level to which a patient is affected and to the severity of atrophy he presents (in temporal pole, prefrontal region and also globally). Conversely, controls subjects score low on the ADAS metric and present low level of cortical atrophy. Therefore, SPCA-TV provides us with clear biomarkers, that are perfectly relevant to the scope of Alzheimer's disease progression.

The reconstruction error is significantly lower in SPCA-TV than in Sparse PCA ($T = 12.7$, $p = 2.1 \cdot 10^{-4}$), ElasticNet PCA ($T = 6.8$, $p = 2.3 \cdot 10^{-3}$) and GraphNet PCA ($T = 2.83$, $p = 4.7 \cdot 10^{-2}$). The results are presented in Tab. IV. Moreover, when assessing the stability of the loading vectors across the folds, the mean Dice index is significantly higher in SPCA-TV than in other methods. Additional details regarding the reconstruction accuracy on both the train and test sets, and the Dice index, is presented in Figure 7 of supplementary materials (available in the supplementary files /multimedia tab).

IV. Conclusion

We proposed an extension of Sparse PCA that takes into account the spatial structure of the data. The optimization

TABLE IV: Scores are averaged across the 5 folds. We tested whether the averaged scores obtained with existing PCA methods are significantly lower from scores obtained with SPCA-TV. Significance notations: ***: $p \leq 10^{-3}$, **: $p \leq 10^{-2}$, *: $p \leq 10^{-1}$.

	Scores	
Methods	Test Data Reconstruction Error	Dice Index
Sparse PCA	2991.8***	0.44**
ElasticNet PCA	2832.6**	0.43**
GraphNet PCA	2813.6*	0.62*
SPCA-TV	2795.0	0.65

scheme is able to minimize any combination of the ℓ_1, ℓ_2, and TV penalties while preserving the exact ℓ_1 penalty. We observe that SPCA-TV, in contrast to other existing sparse PCA methods, yields clinically interpretable results and reveals major sources of variability in data, by highlighting structured clusters of interest in the loading vectors. Furthermore, SPCA-TV 's loading vectors were more stable across the learning samples compared to other methods. SPCA-TV was validated and its applicability was demonstrated on three distinct data sets: we may reach the conclusion that SPCA-TV can be used on any kind of structured configurations, and is able to present structure within the data.

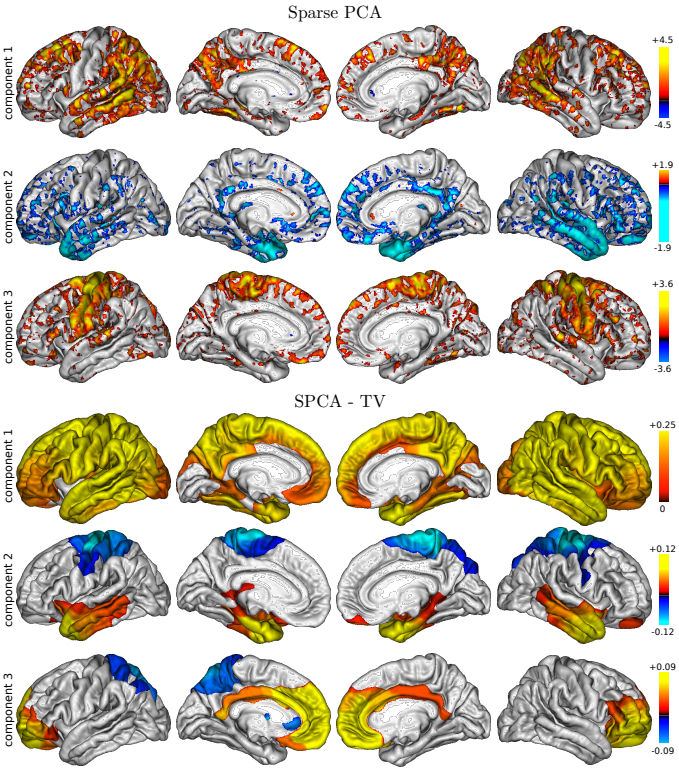

Fig. 4: Loading vectors recovered from the 133 MCI patients using Sparse PCA and SPCA-TV.

SUPPLEMENTARY MATERIAL

a) The ParsimonY Python library:

- *Url:* https://github.com/neurospin/pylearn-parsimony
- *Description:* ParsimonY is Python library for structured and sparse machine learning. ParsimonY is open-source (BSD License) and compliant with scikit-learn API.

b) Data sets and scripts:

- *Url:* ftp://ftp.cea.fr//pub/unati/brainomics/papers/pcatv
- *Description:* This url provides the simulation data set and the Python script used to create Fig.2 for the paper.

REFERENCES

[1] A. Abraham, E. Dohmatob, B. Thirion, D. Samaras, and G. Varoquaux. Extracting brain regions from rest fmri with total-variation constrained dictionary learning. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2013 - 16th International Conference, Nagoya, Japan, 2013, Proceedings, Part II*, pages 607–615, 2013.

[2] Bruce B. Fischl, M. Sereno, and A. Dale. Cortical surface-based analysis: Ii: Inflation, flattening, and a surface-based coordinate system. *NeuroImage*, 9(2):195 – 207, 1999.

[3] A. Bakkour, JC. Morris, and BC. Dickerson. The cortical signature of prodromal ad: regional thinning predicts mild ad dementia. *Neurology*, 72:1048–1055, 2009.

[4] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

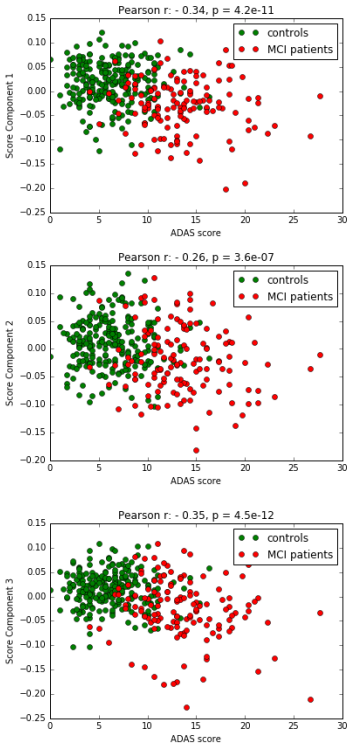

Fig. 5: Correlation of components scores with ADAS test performance

[5] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Processing*, 18(11):2419–2434, 2009.

[6] A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[7] L. Bentaleb, M. Beauregard, P. Liddle, and E. Stip. Cerebral activity associated with auditory verbal hallucinations: a functional magnetic resonance imaging case study. *Journal of psychiatry & neuroscience: JPN*, 27(2):110, 2002.

[8] J.M. Borwein and A.S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. CMS Books in Mathematics. Springer, 2006.

[9] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[10] H. Braak and E. Braak. Neuropathological stageing of alzheimer-related changes. *Acta Neuropathologica*, 82(4):239–259, 1991.

[11] K. Brodmann. Vergleichende lokalisationslehre der grosshirnrinde in ihren prinzipien dargestellt auf grund des zellenbaues. 1909.

[12] VA Cardenas, LL Chao, C Studholme, K Yaffe, BL Miller, C Madison, ST Buckley, D Mungas, N Schuff, and MW Weiner. Brain atrophy associated with baseline and longitudinal measures of cognition. *Neurobiology of aging*, 32(4):572–580, 2011.

[13] Anders Dale, Bruce Fischl, and Martin I. Sereno. Cortical surface-based analysis: I. segmentation and surface reconstruction. *NeuroImage*, 9(2):179 – 194, 1999.

[14] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A Direct Formulation for Sparse PCA Using Semidefinite Programming. *SIAM Review*, 49(3):434–448, 2007.

[15] A Delacourte, JP David, N Sergeant, L Buee, A Wattez, P Vermersch, F Ghozali, C Fallet-Bianco, F Pasquier, F Lebert, et al. The biochemical pathway of neurofibrillary degeneration in aging and alzheimers disease. *Neurology*, 52(6):1158–1158, 1999.

[16] L. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26:297–302, 1945.

[17] BC. Dickerson, E. Feczko, JC. Augustinack, J. Pacheco, JC. Morris, and B. Fischl. Differential effects of aging and alzheimer's disease on medial temporal lobe cortical thickness and surface area. *Neurobiology of aging*, 30:432–440, 2009.

[18] E. Dohmatob, M. Eickenberg, B. Thirion, and G. Varoquaux. Speeding-up model-selection in GraphNet via early-stopping and univariate feature-screening. June 2015.

[19] M. Dubois, F. Hadj-Selem, T. Löfstedt, M. Perrot, C. Fischer, V. Frouin, and É. Duchesnay. Predictive support recovery with TV-Elastic Net penalties and logistic regression: an application to structural MRI. In *Proceedings of the fourth International Workshop on Pattern Recognition in Neuroimaging (PRNI 2014)*, 2014.

[20] H. Eavani, TD. Satterthwaite, RE. Filipovych, RC. Gur, , and C. Davatzikos. Identifying sparse connectivity patterns in the brain using resting-state fmri. *Neuroimage*, 105:286–299, 2015.

[21] D. Felleman and D. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

[22] GB. Frisoni, NC. Fox, CR. Jack, P. Scheltens, and PM. Thompson. The clinical use of structural MRI in Alzheimer disease. *Nat Rev Neurol*, 6(2):67–77, 2010.

[23] L. Grosenick, B. Klingenberg, K. Katovich, B. Knutson, and J. Taylor. Interpretable whole-brain prediction analysis with GraphNet. *NeuroImage*, 72:304–321, May 2013.

[24] R. Guo, M. Ahn, H. Zhu, and the Alzheimers Disease Neuroimaging Initiative. Spatially weighted principal component analysis for imaging classification. *Journal of Computational and Graphical Statistics*, 24:274–296, 2015.

[25] F. Hadj-Selem, T. Lofstedt, V. Frouin, V. Guillemot, and E. Duchesnay. An Iterative Smoothing Algorithm for Regression with Structured Sparsity. *arXiv:1605.09658 [stat]*, 2016. arXiv: 1605.09658.

[26] CR Jack, MM Shiung, JL Gunter, PC Obrien, SD Weigand, David S Knopman, Bradley F Boeve, Robert J Ivnik, Glenn E Smith, RH Cha, et al. Comparison of different mri brain atrophy rate measures with clinical disease progression in ad. *Neurology*, 62(4):591–600, 2004.

[27] R. Jardri, A. Pouchet, D. Pins, and P. Thomas. Cortical activations during auditory verbal hallucinations in schizophrenia: a coordinate-based meta-analysis. *American Journal of Psychiatry*, 168(1):73–81, 2011.

[28] R. Jardri, P. Thomas, C. Delmaire, P. Delion, and D. Pins. The neurodynamic organization of modality-dependent hallucinations. *Cerebral Cortex*, pages 1108–1117, 2013.

[29] R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[30] I. Jolliffe, N. Trendafilov, and M. Uddin. A Modified Principal Component Technique Based on the LASSO. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[31] M. Journe, Y. Nesterov, P. Richtrik, and R. Sepulchre. Generalized Power Method for Sparse Principal Component Analysis. *J. Mach. Learn. Res.*, 11:517–553, 2010.

[32] B. Kandel, D. Wolk, J. Gee, and B. Avants. Predicting Cognitive Data from Medical Images Using Sparse Linear Regression. *Information processing in medical imaging : proceedings of the ... conference*, 23:86–97, 2013.

[33] M. Li, Y. Liu, F. Chen, and D. Hu. Including signal intensity increases the performance of blind source separation on brain imaging data. *IEEE transactions on medical imaging*, 34(2):551–563, 2015.

[34] Lester W. Mackey. Deflation Methods for Sparse PCA. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1017–1024. Curran Associates, Inc., 2009.

[35] J. Mairal. *Sparse coding for machine learning, image processing and computer vision*. PhD thesis, École normale supérieure, Cachan, 2010.

[36] J. Mairal, F. Bach, Jean J. Ponce, and G. Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *J. Mach. Learn. Res.*, 11:19–60, 2010.

[37] CR. McDonald, L. McEvoy, L. Gharapetian, C. Fennema-Notestine, DJ. Hagler, D. Holland, A. Koyama, JB. Brewer, AM. Dale, Alzheimers Disease Neuroimaging Initiative, et al. Regional rates of neocortical atrophy from normal aging to early alzheimer disease. *Neurology*, 73(6):457–465, 2009.

[38] H. Mohr, U. Wolfensteller, S. Frimmel, and H. Ruge. Sparse regularization techniques provide novel insights into outcome integration processes. *NeuroImage*, 104:163–176, January 2015.

[39] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[40] B. Ng, A. Vahdat, G. Hamarneh, and R. Abugharbieh. Generalized Sparse Classifiers for Decoding Cognitive States in fMRI. In *SpringerLink*, pages 108–115, Beijing, China, September 2012. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-15948-0_14.

[41] R. Nieuwenhuys. The myeloarchitectonic studies on the human cerebral cortex of the vogt–vogt school, and their significance for the interpretation of functional neuroimaging data. *Brain Structure and Function*, 218(2):303–352, 2013.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[43] M. Ramezani, K. Marble, H.Trang, and P. Abolmaesumi I.S. Johnsrude. Joint sparse representation of brain activity patterns in multi-task fmri data. *IEEE transactions on medical imaging*, 34(1):2–12, 2015.

[44] B. Ridha, V. Anderson, J. Barnes, R. Boyes, S. Price, M. Rossor, J. Whitwell, L. Jenkins, R. Black, Michae M. Grundman, et al.

Volumetric mri and cognitive measures in alzheimer disease. *Journal of neurology*, 255(4):567–574, 2008.

[45] H. Shen, H. Xu, L. Wang, Y. Lei, L. Yang, P. Zhang, J. Qin, L. Zeng, Z. Zhou, Z. Yang, and D. Hu. Making group inferences using sparse representation of resting-state functional mri data with application to sleep deprivation. *Human Brain Mapping*, 38(9):4671–4689, 2017.

[46] J.G Sled, A.P Zijdenbos, and A.C Evans. A nonparametric method for automatic correction of intensity nonuniformity in mri data. *IEEE Trans Med Imaging*, 17:87–97, 1998.

[47] P. Thompson, K. Hayashi, G. de Zubicaray, A. Janke, S. Rose, J. Semple, M. Hong, D. Herman, D. Gravano, D. Doddrell, and A. Toga. Mapping hippocampal and ventricular change in alzheimer disease. *NeuroImage*, 22(4):1754 – 1766, 2004.

[48] W.-T. Wang and H.-C. Huang. Regularized Principal Component Analysis for Spatial Data. *ArXiv e-prints*, 2015.

[49] D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[50] X.Chen, L. Qihang, K. Seyoung, J. Carbonell., and E. Xing. Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics*, 6(2):719–752, 2012.

[51] H. Zou, T. Hastie, and R. Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.