Structured Sparse Principal Component Analysis

Rodolphe Jenatton
rodolphe.jenatton@inria.fr

Guillaume Obozinski
guillaume.obozinski@inria.fr

Francis Bach
francis.bach@inria.fr

INRIA Willow Project, Laboratoire d'Informatique de l'Ecole Normale Supérieure
(INRIA/ENS/CNRS UMR 8548) - 23, avenue d'Italie, 75214 Paris, France.

Abstract

We present an extension of sparse PCA, or sparse dictionary learning, where the sparsity patterns of all dictionary elements are structured and constrained to belong to a prespecified set of shapes. This *structured sparse PCA* is based on a structured regularization recently introduced by Jenatton et al. (2009). While classical sparse priors only deal with *cardinality*, the regularization we use encodes higher-order information about the data. We propose an efficient and simple optimization procedure to solve this problem. Experiments with two practical tasks, the denoising of sparse structured signals and face recognition, demonstrate the benefits of the proposed structured approach over unstructured approaches.

1 Introduction

Principal component analysis (PCA) is an essential tool for data analysis and unsupervised dimensionality reduction. Its goal is to find, among linear combinations of the data variables, a sequence of orthogonal factors that most efficiently explain the variance of the observations.

One of PCA's main shortcomings is that, even if it finds a small number of important factors, the factor themselves typically involve all original variables. In the last decade, several alternatives to PCA which find sparse and potentially interpretable factors have been proposed, notably non-negative matrix factorization (NMF) (Lee and Seung, 1999) and sparse PCA (SPCA) (Jolliffe et al., 2003; Zou et al., 2006; Zass and Shashua, 2007; Witten et al., 2009).

However, in many applications, only constraining the size of the factors does not seem appropriate because the considered factors are not only expected to be sparse but also to have a certain structure. In fact, the popularity of NMF for face image analysis owes essentially to the fact that the method happens to retrieve sets of variables that are partly localized on the face and capture some features or parts of the face which seem intuitively meaningful given our a priori. We might therefore gain in the quality of the factors induced by enforcing directly this a priori in the matrix factorization constraints. More generally, it would be desirable to encode higher-order information about the supports that reflects the *structure* of the data. For example, in computer vision, features associated to the pixels of an image are naturally organized on a grid and the supports of factors explaining the variability of images could be expected to be localized, connected or have some other regularity with respect to that grid. Similarly, in genomics, factors explaining the gene expression patterns observed on a microarray could be expected to involve groups of genes corresponding to biological pathways or set of genes that are neighbors in a protein-protein interaction network.

Recent research on structured sparsity has highlighted the benefit of exploiting such structure in the context of regression and classification (Jenatton et al., 2009; Jacob et al., 2009; Huang et al., 2009), compressed sensing (Baraniuk et al., 2008), as well as within Bayesian frameworks (He and Carin, 2009). In particular, Jenatton et al. (2009) show that, given any intersection-closed family of patterns $\mathcal{P}$ of variables, such as all the rectangles on a 2-dimensional grid of variables, it is possible to build an ad hoc regularization norm Ω that enforces that the support of the solution of a least-squares regression regularized by Ω belongs to the family $\mathcal{P}$.

Capitalizing on these results, we aim in this paper to go beyond sparse PCA and propose *structured sparse PCA* (SSPCA), which explains the variance of the data by factors that are not only sparse but also respect some a priori structural constraints deemed relevant to model the data at hand. We show how slight variants of the regularization term from Jenatton et al. (2009) can be used successfully to yield a structured and sparse formulation of principal component analysis for which we propose a simple and efficient optimization scheme.

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

The rest of the paper is organized as follows: Section 2 casts the SSPCA problem in the dictionary learning framework, summarizes the regularization considered by Jenatton et al. (2009) and its essential properties, and presents some simple variants which are more effective in the context of PCA. Section 3 is dedicated to our optimization scheme for solving SSPCA. Our experiments in Section 4 illustrate the benefits of our approach through the denoising of sparse structured synthetic signals and an application to face recognition.

Notations. For any vector y in $\mathbb{R}^p$ and any $\alpha > 0$, we denote by $\|y\|_\alpha = (\sum_{j=1}^p |y_j|^\alpha)^{1/\alpha}$ the (quasi-)norm ℓ_α of y. Similarly, for any rectangular matrix $Y \in \mathbb{R}^{n\times p}$, we denote by $\|Y\|_{\mathrm{F}} = (\sum_{i=1}^n \sum_{j=1}^p Y_{ij}^2)^{1/2}$ its Frobenius norm, where Y_{ij} is the (i,j)-th element of Y. We write $Y^j \in \mathbb{R}^n$ for the j-th column of Y. Given w in $\mathbb{R}^p$ and a subset J of $\{1,\dots,p\}$, w_J denotes the vector in $\mathbb{R}^p$ that has the same entries w_j as w for $j \in J$, and null entries outside of J. In addition, the set $\{j \in \{1,\dots,p\}\,;\, w_j \neq 0\}$ is referred to as the *support*, or *nonzero pattern* of the vector $w \in \mathbb{R}^p$. For any finite set A with cardinality $|A|$, we also define the $|A|$-tuple $(y^a)_{a\in A} \in \mathbb{R}^{p\times|A|}$ as the collection of p-dimensional vectors y^a indexed by the elements of A. Furthermore, for two vectors x and y in $\mathbb{R}^p$, we denote by $x \circ y = (x_1y_1,\dots,x_py_p)^\top \in \mathbb{R}^p$ the elementwise product of x and y. Finally, we extend $b \mapsto \frac{a}{b}$ by continuity in zero with $\frac{a}{0} = \infty$ if $a \neq 0$ and 0 otherwise.

2 Problem Statement

It is useful to distinguish two conceptually different interpretations of PCA. In terms of *analysis*, PCA sequentially projects the data on subspaces that explain the largest fraction of the variance of the data. In terms of *synthesis*, PCA finds a basis, or orthogonal dictionary, such that all signals observed admit decompositions with low reconstruction error. These two interpretations recover the same basis of principal components for PCA but lead to different formulations for *sparse* PCA. The *analysis* interpretation leads to sequential formulations (d'Aspremont et al., 2008; Moghaddam et al., 2006; Jolliffe et al., 2003) that consider components one at a time and perform a *deflation* of the covariance matrix at each step (see Mackey, 2009). The *synthesis* interpretation leads to non-convex global formulations (Zou et al., 2006; Mairal et al., 2009; Moghaddam et al., 2006; Lee et al., 2007) which estimate simultaneously all principal components, often drop the orthogonality constraints, and are referred to as matrix factorization problems (Singh and Gordon, 2008) in machine learning, and dictionary learning in signal processing.

The approach we propose fits more naturally in the framework of dictionnary learning, whose terminology we now introduce.

2.1 Matrix Factorization and Dictionary Learning

Given a matrix $X \in \mathbb{R}^{n\times p}$ of n rows corresponding to n observations in $\mathbb{R}^p$, the dictionary learning problem is to find a matrix $V \in \mathbb{R}^{p\times r}$, called the *dictionary*, such that each observation can be well approximated by a linear combination of the r columns $(V^k)_{k\in\{1,\dots,r\}}$ of V called the *dictionary elements*. If $U \in \mathbb{R}^{n\times r}$ is the matrix of the linear combination coefficients or *decomposition coefficients*, the matrix product $UV^\top$ is called a decomposition of X.

Learning simultaneously the dictionary V and the decomposition U corresponds to a matrix factorization problem (see Witten et al., 2009, and reference therein). As formulated by Bach et al. (2008) or Witten et al. (2009), it is natural, when learning a decomposition, to penalize or constrain some norms or quasi-norms of U and V, say Ω_u and Ω_v respectively, to encode prior information — typically sparsity — about the decomposition of X. This can be written generally as

$$\begin{cases} \min\limits_{\substack{U\in\mathbb{R}^{n\times r},\\ V\in\mathbb{R}^{p\times r}}} \dfrac{1}{2np}\left\|X-UV^\top\right\|_{\mathrm{F}}^2 + \lambda\sum_{k=1}^r \Omega_v(V^k) \\ \text{s.t.} \quad \forall k,\ \Omega_u(U^k) \le 1, \end{cases} \tag{1}$$

where the regularization parameter $\lambda \geq 0$ controls to which extent the dictionary is regularized[1]. If we assume that both regularizations Ω_u and Ω_v are convex, problem (1) is convex w.r.t. U for fixed V and vice versa. It is however not *jointly* convex in the pair (U,V).

The formulation of sparse PCA considered by Lee et al. (2007) corresponds to a particular instance of this problem, where the dictionary elements are required to be sparse (without the orthogonality constraint $V^\top V = I$). This can be achieved by penalizing the columns of V by a sparsity-inducing norm, such as the ℓ_1 norm: $\Omega_v(V^k) = \left\|V^k\right\|_1$. In the next section we consider a regularization Ω_v which controls not only the sparsity but also the structure of the supports of dictionary elements.

2.2 Structured Sparsity-Inducing Norms

The work of Jenatton et al. (2009) considered a norm which induces structured sparsity in the following sense: the solutions to a learning problem regularized by this norm have a sparse support which moreover belongs to a certain set of groups of variables. Interesting sets of possible supports include sets of variables forming rectangles when arranged on a grid and more generally convex subsets[2].

[1] From Bach et al. (2008), we know that our formulation is also equivalent to two other ones, penalized respectively by $\frac{\lambda}{2}\sum_{k=1}^r [\Omega_v(V^k)]^2 + [\Omega_u(U^k)]^2$ and $\lambda\sum_{k=1}^r \Omega_v(V^k)\Omega_u(U^k)$.

[2] Although we use the term *convex* informally here, it can however be made precise with the notion of convex subgraphs (Chung, 1997).

The framework of Jenatton et al. (2009) can be summarized as follows: if we denote by $\mathcal{G}$ a subset of the power set of $\{1, \dots, p\}$, such that $\bigcup_{G\in\mathcal{G}} G = \{1, \dots, p\}$, we define the mixed ℓ_1/ℓ_2 norm Ω on a vector $y \in \mathbb{R}^p$ as

$$\Omega(y) = \sum_{G\in\mathcal{G}} \Big\{ \sum_{j\in G} (d_j^G)^2 |y_j|^2 \Big\}^{\frac{1}{2}} = \sum_{G\in\mathcal{G}} \|d^G \circ y\|_2 \,,$$

where $(d^G)_{G\in\mathcal{G}} \in \mathbb{R}^{p\times|\mathcal{G}|}$ is a $|\mathcal{G}|$-tuple of p-dimensional vectors such that $d_j^G > 0$ if $j \in G$ and $d_j^G = 0$ otherwise. This norm Ω linearly combines the ℓ_2 norms of possibly overlapping groups of variables, with variables in each group being weighted by $(d^G)_{G\in\mathcal{G}}$. Note that a same variable y_j belonging to two different groups $G_1, G_2 \in \mathcal{G}$ is allowed to be weighted differently in G_1 and G_2 (by respectively $d_j^{G_1}$ and $d_j^{G_2}$).

For specific choices of $\mathcal{G}$, Ω leads to standard sparsity-inducing norms. For example, when $\mathcal{G}$ is the set of all singletons, Ω is the usual ℓ_1 norm (assuming that all the weights are equal to 1).

We focus on the case of a 2-dimensional grid where the set of groups $\mathcal{G}$ is the set of all horizontal and vertical half-spaces (see Fig. 1 taken from Jenatton et al., 2009). As proved by Jenatton et al. (2009, Theorem 3.1), the ℓ_1/ℓ_2 norm Ω sets to zero some groups of variables $\|d^G \circ y\|_2$, i.e., some entire horizontal and vertical half-spaces of the grid, and therefore induces rectangular nonzero patterns. Note that a larger set of convex patterns can be obtained by adding in $\mathcal{G}$ half-planes with other orientations. In practice, we use planes with angles that are multiples of $\frac{\pi}{4}$, which enables the nonzero patterns to have polygonal shapes with up to 8 faces.

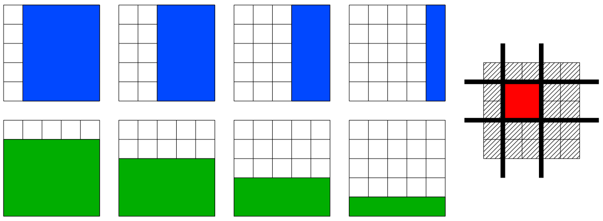

Figure 1: (Left) The set of blue and green groups with their (not displayed) complements to penalize to select rectangles. (Right) In red, an example of recovered pattern in this setting.

Among sparsity inducing regularizations, the ℓ_1 norm is often privileged since it is convex. However, so-called concave penalizations, such as penalization by an ℓ_α quasi-norm, which are closer to the ℓ_0 quasi-norm and penalize more aggressively small coefficients can be preferred, especially in a context where the unregularized problem, here dictionary learning is itself non convex. In light of recent work showing the advantages of addressing sparse problems through concave penalization (e.g., see Zou and Li, 2008), we therefore generalize Ω to a family of non-convex regularizers as follows: for $\alpha\in(0,1)$, we define the quasi-norm Ω^α for all vectors $y\in\mathbb{R}^p$ as

$$\Omega^\alpha(y) = \Big\{ \sum_{G\in\mathcal{G}} \|d^G \circ y\|_2^\alpha \Big\}^{\frac{1}{\alpha}} = \|\, (\|d^G \circ y\|_2)_{G\in\mathcal{G}} \,\|_\alpha \,,$$

where we denote by $(\|d^G \circ y\|_2)_{G\in\mathcal{G}} \in \mathbb{R}^{1\times|\mathcal{G}|}$ the $|\mathcal{G}|$-tuple composed of the different blocks $\|d^G \circ y\|_2$. We thus replace the (convex) ℓ_1/ℓ_2 norm Ω by the (neither convex, nor concave) ℓ_α/ℓ_2 quasi-norm Ω^α. While leading to the same set of (non)zero patterns, the ℓ_α quasi-norm yields sparsity at the group level more aggressively.

3 Optimization

We consider the optimization of Eq. (1) where we use $\Omega_v = \Omega^\alpha$ to regularize the dictionary V. We discuss in Section 3.3 which norms Ω_u we can handle in this optimization framework.

3.1 Formulation as a Sequence of Convex Problems

We now consider Eq. (1) where we take Ω_v to be Ω^α, $\alpha \in (0,1)$, that is,

$$\begin{cases} \min\limits_{\substack{U\in\mathbb{R}^{n\times r},\\ V\in\mathbb{R}^{p\times r}}} \frac{1}{2np} \|X - UV^\top\|_\mathrm{F}^2 + \lambda \sum\limits_{k=1}^{r} \Omega^\alpha(V^k) \\ \text{s.t.} \qquad \forall k,\ \Omega_u(U^k) \le 1, \end{cases} \tag{2}$$

Although the minimization problem in Eq. (2) is still convex in U for V fixed, the converse is not true anymore because of Ω^α. Indeed, the formulation in V is non-differentiable and non-convex. To address this problem, we use the variational equality based on the following lemma that is related[3] to ideas from Micchelli and Pontil (2006):

Lemma 3.1. *Let $\alpha \in (0,2)$ and $\beta = \frac{\alpha}{2-\alpha}$. For any vector $y \in \mathbb{R}^p$, we have the following equality*

$$\|y\|_\alpha = \min_{z\in\mathbb{R}_+^p} \frac{1}{2} \sum_{j=1}^{p} \frac{y_j^2}{z_j} + \frac{1}{2}\|z\|_\beta,$$

and the minimum is uniquely attained for $z_j = |y_j|^{2-\alpha} \|y\|_\alpha^{\alpha-1}$, $\forall j \in \{1,\dots,p\}$.

Proof. Let $\psi : z \mapsto \sum_{j=1}^p y_j^2 z_j^{-1} + \|z\|_\beta$ be the continuously differentiable function defined on $(0,+\infty)$. We have $\lim_{\|z\|_\beta\to\infty} \psi(z) = +\infty$ and $\lim_{z_j\to 0} \psi(z) = +\infty$ if $y_j \neq 0$ (for $y_j = 0$, note that $\min_{z\in\mathbb{R}_+^p} \psi(z) = \min_{z\in\mathbb{R}_+^p, z_j=0} \psi(z)$). Thus, the infimum exists and it is attained. Taking the derivative w.r.t. z_j (for $z_j > 0$) leads to the expression of the unique minimum, expression that is still correct for $z_j = 0$. □

[3] Note that we depart from Micchelli and Pontil (2006) who consider a quadratic upperbound on the *squared* norm. We prefer to remain in the standard dictionary learning framework where the penalization is not squared.

To reformulate problem (2), let us consider the $|\mathcal{G}|$-tuple $(\eta^G)_{G\in\mathcal{G}} \in \mathbb{R}^{r\times|\mathcal{G}|}$ of r-dimensional vectors η^G that satisfy for all $k \in \{1,\dots,r\}$ and $G \in \mathcal{G}$, $\eta_k^G \geq 0$. It follows from Lemma (3.1) that $2\sum_{k=1}^r \Omega^\alpha(V^k)$ is equal to

$$\min_{(\eta^G)_{G\in\mathcal{G}}\in\mathbb{R}_+^{r\times|\mathcal{G}|}} \sum_{k=1}^{r}\Big[\|(\eta_k^G)_{G\in\mathcal{G}}\|_\beta + \sum_{G\in\mathcal{G}}\big\|V^k\circ d^G\big\|_2^2\,(\eta_k^G)^{-1}\Big],$$

that can be rewritten in turn as

$$\min_{(\eta^G)_{G\in\mathcal{G}}\in\mathbb{R}_+^{r\times|\mathcal{G}|}} \sum_{k=1}^{r}(V^k)^\top \mathrm{Diag}\big(\zeta^k\big)^{-1}V^k + \|(\eta_k^G)_{G\in\mathcal{G}}\|_\beta,$$

with $\zeta\in\mathbb{R}^{p\times r}$ defined by[4] $\zeta_{jk} = \big\{\sum_{G\in\mathcal{G},\, G\ni j}(d_j^G)^2(\eta_k^G)^{-1}\big\}^{-1}$. This leads to the following formulation

$$\min_{\substack{U,\,V,\,\Omega_u(U^k)\leq 1\\ (\eta^G)_{G\in\mathcal{G}}\in\mathbb{R}_+^{r\times|\mathcal{G}|}}} \frac{1}{2np}\big\|X-UV^\top\big\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\sum_{k=1}^{r}\Big[(V^k)^\top \mathrm{Diag}\big(\zeta^k\big)^{-1}V^k + \|(\eta_k^G)_{G\in\mathcal{G}}\|_\beta\Big], \quad (3)$$

that is equivalent to Eq. (2) and quadratic with respect to V.

3.2 Sharing Structure among Dictionary Elements

So far, the regularization quasi-norm Ω^α has been used to induce a structure *inside* each dictionary element taken separately. Nonetheless, some applications may also benefit from a control of the structure *across* dictionary elements. For instance it can be desirable to impose the constraint that several dictionary elements share the exact same nonzero patterns. In the context of face recognition, this could be relevant to model the variability of faces as the combined variability of several parts, with each part having a small support (such as eyes), and having its variance itself explained by *several* dictionary elements (corresponding for example to the color and the shape of the eyes).

To this end, we consider $\mathcal{M}$, a partition of $\{1,\dots,r\}$. Imposing that two dictionary elements V^k and $V^{k'}$ share the same sparsity pattern is equivalent to imposing that V_i^k and $V_i^{k'}$ are simultaneously zero or non-zero. Following the approach used for joint feature selection (Obozinski et al., 2009) where the ℓ_1 norm is composed with an ℓ_2 norm, we compose the norm Ω^α with the ℓ_2 norm $V_i^M = \|(V_i^k)_{k\in M}\|_2$, of all i^{th} entries of each dictionary element of a class M of the partition $\mathcal{M}$, leading to the regularization:

$$\sum_{M\in\mathcal{M}}\Omega^\alpha(V_i^M) = \sum_{M\in\mathcal{M}}\Big[\sum_{G\in\mathcal{G}}\big\|(V_i^k d_i^G)_{i\in G,\,k\in M}\big\|_2^\alpha\Big]^{\frac{1}{\alpha}}. \quad (4)$$

In fact, not surprisingly given that similar results hold for the group Lasso (Bach, 2008), it can be shown that the above extension is equivalent to the variational formulation

$$\min_{\substack{U,\,V,\,\Omega_u(U^k)\leq 1\\ (\eta^G)_{G\in\mathcal{G}}\in\mathbb{R}_+^{|\mathcal{M}|\times|\mathcal{G}|}}} \frac{1}{2np}\big\|X-UV^\top\big\|_{\mathrm{F}}^2 + \frac{\lambda}{2}\sum_{M\in\mathcal{M}}\Big[\sum_{k\in M}(V^k)^\top \mathrm{Diag}\big(\zeta^M\big)^{-1}V^k + \|(\eta_M^G)_{G\in\mathcal{G}}\|_\beta\Big],$$

with class specific variables η_M, ζ^M, $M\in\mathcal{M}$, defined in a similar way to η_k and ζ^k, $k\in\{1,\dots,r\}$.

3.3 Algorithm

The main optimization procedure described in Algorithm 1 is based on a cyclic optimization over the three variables involved, namely $(\eta^G)_{G\in\mathcal{G}}$, U and V. We use Lemma (3.1) to solve (2) through a sequence of problems that are convex in U for fixed V (and conversely, convex in V for fixed U). For this sequence of problems, we then present efficient optimization procedures based on block coordinate descent (BCD) (Bertsekas, 1995, Section 2.7). We describe these in detail in Algorithm 1. Note that we depart from the approach of Jenatton et al. (2009) who use an active set algorithm. Their approach does not indeed allow warm restarts, which is crucial in our alternating optimization scheme.

Update of $(\eta^G)_{G\in\mathcal{G}}$. The update of $(\eta^G)_{G\in\mathcal{G}}$ is straightforward (even if the underlying minimization problem is non-convex), since the minimizer $(\eta^G)^*$ in Lemma (3.1) is given in closed-form. In practice, following Micchelli and Pontil (2006), we avoid numerical instability near zero with the smoothed update $\eta_k^G \leftarrow \max\{(\eta_k^G)^*, \varepsilon\}$, with $\varepsilon \ll 1$.

Update of U. The update of U follows the technique suggested by Mairal et al. (2009). Each column U^k of U is constrained separately through $\Omega_u(U^k)$. Furthermore, if we assume that V and $\{U^j\}_{j\neq k}$ are fixed, some basic algebra leads to

$$\begin{aligned} &\operatorname*{arg\,min}_{\Omega_u(U^k)\leq 1} \frac{1}{2np}\big\|X-UV^\top\big\|_{\mathrm{F}}^2 \\ =\; &\operatorname*{arg\,min}_{\Omega_u(U^k)\leq 1} \big\|U^k - \big\|V^k\big\|_2^{-2}\big(X-\sum_{j\neq k}[U^j]^\top V^j\big)V^k\big\|_2^2 \\ =\; &\operatorname*{arg\,min}_{\Omega_u(U^k)\leq 1} \big\|U^k - w\big\|_2^2, \end{aligned} \quad (5)$$

which is simply the Euclidean projection $\Pi_{\Omega_u}(w)$ of w onto the unit ball of Ω_u. Consequently, the cost of the BCD update of U depends on how fast we can perform this projection; the ℓ_1 and ℓ_2 norms are typical cases where the projection can be computed efficiently. In the experiments, we take Ω_u to be the ℓ_2 norm.

In addition, since the function $U^k \mapsto \frac{1}{2np}\big\|X-UV^\top\big\|_{\mathrm{F}}^2$ is continuously differentiable on the (closed convex) unit ball of Ω_u, the convergence of the BCD procedure is guaranteed

[4] For the sake of clarity, we do not specify the dependence of ζ on $(\eta^G)_{G\in\mathcal{G}}$.

since the minimum in Eq. (5) is unique (Bertsekas, 1995, Proposition 2.7.1). The complete update of U is given in Algorithm 1.

Update of V. A fairly natural way to update V would be to compute the closed form solutions available for each row of V. Indeed, both the loss $\frac{1}{2np}\left\|X-UV^\top\right\|_{\mathrm{F}}^2$ and the penalization on V are separable in the rows of V, leading to p independent ridge-regression problems, implying in turn p matrix inversions.

However, in light of the update of U, we consider again a BCD scheme on the columns of V that turns out to be much more efficient, without requiring any non-diagonal matrix inversion. The detailed procedure is given in Algorithm 1. The convergence follows along the same arguments as those used for U.

Algorithm 1 Main procedure for solving Eq. (3).

```
Input: Dictionary size $r$, data matrix $X$.
Initialization: Initialization of $U, V$ (possibly random).
while ( stopping criterion not reached )
    Update $(\eta^G)_{G\in\mathcal{G}}$: closed-form solution.
    Update $U$ by BCD:
    for $t = 1$ to $T_u$, for $k = 1$ to $r$:
        $U^k \leftarrow \Pi_{\Omega_u}(U^k + \left\|V^k\right\|_2^{-2}(XV^k - UV^\top V^k))$.
    Update $V$ by BCD:
    for $t = 1$ to $T_v$, for $k = 1$ to $r$:
        $V^k \leftarrow \mathrm{Diag}(\zeta^k)\,\mathrm{Diag}\Big(\left\|U^k\right\|_2^2 \zeta^k + np\lambda\mathbf{1}\Big)^{-1}(X^\top U^k$
                    $- VU^\top U^k + \left\|U^k\right\|_2^2 V^k)$.
Output: Decomposition $U, V$.
```

Our problem is not *jointly* convex in $(\eta^G)_{G\in\mathcal{G}}$, U and V, which raises the question of the sensitivity of the optimization to its initialization. This point will be discussed in Section 4. In practice, the stopping criterion relies on the relative decrease (typically 10^{-3}) in the cost function in Eq. (2).

Algorithmic complexity. The complexity of Algorithm 1 can be decomposed into 3 terms, corresponding to the update procedures of $(\eta^G)_{G\in\mathcal{G}}$, U and V. We denote by T_u (respectively T_v) the number of updates of U (respectively V) in Algorithm 1. First, computing $(\eta^G)_{G\in\mathcal{G}}$ and ζ costs $O(r|\mathcal{G}| + (|\mathcal{G}| + r)\sum_{G\in\mathcal{G}}|G|) = O(pr|\mathcal{G}| + p|\mathcal{G}|^2)$. The update of U requires $O((p+T_u n)r^2 + (np+C_\Pi T_u)r)$ operations, where C_Π is the cost of projecting onto the unit ball of Ω_u. Similarly, we get for the update of V a complexity of $O((n+T_v p)r^2 + npr)$. In practice, we notice that the BCD updates for both U and V require only few steps, so that we choose $T_u = T_v = 5$. In our experiments, the algorithmic complexity simplifies to $O(p^2 + r^2\max\{n,p\} + rp\max\{p^{1/2}, n\})$ times the number of iterations in Algorithm 1. Note that the complexity is linear in n and is quadratic in r, which is empirically the computational bottleneck.

Extension to NMF. Our formalism does not cover the positivity constraints of non-negative matrix factorization, but it is straightforward to extend it at the cost of an additional cheap threshold operation (to project onto the positive orthant) in the BCD updates of U and V.

4 Experiments

We first consider the denoising of synthetic signals to illustrate the effect of our regularization. We then focus on the application of SSPCA to a face recognition problem and we show that, by adding a sparse structured prior instead of a simple sparse prior, we gain in robustness to occlusions. In preliminary experiments, we considered the exact regularization from Jenatton et al. (2009), i.e., with $\alpha = 1$, but found that the obtained patterns were not sufficiently sparse and salient. We therefore turned to the setting where the parameter α is in $(0,1)$. We chose $\alpha = 0.5$, since much smaller or larger values yield either not sparse enough solutions or numerical instability.

By definition, dictionary learning belongs to unsupervised learning; in that sense, our method may appear first as a tool for exploratory data analysis, which leads us naturally to *qualitatively* analyze the results of our decompositions (e.g., by visualizing the learned dictionaries). This is obviously a difficult and subjective exercise, beyond the assessment of the consistency of the method in artificial examples where the "true" dictionnary is known. For that reason, we endeavor in the experiments to compare our method objectively and *quantitatively* with other techniques. Specifically, we apply our method within either a denoising or a classification setting, and assess its performance respectively by the obtained increase in explained variance or classification accuracy.

A Matlab toolbox implementing our method can be downloaded from `http://www.di.ens.fr/~jenatton/`.

4.1 Denoising of Synthetic Signals

In this first experiment, we consider signals generated by the following noisy linear model

$$u_1\mathbf{V^1} + u_2\mathbf{V^2} + u_3\mathbf{V^3} + \varepsilon \in \mathbb{R}^{400}, \tag{6}$$

where $\mathbf{V} = [\mathbf{V^1}, \mathbf{V^2}, \mathbf{V^3}] \in \mathbb{R}^{400\times 3}$ are sparse and structured dictionary elements organized on a 20×20-dimensional grid ($\mathbf{V}$ is represented on the top row of Fig. 2). The components of the noise vector ε are independent and identically distributed according to a centered Gaussian distribution with its variance set to obtain a signal-to-noise ratio (SNR) of 0.5. The coefficients $[u_1, u_2, u_3]$ that linearly combine the dictionary elements of $\mathbf{V}$ are generated according to a centered Gaussian distribution, with the following covariance matrix

$$\begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{bmatrix}.$$

From 250 of such signals, we learn a decomposition $\hat{U}\hat{V}^\top$ with $r = 3$ dictionary elements, which seems a reasonable choice of r in an attempt to recover the underlying (in this case, known) structure of $\mathbf{V}$. For SPCA and SSPCA, the regularization parameter λ is selected by 5-fold cross-validation on the reconstruction error. Based on the learned dictionary $\hat{V}$, we denoise 1000 new signals generated in the same way. We report in Table 1 the results of the denoising, for PCA, SPCA and SSPCA.

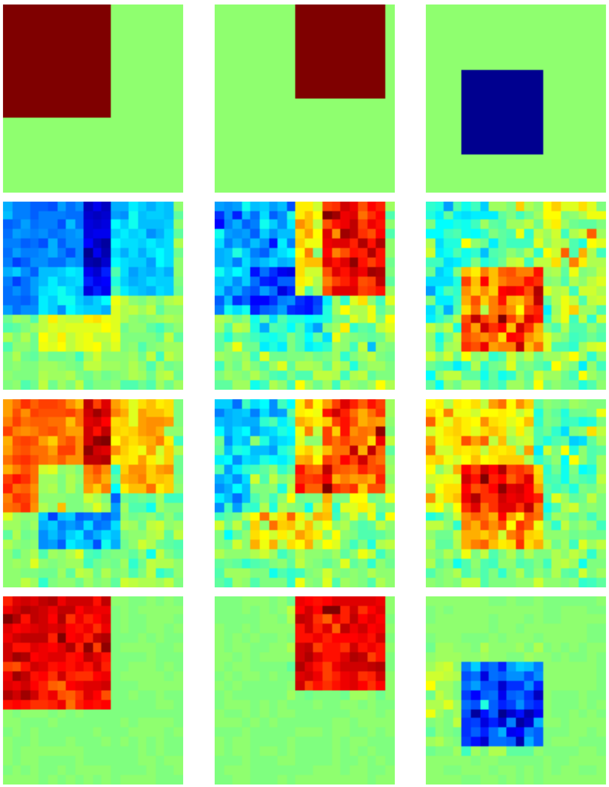

Figure 2: Top row: dictionary $\mathbf{V}=[\mathbf{V^1},\mathbf{V^2},\mathbf{V^3}]\in\mathbb{R}^{400\times3}$ used to generate the signals Eq. (6). From the second to the bottom row: dictionary elements recovered from 250 signals by PCA, SPCA and SSPCA (best seen in color).

The difficulty of this task is essentially twofold and lies in (1) the high level of noise and in (2) the small number of signals (i.e., 250 signals against 400 variables) available to learn the decomposition.

As displayed on Fig. 2, PCA and SPCA learn very scattered and uninterpretable dictionary elements. On the other hand, the sparse structured prior we put through Ω^α helps to recover the initial structure of $\mathbf{V}$, which, in turn, improves upon the denoising performance of SSPCA (see Table 1). Note that in order to assess the statistical significance of the differences between the average denoising performances of Table 1, one has to consider the sample standard deviation *divided* by $\sqrt{1000}$ (Lehmann and Romano, 2005), i.e., roughly ≈ 0.007.

The setting we consider here raises the interesting question of *model identifiability*, i.e., whether we can recover the true dictionary elements that generated the signals, which we defer to future work.

	PCA	SPCA	SSPCA
Estimation error:	0.41±0.22	0.40±0.22	0.34±0.21

Table 1: Average and standard deviation of the normalized estimation error, computed over 1000 signals for PCA, SPCA and SSPCA.

4.2 Face Recognition

We apply SSPCA on the cropped AR Face Database (Martinez and Kak, 2001) that consists of 2600 face images, corresponding to 100 individuals (50 women and 50 men). For each subject, there are 14 non-occluded poses and 12 occluded ones (the occlusions are due to sunglasses and scarfs). We reduce the resolution of the images from 165×120 pixels to 38×27 pixels for computational reasons.

Fig. 3 shows examples of learned dictionaries (for $r = 36$ elements), for NMF, SSPCA and SSPCA with shared structure (see Section 3.2). While NMF finds sparse but spatially unconstrained patterns, SSPCA select sparse convex areas that correspond to a more natural segment of faces. For instance, meaningful parts such as the mouth and the eyes are recovered by the dictionary.

We now quantitatively compare SSPCA, SPCA, PCA and NMF on a face recognition problem. We first split the data into 2 parts, the occluded faces and non-occluded ones. For different sizes of the dictionary, we apply each of the aforementioned dimensionality reduction techniques to the non-occluded faces. Keeping the learned dictionary V, we decompose both non-occluded and occluded faces on V. We then classify the occluded faces with a k-nearest-neighbors classifier (k-NN), based on the obtained low-dimensional representations U. Given the size of the dictionary, we choose the number of neighbor(s) and the amount of regularization λ by cross-validation[5] on the non-occluded faces.

The formulations of NMF, SPCA and SSPCA are nonconvex and as a consequence, the local minima reached by those methods might a priori be sensitive to the initialization. To evaluate this sensitivity, we repeat the protocol described above 10 times and display in Fig. 4 the median, first and third quartile of the classification scores obtained in this way. In practice we found the performance on the test set to be pretty stable as a function of the initialization. We denote by shared-SSPCA (resp. shared-SPCA) the models where we impose, on top of the structure of Ω^α, to have only 10 different nonzero patterns among the learned dictionaries (see Section 3.2).

[5] We perform 5-fold cross-validation and the number of nearest neighbor(s) is searched in $\{1, 3, 5\}$ while $\log_{10}(\lambda)$ is in $\{-11, -10.5, \ldots, -7\}$. For the dictionary, we consider the sizes $r \in \{10, 20, \ldots, 150\}$.

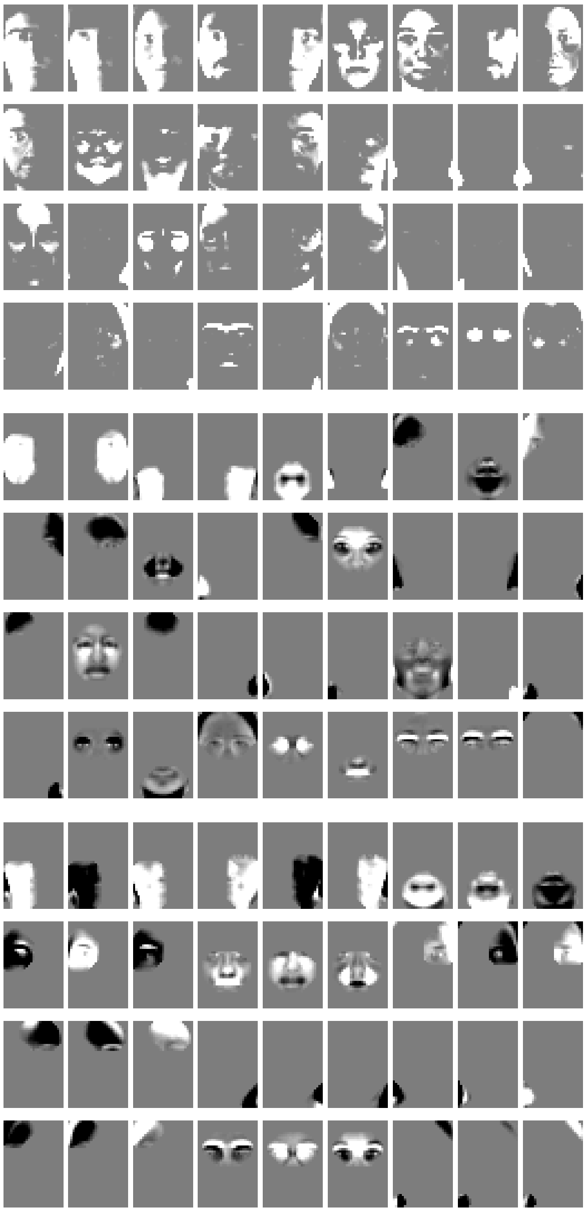

Figure 3: Three learned dictionaries of faces with $r = 36$: NMF (top), SSPCA (middle) and shared-SSPCA (bottom) (i.e., SSPCA with $|\mathcal{M}| = 12$ different patterns of size 3). The dictionary elements are sorted in decreasing order of explained variance. While NMF gives sparse spatially unconstrained patterns, SSPCA finds convex areas that correspond to more natural face segments. SSPCA captures the left/right illuminations and retrieves pairs of symmetric patterns. Some displayed patterns do not seem to be convex, e.g., nonzero patterns located at two opposite corners of the grid. However, a closer look at these dictionary elements shows that convex shapes are indeed selected, and that small numerical values (just as regularizing by ℓ_2 norm may lead to) give the visual impression of having zeroes in convex nonzero patterns. This also shows that if a nonconvex pattern has to be selected, it will be, by considering its convex hull.

We performed a Wilcoxon signed-rank (Lehmann and Romano, 2005) between the classification scores of NMF and SSPCA, and for dictionary sizes greater than 100 (up to 150), our approach performs better than NMF at the 5% significance level. For smaller dictionaries, NMF and SSPCA perform similarly. The other methods, including PCA and SPCA, obtained overall lower scores than NMF and can also be shown to perform significantly worse than SSPCA.

As a baseline, we also plot the classification score that we obtain when we directly apply k-NN on the raw data, without preprocessing. Because of its local dictionary, SSPCA proves to be more robust to occlusions and therefore outperforms the other methods on this classification task. On the other hand, SPCA, that yields sparsity without a structured prior, performs poorly. Sharing structure across the dictionary elements (see Section 3.2) seems to help SPCA for which no structure information is otherwise available.

The goal of our paper is not to compete with state-of-the-art techniques of face recognition, but to demonstrate the improvement obtained between the ℓ_1 norm and more structured norms. We could still improve upon our results using non-linear classification (e.g., with a SVM) or by refining our features (e.g., with a Laplacian filter).

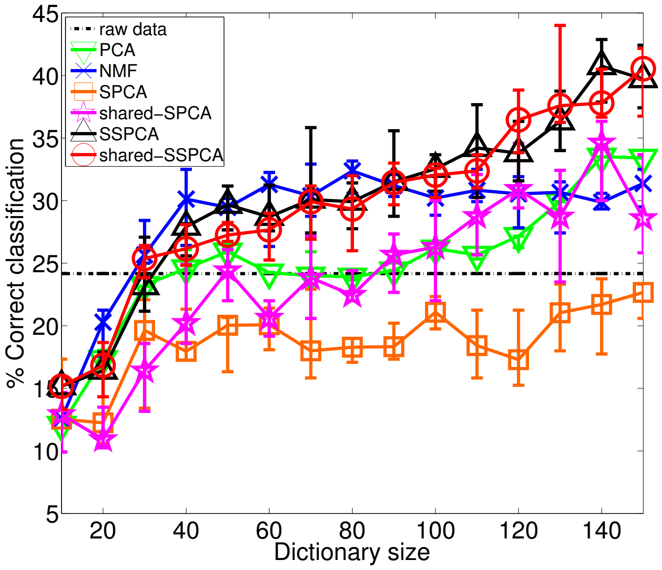

Figure 4: Classification accuracy versus dictionary size: each dimensionality reduction technique is used with k-NN to classify occluded faces. SSPCA shows better robustness to occlusions. The points, lower and upper error bars on the curves respectively represent the median, first and third quartile, based on 10 runs.

5 Conclusions

We proposed to apply a non-convex variant of the regularization introduced by Jenatton et al. (2009) to the problem of structured sparse dictionary learning. We present an efficient block-coordinate descent algorithm with closed-form updates. In a denoising task of sparse structured signals, our approach led to better performance and to a more interpretable decomposition of the data. For face recognition, the dictionaries learned have increased robustness to occlusions compared to NMF.

In future work, we would like to investigate Bayesian frameworks that would define similar structured priors and allow the principled choice of the regularization parameter and the number of dictionary elements (Zhou et al., 2009). Moreover, although we focus in this work on controlling the structure of the dictionary V, we could instead impose structure on the decompostion coefficients U and study the induced effect on the dictionary V (Kavukcuoglu et al., 2009). This could be straightforward ti do with the same formulation, by transposing the data matrix X. Finally, we intend to apply this structured sparsity-inducing regularization for multi-task learning, in order to take advantage of the structure between tasks.

Acknowledgments

This paper was supported in part by a grant from the Agence Nationale de la Recherche (MGA Project) and a grant from the European Research Council (SIERRA Project). We thank Julien Mairal for sharing his insights on dictionary learning.

References

F. Bach. Consistency of the group Lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008.

F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. Technical report, arXiv:0812.1869, 2008.

R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. Technical report, 2008. Submitted to IEEE T. Inform. Theory.

D. P. Bertsekas. *Nonlinear programming*. Athena scientific, 1995.

F.R.K. Chung. *Spectral graph theory*. American Mathematical Society, 1997.

A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *J. Mach. Learn. Res.*, 9:1269–1294, 2008.

L. He and L. Carin. Exploiting structure in wavelet-based Bayesian compressive sensing. *IEEE T. Signal Proces.*, 57:3488–3497, 2009.

J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *Proc. ICML*, 2009.

L. Jacob, G. Obozinski, and J. P. Vert. Group Lasso with overlap and graph Lasso. In *Proc. ICML*, 2009.

R. Jenatton, J-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, arXiv:0904.3523, 2009.

I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the Lasso. *J. Comput. Graph. Stat.*, 12(3):531–547, 2003.

K. Kavukcuoglu, M. A. Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. In *Proc. CVPR*, 2009.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Adv. NIPS*, 2007.

E. L. Lehmann and J. P. Romano. *Testing statistical hypotheses*. Springer Verlag, 2005.

L. Mackey. Deflation methods for sparse PCA. In *Adv. NIPS*, 2009.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proc. ICML*, 2009.

A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE T. Pattern. Anal.*, 23(2):228–233, 2001.

C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *J. Mach. Learn. Res.*, 6(2):1099, 2006.

B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact and greedy algorithms. In *Adv. NIPS*, 2006.

G. Obozinski, B. Taskar, and M. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Stat. Comput.*, 2009.

A. P. Singh and G. J. Gordon. A unified view of matrix factorization models. In *Proc. ECML*, 2008.

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, 2009.

R. Zass and A. Shashua. Nonnegative sparse PCA. In *Adv. NIPS*, 2007.

M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin. Non-parametric bayesian dictionary learning for sparse image representations. In *Adv. NIPS*, 2009.

H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Ann. Stat.*, 36(4):1509–1533, 2008.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J. Comput. Graph. Stat.*, 15(2):265–286, 2006.