Classification with Incoherent Kernel Dictionary Learning

Denis C. ILIE-ABLACHIM
Faculty of Automatic Control and Computers,
University Politehnica of Bucharest
denis.ilie_ablachim@upb.ro

Bogdan DUMITRESCU
Faculty of Automatic Control and Computers,
University Politehnica of Bucharest
bogdan.dumitrescu@upb.ro

***Abstract*—In this paper we present a new method of classification based on Dictionary Learning (DL). The main contribution consists of a kernel version of incoherent DL, derived from its standard linear counterpart. We also propose an improvement of the AK-SVD algorithm concerning the representation update. Our algorithms are tested on several popular databases of classification problems.**

***Index Terms*—dictionary learning, kernel, incoherence, classification**

I. Introduction

Dictionary Learning (DL) is a representation learning method used in signal processing and machine learning that aims to find a sparse representation for input data organized as vectors. DL has many applications starting from simple ones like image denoising, inpainting or signal reconstruction and going to coding, clustering or classification. For a given set of samples, Y, represented by a matrix of N columns (signals) of size m, we intend to find a dictionary D of size $m \times n$ and a sparse representation X of size $n \times N$ such that good sparse representations $Y \approx DX$ are obtained. The representation is based on linear combinations of the columns of the dictionary D, named atoms. The DL problem can be formulated as follows

$$\begin{array}{ll} \min_{D,X} & \|Y - DX\|_F^2 \\ \text{s.t.} & \|x_\ell\|_0 \leq s, \ell = 1:N \\ & \|d_j\| = 1, j = 1:n, \end{array} \tag{1}$$

where $\|\cdot\|_0$ represents the 0-pseudo-norm and s is the sparsity level. More precisely, each signal is represented as a linear combination of at most s atoms.

There are several successful DL methods, including K-SVD [1], MOD [2]; improved methods and variations of the DL problem including regularization and coherence reduction are presented in [3]. All these algorithms are iterative and in most of them an iteration consists of computing the sparse representations X with fixed dictionary D and then updating the atoms successively, possibly together with the coefficients with which an atom contributes to representations. Of special interest is the Approximate version of K-SVD problem (AK-SVD) [4], which does not seek exact optimality for both an atom and its representation coefficients, but optimizes them successively. AK-SVD has lower complexity than other algorithms and gives similar end results in most DL problems.

In this paper we present a new perspective on a classification problem via dictionary learning with incoherent atoms. This problem was first introduced in [5], where the solution is computed by optimizing the whole dictionary. We introduce a new optimization method in AK-SVD style, in which the dictionary D is updated atom by atom. Our contribution is to extend the problem by projecting the signals in a nonlinear space, as linear spaces can hinder classification performance. To this purpose, we use kernel representations in order to better quantify the similarity between signals. Another contribution is to introduce a new update rule for the coefficients representations, by taking into consideration only the most recent atoms in all computations; this improvement can lead to the increase of classification accuracy.

The contents of this paper is as follows. In Section II-A we introduce the classification problem and the principle of its solution via DL. Section II-B presents an incoherent DL algorithm suited for classification. Section II-C contains our main contribution: the kernel version of the incoherent DL algorithm and the new update rule for representations. Section III is dedicated to experimental results, obtained by running simulations on three publicly available datasets, namely YaleB, AR Face and Caltech 101.

II. Classification with Dictionary Learning

A. Standard Dictionary Learning classification

The representation learning approach (1) can be also used in classification problems. Considering a set of feature vectors classes $Y = [Y_1, \ldots, Y_c, \ldots, Y_C]$, where the columns of matrix $Y_c \in \mathbb{R}^{m \times N_c}$ are the vectors belonging to class c, we intend to learn a specific dictionary, D_c, for each class. For a given test signal $y \in \mathbb{R}^m$ the classification is achieved by finding the dictionary with the smallest residual of the representation:

$$c = \text{argmin}_{i=1:C} \|y - D_i x_i\|, \text{ with } \|x_i\|_0 \leq s. \tag{2}$$

B. Incoherent Dictionary Learning classification

In order to improve the performance of the classification, the problem can be extended by adding discriminative power

This work was supported by a grant of the Romanian Ministry of Education 2and Research, CCCDI - UEFISCDI, project number PN-III-P2-2.1-PED-2019-3248, within PNCDI III.

Algorithm 1: Incoherent AK-SVD Dictionary Update

Data: current dictionary $D \in \mathbb{R}^{m \times n}$
complementary dictionary $\bar{D} \in \mathbb{R}^{m \times (n-1)}$
representation matrix $X \in \mathbb{R}^{n \times N}$
Result: updated dictionary D
Compute error $E = Y - DX$
for $j = 1$ **to** n **do**
  Modify error: $F = E_{\mathcal{I}_j} + d_j X_{j,\mathcal{I}_j}$
  Update atom: $d_j = F X_{j,\mathcal{I}_j}^\top - 2\gamma \bar{D} \bar{D}^\top d_j$
  Normalize atom: $d_j \leftarrow d_j \|d_j\|$
  Update representation: $X_{j,\mathcal{I}_j}^\top = F^\top d_j$
  Recompute error: $E_{\mathcal{I}_j} = F - d_j X_{j,\mathcal{I}_j}$

to each dictionary. By this, we intend to maintain a good sparse representation for its own class while achieving a bad representation for the other classes. A solution for this problem was presented in [6] where a penalty term was added to the DL problem, transforming it into

$$\min_{D_i, X_i} \sum_{i=1}^{C} \|Y_i - D_i X_i\|_F^2 + \gamma \sum_{i=1}^{C} \sum_{l \neq i} \left\| D_i^\top D_l \right\|_F^2 . \quad (3)$$

The second term introduces an incoherence measure between pairs of dictionaries from different classes. By this formulation we intend to project dictionaries into quasi-orthogonal spaces, while retaining most of their representation ability.

The DL problem (3) can be approximately solved by an approach similar to Approximated K-SVD [1]. The optimization consists of an iterative process in which the representations X_i and the dictionaries D_i are alternately optimized while all other variables are fixed. The representations are computed with Orthogonal Matching Pursuit [7] as usual in DL, since the penalty term does not depend on X_i. The dictionaries are updated sequentially, atom by atom. Let us assume that we optimize atom d_j from dictionary D_i. The optimization problem (3) becomes

$$\min_{d_j} \left\| F_{ij} - d_j X_{j,\mathcal{I}_j} \right\|_F^2 + 2\gamma \sum_{l \neq i} \left\| D_l^\top d_j \right\|_F^2 , \quad (4)$$

where $F_{ij} = \left[Y_i - \sum_{\ell \neq j} d_\ell x_\ell^\top \right]_{\mathcal{I}_j}$ is the representation error when all atoms but d_j are considered and $\mathcal{I}_j$ denotes the indices of the nonzero positions on the jth row of X_i (those containing the coefficients of d_j in the representations). The solution has been previously presented in [6] and is

$$d_j \leftarrow F_{ij} x - 2\gamma \bar{D} \bar{D}^\top d_j , \quad (5)$$

where we have denoted $x = X_{j,\mathcal{I}_j}$ and $\bar{D} = [D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_C]$ is the complementary dictionary to the current one.

The atom update operations of the IDL algorithm based on AK-SVD are summarized in Algorithm 1 [3, Alg.4.2]. Note that the representations are also updated and that the representation error is manipulated efficiently.

C. Incoherent Kernel Dictionary Learning classification

In order to evade the linear character of the representation, kernel dictionary learning (KDL) was introduced in [8], [9]. Through this method, the space of signals is extended to a nonlinear feature vector space. We associate with each signal y the feature vector $\varphi(y)$, where $\varphi(y)$ is a nonlinear function. The dictionary D is also extended to a nonlinear space by $\varphi(Y)A$, where A contains the coefficients of the dictionary. So, the DL problem (1) is transformed into

$$\begin{array}{ll} \min_{A,X} & \|\varphi(Y) - \varphi(Y) A X\|_F^2 \\ \text{s.t.} & \|x_\ell\|_0 \leq s, \ell = 1:N \\ & \|a_j\| = 1, j = 1:n. \end{array} \quad (6)$$

The problem becomes computationally tractable by the use of Mercer kernels, which allows the substitution of scalar product of feature vectors with the computation of a kernel function $k(x,y) = \varphi(y)^\top \varphi(x)$. Denoting $K_{il} = \varphi(Y_l)^\top \varphi(Y_i)$, the incoherent DL problem (3) is transformed into the Incoherent Kernel Dictionary Learning (IKDL) problem

$$\min_{A_i, X_i} \sum_{i=1}^{C} \|\varphi(Y_i) - \varphi(Y_i) A_i X_i\|_F^2 + \gamma \sum_{i=1}^{C} \sum_{l \neq i} \left\| A_l^\top K_{il} A_i \right\|_F^2 . \quad (7)$$

Using a similar alternate optimization technique and similar notations, the kernel correspondent of problem (4) for optimizing an atom a_j is

$$\min_{a_j} \left\| \varphi(Y_i) F_{ij} - \varphi(Y_i) a_j X_{j,\mathcal{I}_j} \right\|_F^2 + 2\gamma \sum_{l \neq i} \left\| A_l^\top K_{il} a_j \right\|_F^2 . \quad (8)$$

In order to solve this optimization problem, we compute the partial derivatives with respect to atom a_j as follows:

$$\frac{\partial \left\| \varphi(Y_i) \left(F_{ij} - a_j x^\top \right) \right\|_F^2}{\partial a_j} = -2 K_{ii} \left(F_{ij} - a_j x^\top \right) x \quad (9)$$

and

$$\frac{\partial \left\| A_l^\top K_{il} a_j \right\|_F^2}{\partial a_j} = 2 K_{il}^\top A_l A_l^\top K_{il} a_j . \quad (10)$$

By using (9) and (10), the minimum in (8) is obtained when

$$-K_{ii} \left(F_{ij} - a_j x^\top \right) x + 2\gamma \sum_{l \neq i} K_{il}^\top A_l A_l^\top K_{il} a_j = 0 \quad (11)$$

and so the solution is

$$a_j = \left(K_{ii} \|x\|^2 + 2\gamma \sum_{l \neq i} K_{il}^\top A_l A_l^\top K_{il} \right)^{-1} K_{ii} F_{ij} x . \quad (12)$$

The resulting atom is the solution of a $m \times m$ linear system. Given the complexity of the problem, we intend to find a more convenient approximation.

We note that, given the atom a_j, the optimal associated representation in (8) is $X_{j,\mathcal{I}_j}^\top = F_{ij}^\top K_{ii} a_j$, like in the kernel

AK-SVD algorithm (the penalty does not contain the representation). We insert this optimal representation in (8) and obtain

$$\min_{a_j} \left\|\varphi(Y_i)\left(F_{ij} - a_j a_j^\top K_{ii} F_{ij}\right)\right\|_F^2 + 2\gamma \left\|\hat{K}_i a_j\right\|_F^2, \quad (13)$$

where

$$\hat{K}_i = \left[K_{i1}^\top A_1 \ \ldots \ K_{i,i-1}^\top A_{i-1} \ K_{i,i+1}^\top A_{i+1} \ \ldots \ K_{iC}^\top A_C\right]^\top. \quad (14)$$

Expressing the Frobenius norm via its trace form, the new objective from (13) becomes

$$Tr\left[\left(F_{ij} - a_j a_j^\top K_{ii} F_{ij}\right)^\top K_{ii}\left(F_{ij} - a_j a_j^\top K_{ii} F_{ij}\right)\right] + 2\gamma Tr\left[a_j^\top \hat{K}_i^\top \hat{K}_i a_j\right]. \quad (15)$$

After direct transformations and neglecting the terms that do not depend on a_j, we are left with the minimization of

$$- a_j^\top \left(K_{ii} F_{ij} F_{ij}^\top K_{ii} - 2\gamma \hat{K}_i^\top \hat{K}_i\right) a_j. \quad (16)$$

The solution is the eigenvector corresponding to the maximum eigenvalue of the matrix

$$H = K_{ii} F_{ij} F_{ij}^\top K_{ii} - 2\gamma \hat{K}_i^\top \hat{K}_i. \quad (17)$$

Since this is again a high complexity operation, we make a single iteration of the power method on the matrix H. So, given the current atom $a_j^{(k)}$ (at iteration k), the new atom is

$$a_j^{(k+1)} = H a_j^{(k)} = K_{ii} F_{ij} x - 2\gamma \hat{K}_i^\top \hat{K}_i a_j^{(k)}, \quad (18)$$

followed by atom normalization. We have denoted again $x = X_{j,\mathcal{I}_j}$. The atom update (18) is the kernel version of (5).

The atom update operations of the IKDL algorithm are summarized in Algorithm 2 for a single dictionary (hence the index i has disappeared). We also propose an improvement with respect to the structure of Algorithm 1. We note that the representation update uses the most recent version of the current atom; however, the error matrix F is computed using the previous version of the atom. By introducing the most recent version of the atom in the error, the representation update becomes

$$\left(X_{j,\mathcal{I}_j}^\top\right)^{(k+1)} = F^\top K a_j = E_{\mathcal{I}_j}^\top K a_j + \left(X_{j,\mathcal{I}_j}^\top\right)^{(k)}. \quad (19)$$

Due to normalization, we have $a_j^\top K a_j = 1$ and so this product has disappeared from the second term above. We name Updated-error AK-SVD (UAK-SVD) this version of the algorithm and we will compare it with the usual AK-SVD update. The difference is only in the representation updates, step 6 of Algorithms 1 and 2.

For the classification scheme we need only the reconstruction errors from equation (2). For the kernel version, the classification of a signal y results from

$$c = \text{argmin}_{i=1:C} \|\varphi(y) - \varphi(Y_i) A_i x_i\|, \text{ with } \|x_i\|_0 \le s, \quad (20)$$

Algorithm 2: Incoherent Kernel UAK-SVD Dictionary Update

Data: kernel matrix $K \in \mathbb{R}^{N \times N}$
current dictionary $A \in \mathbb{R}^{N \times n}$
complementary dictionary $\hat{K} \in \mathbb{R}^{(N-1) \times N}$
representation matrix $X \in \mathbb{R}^{n \times N}$
Result: updated dictionary D

1 Compute error $E = I - AX$
2 **for** $j = 1$ **to** n **do**
3 Modify error: $F = E_{\mathcal{I}_j} + a_j X_{j,\mathcal{I}_j}$
4 Update atom: $a_j = KFX_{j,\mathcal{I}_j} - 2\gamma \hat{K}^\top \hat{K} a_j$
5 Normalize atom: $a_j \leftarrow \left(a_j^\top K a_j\right)^{\frac{1}{2}}$
6 Update representation: $X_{j,\mathcal{I}_j}^\top \leftarrow E_{\mathcal{I}_j}^\top K a_j + X_{j,\mathcal{I}_j}^\top$
7 Recompute error: $E_{\mathcal{I}_j} = F - a_j X_{j,\mathcal{I}_j}$

which leads to

$$c = \text{argmin}_{i=1:C} k(y,y) + x_i^\top A_i^\top K_i A_i x_i - 2k(y, Y_i) A_i x, \text{ with } \|x_i\|_0 \le s. \quad (21)$$

Here, as well as in the IKDL algorithm, the representations are computed with Kernel OMP [8].

III. Experiments

In this section we present the main results obtained with the Incoherent Kernel Dictionary Learning algorithm. The datasets used in the simulation are YaleB [10], AR Face [11] and Caltech 101 [12].

For the evaluation step, each dataset is independently used and was provided in [13]. We measure performance through classification accuracy, training time and testing time. All the algorithms were developed in Matlab $2018a$, on a laptop with 3.5GHz Intel CPU and 16 GB RAM memory. The execution time and accuracy are reported as the average over the 3 best results. For the methods that require the use of a kernel function, we used two types of kernels: radial basis function kernel ($k(x,y) = \exp \frac{-||x-y||_2^2}{2\sigma^2}$) and polynomial kernel ($k(x,y) = (x^\top y + \alpha)^\beta$). For the kernel problems, we have tried different numerical forms in our simulations. We have chosen the final form based on the best results from these simulations. The code for the proposed algorithms is available at https://github.com/denisilie94/Incoherent-Kernel-Dictionary-Learning.

YaleB Database is organized into two sub-datasets, according to the extended and cropped images. The dataset is composed of 16128 images of 38 human subjects under 9 poses and 64 illumination conditions. During the simulation step only the extended dataset was used, including 2414 face images of 38 persons. For the training and testing step the images per subject were split in half. The dimension of the feature vectors is 504.

AR Face Database is a face dataset which contains over 4000 color images corresponding to 126 different people (70 men and 56 women). The images were taken having a frontal view with different facial expressions, illumination conditions and occlusions. For the experimental phase a set of 2600 images of 50 females and 50 male subjects are extracted. For each subject, 20 images were used for training and 6 for testing.

Beside the face recognition task, an object recognition task was attempted in the simulations. For this we used **Caltech 101 Database**. The dataset includes 9,144 images from 102 classes (101 common object classes and a background class). The number of samples in each category varies from 31 to 800. In the experiments, 30 samples per category were used for training, while the rest are used for testing.

During the simulations we performed tests with dictionaries of different sizes (40, 60, 80 and 100 atoms) having a sparsity constraint equal to 10%, 20%, 50% and 80% of the number of atoms. Taking into account the training time and the resulted classification accuracy, we chose to use only dictionaries with 40 atoms and a sparsity constraint of 20. Increasing sparsity can improve the results, but this will also affect the training time. All tests were performed on 10 DL iterations. For a larger number of iterations the improvement in accuracy is insignificant. We set the hyperparameters of the optimization problem following a grid search: $\gamma \in [0.01, 0.1, 0.5, 1, 2, 4, 6]$, $\sigma \in [0.5, 1, 2, 4, 5, 6, 8, 10]$, $\alpha \in [0.5, 1, 2, 4]$ and $\beta \in [2, 3]$. In the case of all datasets, for the IDL problem we used $\gamma = 4$, while for the IKDL problem γ was set to 0.1. Regarding the kernel functions, we used the following parameters: $\sigma = 4$, $\alpha = 2$ and $\beta = 2$ for YaleB dataset; $\sigma = 8$, $\alpha = 4$ and $\beta = 2$ for AR Face dataset; and $\sigma = 5$, $\alpha = 4$ and $\beta = 2$ for Caltech 101 dataset.

The main results are summarized in Tables I, II for classification with incoherent DL; Tables III, IV contain results with IKDL and the RBF kernel; Tables V and VI contain results with IKDL and polynomial kernel. As we can see, the results vary depending on the chosen algorithm. The UAK-SVD method usually improves the classification accuracy, although sometimes only slightly. Regarding the kernel extension, the introduced nonlinearity does not always insure an improvement, as we can see for YaleB dataset, but there is a strong improvement regarding the execution time. In the case of YaleB dataset, the training time decreased by 10 times, while for the AR Face dataset the training is done 25 times faster. The best improvement is visible for the Caltech 101 dataset, where training time has been reduced 200 times. The execution time is reduced due to the small size of the dictionaries in the kernel version. This property is valid only for cases where the signal size is much larger than the number of signals per class; for example, in the YaleB case, the dictionary of a class has size 504×40 in the IDL approach, but size only 32×40 in IKDL; it is thus remarkable that the accuracy loss is so small when kernels are used. This property is also valid for the other datasets, where we have signals of size 540 for AR Face dataset and 3000 for Caltech 101 dataset.

In order to better understand the classification problem we compute the reconstruction error (figures 1, 2 and 3) and the discriminative term (figures 4, 5 and 6). Based on the exploitation of the two terms we can easily see that the reconstruction error achieves good representation for YaleB and AR Face datasets, while the discriminative term does not produce the quasi-orthogonality of the dictionaries. For these problems, the classification obtains good results by taking γ small enough so that the discriminative term does not have an important weight in the objective function. On the other side, the Caltech 101 dataset does not achieve a separable error reconstruction, but the discriminative term is stronger and thus classification can be performed.

TABLE I
AK-SVD INCOHERENT DICTIONARY LEARNING

Dataset \ Perf.	Train. time	Test. time	Accuracy
YaleB	82.9 [sec]	20.6 [sec]	94.00%
AR Face	558 [sec]	27 [sec]	93.22%
Caltech101	14332 [sec]	329 [sec]	67.30%

TABLE II
UAK-SVD INCOHERENT DICTIONARY LEARNING

Dataset \ Perf.	Train. time	Test. time	Accuracy
YaleB	87.5 [sec]	20.8 [sec]	94.11%
AR Face	560 [sec]	26.3 [sec]	93.33%
Caltech101	14367 [sec]	329 [sec]	66.98%

TABLE III
AK-SVD INCOHERENT KERNEL DICTIONARY LEARNING
(RBF KERNEL)

Dataset \ Perf.	Train. time	Test. time	Accuracy
YaleB	6.8 [sec]	23.8 [sec]	93.88%
AR Face	19.5 [sec]	25.3 [sec]	93.42%
Caltech101	62.4 [sec]	403 [sec]	70.67%

TABLE IV
UAK-SVD INCOHERENT KERNEL DICTIONARY LEARNING
(RBF KERNEL)

Dataset \ Perf.	Train. time	Test. time	Accuracy
YaleB	7.2 [sec]	24.7 [sec]	94.05%
AR Face	19.2 [sec]	25.3 [sec]	93.50%
Caltech101	62.7 [sec]	402 [sec]	70.12%

TABLE V
AK-SVD INCOHERENT KERNEL DICTIONARY LEARNING
(POLYNOMIAL KERNEL)

Dataset \ Perf.	Train. time	Test. time	Accuracy
YaleB	9.0 [sec]	26.2 [sec]	94.00%
AR Face	24.5 [sec]	30.5 [sec]	94.83 %
Caltech101	73.3 [sec]	430 [sec]	70.83%

TABLE VI
UAK-SVD INCOHERENT KERNEL DICTIONARY LEARNING
(POLYNOMIAL KERNEL)

Dataset \ Perf.	Train. time	Test. time	Accuracy
YaleB	9.0 [sec]	27.1 [sec]	94.07%
AR Face	24.5 [sec]	30.3 [sec]	94.83%
Caltech101	73.6 [sec]	428 [sec]	71.46%

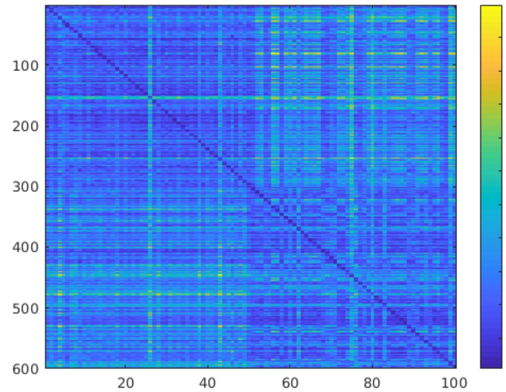

Fig. 1. $\|\varphi(y) - \varphi(Y_i)A_i x\|_F^2$ (YaleB - UAK-SVD IKDL)

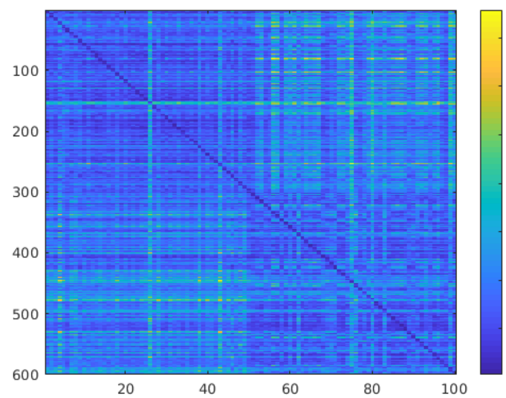

Fig. 2. $\|\varphi(y) - \varphi(Y_i)A_i x\|_F^2$ (AR Face - UAK-SVD IKDL)

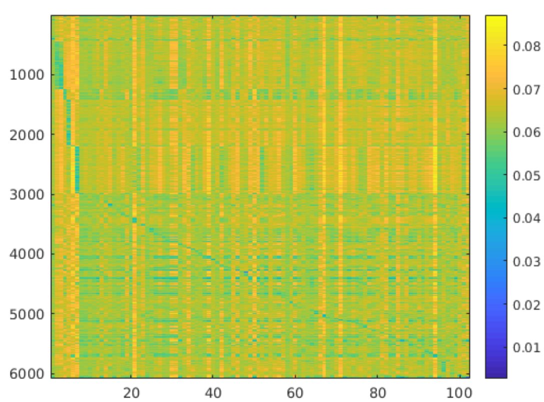

Fig. 3. $\|\varphi(y) - \varphi(Y_i)A_i x\|_F^2$ (Caltech101 - UAK-SVD IKDL)

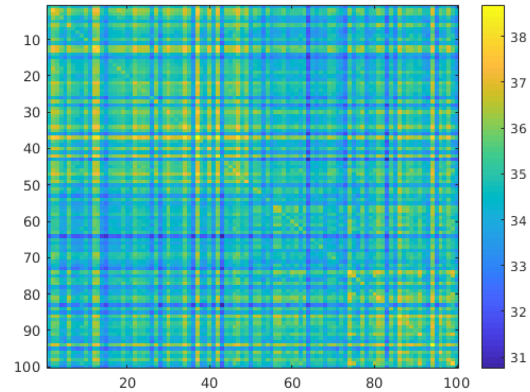

Fig. 4. $\left\|A_l^\top K_{il} A_i\right\|_F^2$ (YaleB - UAK-SVD IKDL)

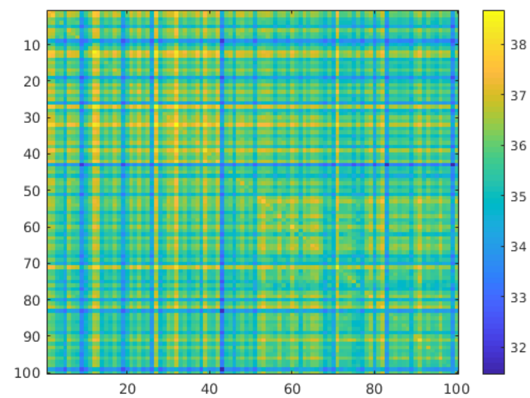

Fig. 5. $\left\|A_l^\top K_{il} A_i\right\|_F^2$ (AR Face - UAK-SVD IKDL)

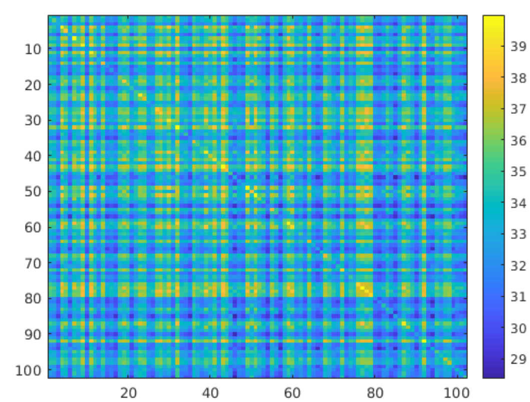

Fig. 6. $\left\|A_l^\top K_{il} A_i\right\|_F^2$ (Caltech101 - UAK-SVD IKDL)

IV. Conclusions

In this paper we have extended the family of dictionary learning algorithms for classification problems. We have presented a modified version of AK-SVD in which the most recent version of an atom is used in all respects in the representation update. We have proposed a kernel version of incoherent AK-SVD that can improve classification performance by increasing the separation of dictionaries dedicated to different signal classes. The experimental results confirm the good behavior of our algorithms, especially in terms of complexity.

REFERENCES

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[2] K. Engan, S.O. Aase, and J.H. Husoy. Method of optimal directions for frame design. In *IEEE Int. Conf. Acoustics Speech Signal Proc.*, volume 5, pages 2443–2446, 1999.

[3] Bogdan Dumitrescu and Paul Irofti. *Dictionary learning algorithms and applications*. Springer, 2018.

[4] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit. Technical Report CS-2008-08, Technion Univ., Haifa, Israel, 2008.

[5] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508. IEEE, 2010.

[6] Paul Irofti and Bogdan Dumitrescu. Regularized algorithms for dictionary learning. In *2016 International Conference on Communications (COMM)*, pages 439–442. IEEE, 2016.

[7] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

[8] Hien Van Nguyen, Vishal M Patel, Nasser M Nasrabadi, and Rama Chellappa. Design of non-linear kernel dictionaries for object recognition. *IEEE Transactions on Image Processing*, 22(12):5123–5135, 2013.

[9] J.J. Thiagarajan, K.N. Ramamurthy, and A. Spanias. Multiple Kernel Sparse Representations for Supervised and Unsupervised Learning. *IEEE Trans. Image Proc.*, 23(7):2905–2915, July 2014.

[10] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):643–660, 2001.

[11] Aleix M Martinez. The ar face database. *CVC Technical Report24*, 1998.

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[13] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2651–2664, 2013.