Designing Structured Sparse Dictionaries for Sparse Representation Modeling

G. Tessitore and R. Prevete

Abstract. Linear approaches to the problem of unsupervised data dimensionality reduction consist in finding a suitable *set of factors*, which is usually called *dictionary*, on the basis of which data can be represented as a linear combination of the dictionary elements. In recent years there have been relevant efforts for searching data representation which are based on *sparse* dictionary elements or a sparse linear combination of the dictionary elements. Here we investigate the possibility to combine the advantages of both sparse dictionary elements and sparse linear combination. Notably, we also impose a *structure* on the dictionary elements. We compare our algorithm with two other different approaches presented in literature which impose either sparse structured dictionary elements or sparse linear combination. These (preliminary) results suggests that our approach presents some promising advantages, in particular a greater possibility of interpreting the data representation.

1 Introduction

Unsupervised dimensionality reduction seems to bring relevant benefits in many contexts which include classification, regression and control problems [12, 13]. Among linear techniques for dimensional reduction Principal Component Analysis (PCA) [7] is widely used. The PCA approach enables one to approximate signals as a linear combination of a restricted number of orthogonal *factors* that most efficiently explain the data variance. Accordingly a signal composed of a large number of variables can be represented as a small number of coefficients of the linear combination of the factors. The orthogonal factors typically involve all original variables. This aspect can cause some difficulties, notably for the interpretation of the signals expressed as linear combination of the orthogonal factors. For example, in the context of the research on human actions [12] can be useful to determine which "pieces" of the action are more relevant than others in classifying or controlling the action itself.

G. Tessitore · R. Prevete
Department of Physical Sciences, University of Naples Federico II, Naples, Italy
e-mail: {prevete,tessitore}@na.infn.it

R. Burduk et al. (Eds.): Computer Recognition Systems 4, AISC 95, pp. 157–166.
springerlink.com

In recent years, there has been a growing interest for identifying alternatives to PCA which find interpretable and more powerful representations of the signals. In these approaches the factors or the coefficients of the linear combination are obtained using prior information represented by penalization terms or constraints in a minimization problem. One can isolate two different types of approaches: a) *Sparse factors*. In this case each factor involves just a small number of the original variables (for example see Sparse-PCA [14] and Structured-Sparse-PCA [9]). b) *Sparse linear combination of the factors*. Here an *overcomplete* set of factors is chosen in advance or learned from the data, but the approximation of each signal involves only a restricted number of factors. Hence, the signals are represented by sparse linear combinations of the factors (for example see K-SVD [1] and MOD [6]). In both approaches, the orthogonality constraint on the factors is usually violated, the set of factors is called *dictionary*, and each factor is called *dictionary element* or *atom*.

Here we investigate the possibility of combining the advantages of both approaches simultaneously finding sparse dictionary elements and sparse linear combination of the dictionary elements. Importantly, in addition to sparse dictionary elements we also impose a *structure* on the atoms [9]. To this aim, we propose a method for *Sparse Representation with Structured Sparse Dictionary (SR-SSD)*. Our approach is tested by using both synthetic and real datasets. The rest of the paper is organized as follows: In Section 2, we give the theoretical formulation of the problem. The description of our algorithm and its relation to previously proposed method is presented in Section 3. The results of our algorithm on synthetic and real data, compared with the results of other two standard algorithms, are presented in Section 4. Finally, in Section 5 we discuss the results obtained and draw our main conclusions.

Notations: Bold uppercase letters refer to matrices, e.g., $\mathbf{X}$,$\mathbf{V}$, and bold lowercase letters designate vectors, e.g., $\mathbf{x}$,$\mathbf{v}$. We denote by $\mathbf{X}_i$ and $\mathbf{X}^j$ the i-th row and the j-th column of a matrix $\mathbf{X}$, respectively. While we use the notation x_i and v_{ij} to refer to the i-th element of the vector $\mathbf{x}$ and the element in the i-th row and the j-th column of the matrix $\mathbf{V}$, respectively. Given $\mathbf{x} \in \mathbb{R}^p$ and $q \in [1,\infty)$, we use the notation $\|x\|_q$ to refer to its l_q-norm defined as $\|x\|_q = (\sum_{i=1}^{p} |x_i|^q)^{1/q}$. Given two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^p$, we denote by $\mathbf{x} \circ \mathbf{y} = (x_1 y_1, x_2 y_2, ..., x_p y_p)^T \in R^p$ the element-wise product of $\mathbf{x}$ and $\mathbf{y}$.

2 Problem Statement

Let us define a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ of n rows in $\mathbb{R}^p$, each one corresponding to an experimental observation, the learning problem we are addressing can be solved by finding a matrix $\mathbf{V} \in \mathbb{R}^{p \times r}$ such that each row of $\mathbf{X}$ can be approximated by a linear combination of the r columns of $\mathbf{V}$. $\mathbf{V}$ is the *dictionary*, and the r columns $\mathbf{V}^k$ of $\mathbf{V}$ are the *dictionary elements* or *atoms*. Let us call $\mathbf{U} \in \mathbb{R}^{n \times r}$ the matrix of the linear combination coefficients, i.e., the i-th row of $\mathbf{U}$ corresponds to the r coefficients of the linear combination of the r columns of $\mathbf{V}$ in order to approximate the i-th row of $\mathbf{X}$. Consequently, $\mathbf{U}\mathbf{V}^T$ is an approximation of $\mathbf{X}$. The learning problem can be expressed as:

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2np}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 + \lambda \sum_{k=1}^{r} \Omega_v(\mathbf{V}^k) \; s.t. \; \forall j, \Omega_u(\mathbf{U}_j) < T_0 \tag{1}$$

where $\Omega_v(\mathbf{V}^k)$ and $\Omega_u(\mathbf{U}_j)$ are some norms or quasi-norms that constrain or regularize the solutions of the minimization problem, and $\lambda \geq 0$ and $T_0 > 0$ are parameters that controls to which extension the dictionary and the coefficients are regularized. If one assumes that both regularizations Ω_u and Ω_v are convex, for $\mathbf{V}$ fixed the problem (1) is convex w.r.t. $\mathbf{U}$ and vice versa.

As we want to induce a sparsity for the linear combination of the dictionary elements, we can choose either ℓ_0 or ℓ_1 norm for the coefficients. These norms penalize linear combinations containing many coefficients different from zero. If not specified differently, we make the following choice: $\Omega_u(\mathbf{U}_h) = \|\mathbf{U}_h\|_0$.

Following [9] the structured sparsity of the dictionary elements can be imposed by choosing

$$\Omega_v(\mathbf{V}^k) = \left\{ \sum_{i=1}^{s} \| \mathbf{d}^i \circ \mathbf{V}^k \|_2^{\alpha} \right\}^{\frac{1}{\alpha}} \tag{2}$$

where $\alpha \in (0,1)$, and each $\mathbf{d}^i$ is a p-dimensional vector satisfying the condition $d^i_j \geq 0$, with $i = 1,2,...,s$. The s vectors $\mathbf{d}^i$ allow to define the structure of the dictionary elements. More specifically each $\mathbf{d}^i$ individuates a group of variables corresponding to the set $G^i = \left\{ j \in \{1,\ldots,p\} : d^i_j = 0 \right\}$. The norm $\| \mathbf{d}^i \circ \mathbf{V}^k \|_2^{\alpha}$ penalizes the variables for which $d^i_j > 0$ and therefore induces non zero values for variables v_{jk} with $j \in G^i$. The resulting set of selected variables depends on the contribution of each $\mathbf{d}^i$ as described in [8]. For example, if the vectors $\mathbf{d}^i$ induce a partition on the set $\{1,\ldots,p\}$, then the penalization term (2) favours the formation of dictionary elements $\mathbf{V}^k$ composed of non-zero variables belonging to just one part of the partition. From [9], who in turn follows [10], the problem (1) considering (2) can be reformulated as follows:

$$\min_{U,V,H} \frac{1}{2np}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 + \frac{\lambda}{2} \sum_{k=1}^{r} \left[(\mathbf{V}^k)^T Diag(\mathbf{Z}^k)^{-1} \mathbf{V}^k) + \|\mathbf{H}_k\|_{\beta} \right] \quad s.t. \; \forall j, \|\boldsymbol{U}_j\|_0 \leq T_0 \tag{3}$$

where $\mathbf{H} \in \mathbb{R}_+^{r \times s}$ is a matrix satisfying the condition $h_{ki} \geq 0$, and $\beta = \frac{\alpha}{2-\alpha}$. The matrix $\mathbf{Z} \in \mathbb{R}^{p \times r}$ is defined as $z_{jk} = \left\{ \sum_{i=1}^{s} \left(d^i_j \right)^2 (h_{ki})^{-1} \right\}^{-1}$. Notice that minimizer of 3 for fixed both $\mathbf{U}$ and $\mathbf{V}$ is given in a closed form, and it is equal to $h_{ki} = \bar{h}_{ki} = |y^k_i|^{2-\alpha} \|\mathbf{y}^k\|_{\alpha}^{\alpha-1}$, for $k = 1,2,...,r$ and $i = 1,2,...,s$, where each $\mathbf{y}^k \in \mathbb{R}^{1 \times s}$ is the vector $\mathbf{y}^k = (\|\mathbf{d}^1 \circ \mathbf{V}^k\|_2, \|\mathbf{d}^2 \circ \mathbf{V}^k\|_2, \ldots, \|\mathbf{d}^s \circ \mathbf{V}^k\|_2)$.

In order to solve the problem (3), we follow the usual approach of finding the minimum by alternating optimizations with respect to the values $\mathbf{H}$, to the coefficients $\mathbf{U}$ and to the dictionary $\mathbf{V}$. Most methods are based on this alternating scheme of optimization [2].

3 *SR-SSD* Algorithm

SR-SSD algorithm proposed here is composed of three alternate stages: *Update of* the matrix $\mathbf{H}$ *, Sparse Coding* Stage, and *Structured Dictionary Stage*. Notice that the problem 3 is convex in $\mathbf{U}$ for fixed $\mathbf{V}$ and vice versa.

***Update of matrix* H.** In this stage, we assume that both $\mathbf{U}$ and $\mathbf{V}$ are fixed and update the $\mathbf{H}$'s values. As said above, one can update $\mathbf{H}$ by a straightforward equation $h_{ki} = \bar{h}_{ki} = |y_i^k|^{2-\alpha}\|\mathbf{y}^k\|_\alpha^{\alpha-1}$, however in order to avoid numerical instability near zero a smoothed update is used as follows: $\mathbf{H}_k \leftarrow max\{\bar{\mathbf{H}}_k, \varepsilon\}$ with $\varepsilon \ll 1$.

Sparse Coding Stage. The second stage of the algorithm proposed here consists in updating the $\mathbf{U}$'s values in such a way to obtain a sparse representation of the signals $\mathbf{X}$, for fixed both $\mathbf{V}$ and $\mathbf{H}$. Note that the equation (3) is composed of two terms to be minimized, and the second term does not depend on $\mathbf{U}$. Therefore, the optimization problem posed in (3) can be, in this stage, reformulated as follows: $\min_{\mathbf{U}} \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$ *s.t.* $\forall j, \|\mathbf{U}_j\|_0 \leq T_0$. There are a number of well-known "pursuit algorithms" which finds an approximate solution for this type of problem (see for example Basis Pursuit (BP) [4] and Ortoghonal Matching Pursuit (OMP) [11]). In our approach we use OMP in experiments in Section 4.1 whereas Iterative Soft-Thresholding (IST) algorithm [3] for experiments described in Section 4.2 where we replaced the ℓ_0 with the ℓ_1 norm.

Structured Dictionary Element Stage. The update of the dictionary $\mathbf{V}$ is performed in this stage, and, more importantly, following the approach suggested by [9] a structured sparse representation for the atoms is found. Fixed both $\mathbf{U}$ and $\mathbf{H}$ the problem (3) can be reformulated as follows

$$\min_V \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \frac{\lambda np}{2}\sum_{k=1}^{r}(\mathbf{V}^k)^T Diag(\mathbf{Z}^k)^{-1}\mathbf{V}^k) \tag{4}$$

Although in this case both the two terms of the problem 4 are convex and differentiable with respect to $\mathbf{V}$, for fixed both $\mathbf{U}$ and $\mathbf{H}$, leading to a closed form solution for each row of $\mathbf{V}$, in order to avoid p matrix inversions, we consider a proximal method to update $\mathbf{V}$:

$$\mathbf{V}^k \leftarrow Diag(\mathbf{Z}^k)Diag(\|\mathbf{U}^k\|_2^2\mathbf{Z}^k + np\lambda\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{U}^k - \mathbf{V}\mathbf{U}^T\mathbf{U}^k + \|\mathbf{U}^k\|_2^2\mathbf{V}^k) \tag{5}$$

where the update rule is obtained by composing a forward gradient descent step on the first term with the proximity operator of the second term of (4).

A description of our method is given in Algorithm 1.

4 Experiments

Two different kinds of experiments were conducted: the first series of experiments is aimed at testing the ability of the proposed method in retrieving the original dictionary from synthetic data. In the second kind of experiments, we test the ability

Algorithm 1 SR-SSD Algorithm

Input: X, **Output**: $\mathbf{U}, \mathbf{V}$
while stop-criterion is not reached
- update $\mathbf{H}$: closed form solution given by $\mathbf{H}_k \leftarrow max\{\tilde{\mathbf{H}}_k, \varepsilon\}$ with $\varepsilon \ll 1$.
- sparse coding stage: use OMP or IST algorithm to update $\mathbf{U}$
- dictionary update:
  for $k \leftarrow 1$ **to** r
    $\mathbf{V}^k \leftarrow Diag(\mathbf{Z}^k)Diag(\|\mathbf{U}^k\|_2^2\mathbf{Z}^k + np\lambda\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{U}^k - \mathbf{V}\mathbf{U}^T\mathbf{U}^k + \|\mathbf{U}^k\|_2^2\mathbf{V}^k)$
  endfor
endwhile

of our approach in finding a meaningful sparse representation of grasping actions which were recorded by means of a dataglove.

4.1 Test 1: Retrieving the Original Dictionary

In this first series of experiments all datasets have been generated by a linear combination of atoms with a fixed structure, and varying the sparsity of both the atoms and the coefficients of the linear combinations. Three methods have been compared: K-SVD [1], which is able to find a sparse representation of the signals with neither structured nor sparse atoms, SSPCA [9] which is able to create structured sparse atoms with no sparse representation of the signals, and the proposed approach SR-SSD.

Experimental Set-up and Procedures. In this series of experiments the i-th signal $\mathbf{x}$ of the dataset is computed as $\mathbf{x} = \mathbf{U}_i\mathbf{V}^T + \varepsilon$, where each row $\mathbf{U}_i$ is sparse, and ε is a noise vector drawn from a Gaussian distribution with zero mean and varying the variance according to the values in table 1. The non zero coefficients of $\mathbf{U}_i$ are again drawn from a Gaussian distribution with zero mean and unit variance. The indexes of the non zero coefficients were chosen in a random way according to a uniform distribution. The number of non zero coefficients of $\mathbf{U}_i$ was varied as reported in table 1.

Following [9] we have organized the elements of $\mathbf{V}$ on a $N \times N$ dimensional grid with $N = p^{\frac{1}{2}}$. Only a fraction of the elements of $\mathbf{V}$ are non zero and correspond to a square on the grid. The number of non zero elements was chosen in order to have a certain degree of sparsity for the atoms as reported in table 1. Thus, for a given degree of sparsity, the size of the square is fixed while its position is chosen in a random way. The vectors $\mathbf{d}^i$ are chosen in such a way to favour atoms with non zero elements corresponding to a square on the grid as suggested in [9]. In order to compare the computed dictionary with the original dictionary we have used the same procedure proposed in [1]. In particular for each atom of the original dictionary $\mathbf{V}$ we search the closest column in the computed dictionary $\tilde{\mathbf{V}}$ where the distance between two atoms $\mathbf{V}^j$ and $\tilde{\mathbf{V}}^k$ is defined as $1 - |\left(\mathbf{V}^j\right)^T \tilde{\mathbf{V}}^k|$. A distance less than 0.01 is considered a success. For both SSPCA and SR-SSD methods the regularization

Table 1 Parameters used for Test 1.

(r) num. atoms	50
(p) dim. signals	400
(L) num. atoms for each signals (SR-SSD and K-SVD only)	$2, 8, 16$
(ε) noise std σ	$0, 0.025, 0.05, 0.075, 0.125$
(n) number of signals	250
percentage of zero elements of the atoms	$75\%, 90\%, 99\%$
$log_2(\lambda)$ is searched in the range	$[-11, -19]$ at step -1

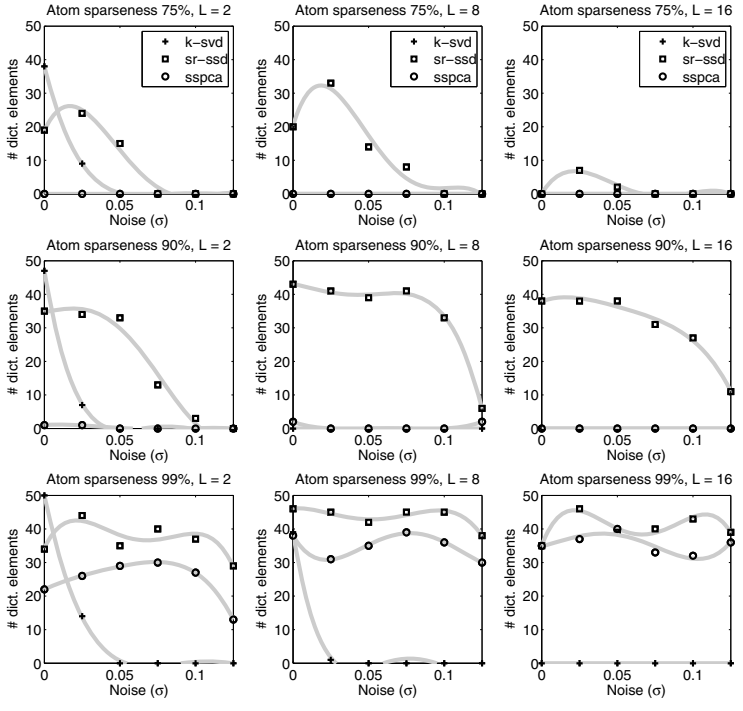

Fig. 1 Performance in retrieving the original structured dictionary. The figure shows the performances of K-SVD, SR-SSD, and SSPCA in retrieving the original dictionary, composed of 50 atoms, against the variation of the atom sparseness (on each row), the coefficient sparseness (on each column), and the noise level (in each plot).

parameter λ has been chosen by a 5-fold cross validation on the representation error. For all methods the number of atoms to retrieve has been set to the number r of atoms composing the original dictionary.

Results. In Figure 1 the results of Test 1 were reported. As expected K-SVD algorithm gives good performance when there is a strong sparsity of the coefficients (see first column). However, the performance decreases rapidly as soon as the noise

variance or the sparsity of the atoms increases. On the other hand, the SSPCA algorithm gives a good performance when sparsity of the atoms is very high (see last row) and the sparsity of the coefficients is not so high (compare first and third column in the last row). SR-SSD algorithm performs very well when there is sparsity on both coefficients and atoms (see last two rows) and it seems more tolerant to noise with respect to both K-SVD and SSPCA. Moreover a relative good performance for SR-SSD is also retained when the sparsity on the atoms is not so high (see first row).

4.2 Test 2: Finding a Meaningful Decomposition of Grasping Actions

Here we compare our method SR-SSD with SSPCA method. The experiments are aimed at testing the ability of both methods in finding a "compact" representation of grasping actions executed by human beings preserving the possibility to identify meaningful parts of the actions which are most discriminant in a classification process.

Experimental Set-up and Procedures. The dataset was composed of two different types of grasps: *precision-grasp* and *power-grasp*. In performing a power-grasp, the object is held in a clamp formed by fingers and palm; in performing a precision-grasp, the object is pinched between the tip of index finger and the opposing thumb. A *bottle top* and a *tennis ball* were used for the precision-grasp and power-grasp, respectively. The actions were recorded by means of the HumanGlove (Humanware S.r.l., Pontedera (Pisa), Italy) endowed with 16 sensors.

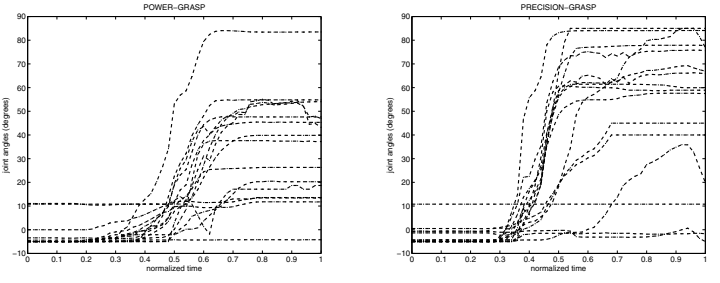

Fig. 2 Example of (left) precision-grasp and (right) power-grasp action recorded by means of the dataglove. Each graph shows the temporal profile of the 16 hand joint angles.

A total of 40 grasping actions were recorded (20 for each type) executed by one subject. A subject was seated at a table with two clearly visible surface marks (m_1 and m_2) placed at a distance of roughly 40 cm from each other. For each target object, the subject was asked to position the right hand on starting position m_1 and in a prone position, and to reach and grasp the target object placed on mark m_2. All actions were aligned to a fixed length *Len* and a generic input vector $\mathbf{x}$ was construed as the concatenation of all sensors for all times thus having a dimension of $Len \times 16$.

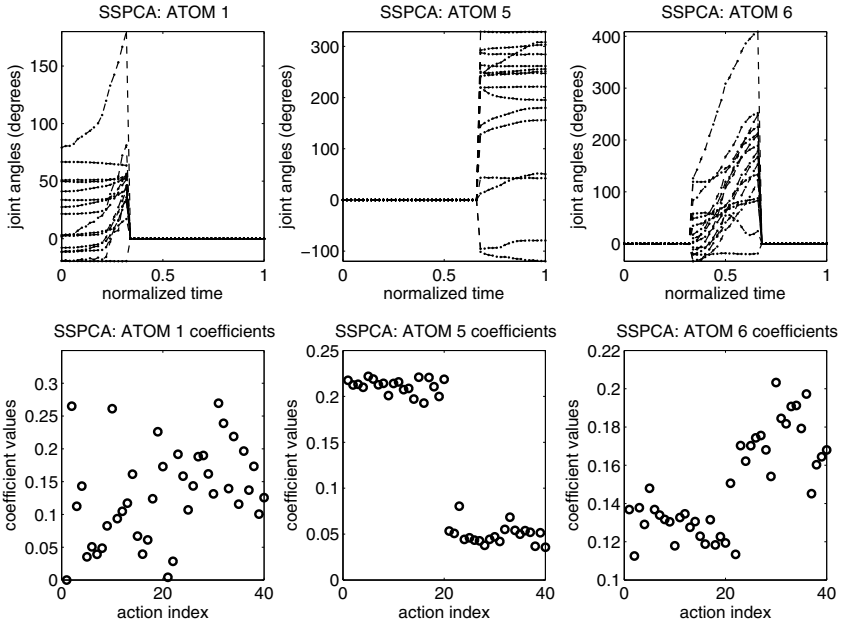

Fig. 3 Dictionary computed by SSPCA. The figure shows (top) 3 out of 6 atoms computed by SSPCA together with (bottom) their coefficients.

A simple dictionary structure was imposed in order to select variables related to the first, the central, and the latter part of the action. To this aim three vectors $\mathbf{d}^i$ were used with elements $d^i_j = 1$ if $\frac{(i-1)*Len}{3} + 1 \leq j \leq \frac{i*Len}{3}$ otherwise $d^i_j = 0$. For both methods six atoms suffice to have a reconstruction error of the data ≤ 0.3. As for test 1, cross validation was used to find the penalization parameters λ for both methods. For SR-SSD since we do not want to impose a fixed number of coefficients for each signal we prefer to work with an ℓ_1 IST algorithm [5].

Results. In Figures 3 and 4 are reported 3 out of 6 computed atoms together with the corresponding coefficients. The actions are ordered so that the first 20 coefficients are related to precision-grasp actions and the remaining 20 to power-grasp actions. As one can see from Figure 3, SSPCA method tends to select atoms with elements $\neq 0$ corresponding to just one of three consecutive parts of the actions as imposed by the vectors d^i. In particular, the coefficients of the first atom (see Figure 3) are almost all different from zeros with similar values for both action classes. This is not surprising since the first part of the action is the less informative in discriminating the two classes as the hand always starts in the same prone position. On the other hand the proposed SR-SSD method selects atoms with elements > 0 corresponding to the second and the third part of the actions only, which are the action parts most informative in terms of discriminating the two classes of action (see Figure 4).

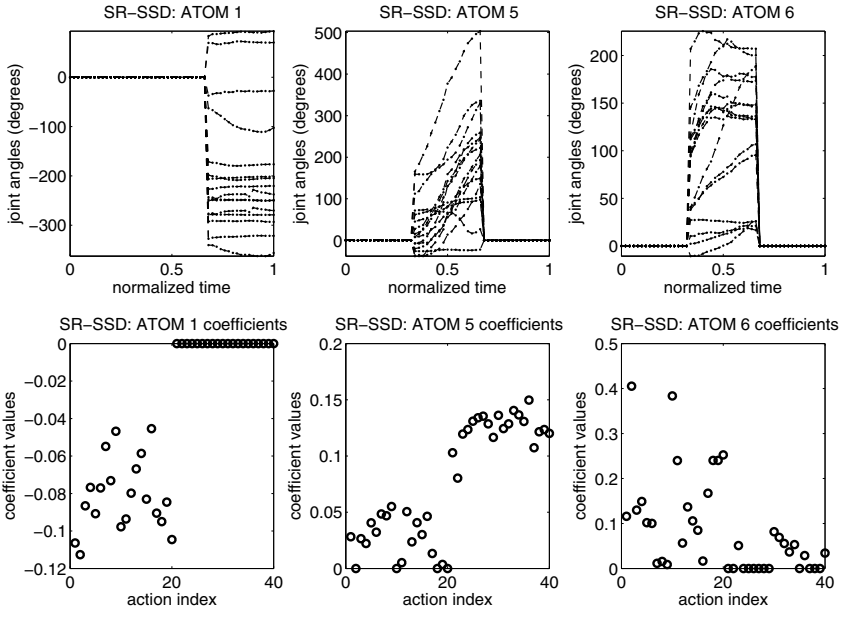

Fig. 4 Dictionary computed by SR-SSD. The figure shows (top) 3 out of 6 atoms computed by SR-SSD together with (bottom) their coefficients.

5 Conclusions

In this paper, we proposed a method to obtain a sparse data representation using a structured sparse dictionary. The results of experiments in Section 4.1 show that the proposed approach (SR-SSD) has the benefits of both K-SVD and SSPCA which use either sparse data representation or structured sparse dictionary. Moreover, results of experiments in Section 4.2, show how the proposed algorithm could be useful for obtaining an interpretable data dimensionality reduction. To this regard note that one could correctly classify the actions on the basis of SSPCA or SR-SSD coefficients. However, the proposed method is also able to select only those "parts" of the actions that most probably contribute in positive to the classification of the actions.

Acknowledgements. This work was partly supported by the project Dexmart (contract n. ICT-216293) funded by the EC under the VII Framework Programme and from the project Action Representations and their Impairment (2010-2012) funded by Fondazione San Paolo (Torino) under the Neuroscience Programme.

References

[1] Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54(11), 4311–4322 (2006)

[2] Basso, C., Santoro, M., Verri, A., Villa, S.: Paddle: Proximal algorithm for dual dictionaries learning (2010), CoRR, abs/1011.3728
[3] Bredies, K., Lorenz, D.: Linear Convergence of Iterative Soft-Thresholding. J. Fourier Anal. Appl. 14(5-6), 813–837 (2008)
[4] Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing 20(1), 33–61 (1998)
[5] Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on Pure and Applied Mathematics 57(11), 1413–1457 (2004)
[6] Engan, K., Aase, S.O., Hakon Husoy, J.: Method of optimal directions for frame design. In: Proc. of ICASSP 1999, vol. 5, pp. 2443–2446. IEEE Computer Society, Los Alamitos (1999)
[7] Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference and Prediction, corrected edition. Springer, Heidelberg (2003)
[8] Jenatton, R., Audibert, J.Y., Bach, F.: Structured variable selection with sparsity-inducing norms. Technical report, arXiv:0904.3523 (2009)
[9] Jenatton, R., Obozinski, G., Bach, F.: Structured sparse principal component analysis. In: International Conference on AISTATS (2010)
[10] Micchelli, C.A., Pontil, M.: Learning the kernel function via regularization. Journal of Machine Learning Research 6, 1099–1125 (2005)
[11] Tropp, J.A.: Greed is good: Algorithmic results for sparse approximation. IEEE Trans. Inform. Theory 50, 2231–2242 (2004)
[12] Vinjamuri, R., Lee, H.N., Mao, Z.H.: Dimensionality reduction in control and coordination of the human hand. IEEE Trans. Biomed. Eng. 57(2), 284–295 (2010)
[13] Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T.S., Yan, S.: Sparse Representation for Computer Vision and Pattern Recognition. Proceedings of the IEEE 98(6), 1031–1044 (2010)
[14] Zou, H., Hastie, T., Tibshirani, R.: Sparse Principal Component Analysis. Journal of Computational and Graphical Statistics 15 (2004)