CLASSIFICATION WITH DICTIONARY LEARNING AND A DISTANCE BARRIER PROMOTING INCOHERENCE

Denis C. Ilie-Ablachim, Bogdan Dumitrescu

University Politehnica of Bucharest
Department of Automatic Control and Computers
313 Spl. Independenţei, 060042 Bucharest, Romania

ABSTRACT

We present a new approach to the incoherent dictionary learning problem using a barrier function that promotes incoherence. This function has a context-dependent quadratic term and a distance barrier term that can be used in both local and global structures. This strategy achieves better results in terms of error representation and incoherence of the dictionary, compared with the standard problem. We demonstrate on several datasets that this function can improve the performance of dictionaries in classification problems.

***Index Terms*—** dictionary learning, incoherence, sparse representation, classification

1. INTRODUCTION

Dictionary Learning (DL) is a method used in signal processing and machine learning to represent signals as a linear combination of an overcomplete basis of vectors, named atoms. The signals are usually represented under a sparsity constraint. This refers to the property of involving only a small number of atoms in the representation problem. This strategy is useful and it has a great impact on many applications, such as image denoising, inpainting, compression, feature extraction, signal reconstruction, clustering, and classification.

For a given set of N signals of size m, stored compactly in matrix $\boldsymbol{Y} \in \mathbb{R}^{m \times N}$, the dictionary learning problem can be formulated as

$$\begin{array}{ll} \min\limits_{\boldsymbol{D},\boldsymbol{X}} & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \\ \text{s.t.} & \|\boldsymbol{x}_\ell\|_0 \leq s, \ell = 1:N \\ & \|\boldsymbol{d}_j\| = 1, j = 1:n, \end{array} \tag{1}$$

where the variables are the dictionary matrix $\boldsymbol{D}$ of size $m \times n$, the sparse representation matrix $\boldsymbol{X}$ of size $n \times N$ and the sparse constraint s. The main task of this problem is to find a suitable approximation $\boldsymbol{Y} \approx \boldsymbol{D}\boldsymbol{X}$. In this way, each signal from $\boldsymbol{Y}$ is computed as a weighted sum of s atoms from $\boldsymbol{D}$ having the corresponding coefficients in matrix $\boldsymbol{X}$.

The optimization problem is solved in two main stages: dictionary update and sparse coding update. In the first stage, the sparse representation is considered fixed, and the dictionary $\boldsymbol{D}$ is updated in the direction that minimizes the reconstruction error. In the next stage, given the current dictionary, the sparse coefficients are updated for each signal. The problem is solved iteratively by successively going through the two stages, usually for a fixed number of iterations. In this way, good local optima can be obtained. For the update of the coefficient representation matrix $\boldsymbol{X}$, a fitting solution is Orthogonal Matching Pursuit (OMP) [1] algorithm; for the update of the dictionary matrix $\boldsymbol{D}$, there are several available methods [2]. A good choice for the current problem is the AK-SVD algorithm [3], which is an approximate form of the K-SVD method [4]. The AK-SVD algorithm is also preferable due to its low complexity and good performance.

Sparse representations and dictionary learning methods can be applied to problems involving supervised classification. An example of a sparse representation method is SRC (sparse representation-based classification) [5]. Two interesting dictionary learning methods involving discriminative penalty functions are Discriminative K-SVD [6] and Label Consistent K-SVD [7]. The first method solves the sparse representation problem while building a classifier matrix responsible for the feature classification, which shares the same coefficient matrix with the dictionary. The second method extends the discriminative problem by adding a new term responsible for producing sparse representations consistent with the class labels. Other relevant studies regarding the incoherence representation of sparse metrics used in classification problems exist. For example, in [8] the authors improve sparse signal representation by updating the dictionary using the method of optimal directions (MOD) and then applying a dictionary rank shrinkage step to reduce mutual coherence. Another good example is presented in [9], where the authors introduce an adaptive dictionary learning

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS - UEFISCDI, project number PN-III-P4-PCE-2021-0154, within PNCDI III.
It has also been funded by the Ministry of Investments and European Projects through the Human Capital Sectoral Operational Program 2014-2020, Contract no. 62461/03.06.2022, SMIS code 153735.

algorithm that addresses the consistency of learned quantities by approximating observed signals as a sum of rank one matrices.

Overall, incoherence is a desirable property in Dictionary Learning algorithms for classification problems. It enhances the discriminative power of the learned dictionaries. Incoherent dictionaries facilitate independent representation of the data, making it easier to discriminate between different classes. This is the main motivation for our work.

The content of this paper is organized as follows. In Section 2, we present the Incoherent Dictionary Learning (IDL) problem in a form that can be used for classification problems. Section 3 contains our main contribution, namely an IDL approach to classification using a distance barrier function (promoting incoherence), which can be a suitable substitute for the existing problem. We propose two formulations; the first is the simple modification of the IDL problem with the proposed barrier; the second is inspired by the triplet loss function used in machine learning, but never in the context of dictionary learning. Section 4 contains the experimental details of our tests and the main results. We present here results obtained on seven datasets: YaleB Face [10], AR Face [11], CMU PIE [12], Scene 15 [13], Caltech101 [14], UCF50 and HMDB51 action banks [15]. The proposed methods were compared to the standard IDL problem and with three other dictionary learning approaches, Projective Dictionary Pair Learning (DPL) [16], Self-Expressive Locality-Adaptive Latent Dictionary Pair Learning (SLatDPL) [17] and Low-rank Shared Dictionary Learning (LRSDL) [18], that are part of the state of the art competitors.

2. INCOHERENT DICTIONARY LEARNING

This section presents the standard Incoherent Dictionary Learning problem for classification tasks. Considering a set of feature vectors $\boldsymbol{Y} = [\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_c, \ldots, \boldsymbol{Y}_C]$, where $\boldsymbol{Y}_c \in \mathbb{R}^{m \times N_c}$ represent the set of samples corresponding to class c and C is the number of classes, we intend to learn a local dictionary, $\boldsymbol{D}_c$, for each class. The role of each dictionary is to obtain good representations for the corresponding class and bad representations for the rest of the classes. The objective function is built by extending the original DL problem (1) with an additional term that promotes incoherence between classes:

$$\sum_{i=1}^{C} \|\boldsymbol{Y}_i - \boldsymbol{D}_i \boldsymbol{X}_i\|_F^2 + \gamma \sum_{i=1}^{C} \sum_{j \neq i} \left\|\boldsymbol{D}_i^\top \boldsymbol{D}_j\right\|_F^2, \qquad (2)$$

where $\gamma > 0$ is a trade-off factor. After solving (2), a test signal $\boldsymbol{y} \in \mathbb{R}^m$ is classified by identifying the dictionary with the smallest representation error

$$c = \underset{i=1:C}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{D}_i \boldsymbol{x}_i\|, \text{ with } \|\boldsymbol{x}_i\|_0 \leq s. \qquad (3)$$

The problem (2) was presented in [19], where the authors introduced a discriminative measure between pairs of dictionaries from different classes. In this way, the dictionaries are projected into quasi-orthogonal spaces, while preserving the majority of their representational capabilities.

Problem (2) can be solved following an approach similar to the AK-SVD method. The sparse representation stage can be solved with OMP, since the incoherence terms do not depend on the coefficients $\boldsymbol{X}$. In the dictionary update stage, each atom of a dictionary $\boldsymbol{D}_i$ is updated sequentially in the direction of a space capable of good representation for the current class, being at the same time nearly orthogonal to the other class spaces. Notice that the objective depends on the scalar products of an atom with all atoms from different classes during the update. However, there may be atoms far enough for which the incoherence is satisfactory for classification and should not be considered. In the following section, we propose to solve this issue by introducing a selection procedure based on the coherence between atoms.

3. INCOHERENT DICTIONARY LEARNING VIA DISTANCE BARRIER

In this section, we reformulate the IDL problem in the context of defining an incoherent distance barrier (IDB) function that is responsible for the discrimination of dictionaries from different classes. This formulation was used before in the context of incoherent frames [20]. Moreover, this strategy demonstrates good behavior in problems involving the design of dictionaries for the simple purpose of representation. We propose extending the approach to the classification problem to promote inter-class incoherence. The IDB problem has the objective function

$$\sum_{i=1}^{C} \|\boldsymbol{Y}_i - \boldsymbol{D}_i \boldsymbol{X}_i\|_F^2 + \gamma \sum_{i=1}^{C} \sum_{j \neq i} f\left(\boldsymbol{D}_i, \boldsymbol{D}_j\right), \qquad (4)$$

where $f(\cdot)$ is the function promoting incoherence.

3.1. Incoherent Dictionary Learning via distance barrier

In this subsection, we present the first proposed version of IDB. We use the distance barrier function

$$b(\boldsymbol{D}_i, \boldsymbol{D}_j) = \sum_k \sum_l \Big[\max(0, M - \|\boldsymbol{d}_k^{(i)} - \boldsymbol{d}_l^{(j)}\|^2) + \max(0, M - \|\boldsymbol{d}_k^{(i)} + \boldsymbol{d}_l^{(j)}\|^2)\Big], \qquad (5)$$

where we define a soft margin M of the distance between atoms. Notice that the function takes into account both scenarios, in which the atoms have the same or the opposite direction. Denoting μ the desired mutual coherence between two atoms of different classes, we can define a margin M as in [20].

Remark 1. *The soft margin can be defined based on the mutual coherence constraint between a pair of atoms* $(\boldsymbol{d}_i, \boldsymbol{d}_j)$. *The imposed coherence constraint*

$$|\boldsymbol{d}_i^T \boldsymbol{d}_j| \leq \mu, \ \forall i \neq j \tag{6}$$

is equivalent to

$$\left\{ \begin{array}{l} \|\boldsymbol{d}_i - \boldsymbol{d}_j\|^2 \geq M \\ \|\boldsymbol{d}_i + \boldsymbol{d}_j\|^2 \geq M \end{array} \right. , \forall i \neq j \tag{7}$$

where $M = 2(1-\mu)$.

To improve the incoherence target, we define a function $f(\cdot)$ to be inserted in (4), by the addition of terms of the form

$$f_j(\boldsymbol{d}_j) = \|\boldsymbol{W}_j \bar{\boldsymbol{D}}_j^T \boldsymbol{d}_j\|^2 + \lambda b(\boldsymbol{d}_j), \tag{8}$$

where $\bar{\boldsymbol{D}}_j$ is the matrix obtained by concatenating all the dictionaries but that of the j-th class. The first term is quadratic and the second one is the barrier defined before. The matrix $\boldsymbol{W}_j$ is a diagonal weighting matrix

$$w_{ij}^2 = \max(|\boldsymbol{d}_i^T \boldsymbol{d}_j|/\mu, 1), \tag{9}$$

that imposes conditions for all atoms. The barrier functions $b(\cdot)$ prioritizes the update of the nearby atoms [21]. The first term discourages coherence for all-atom pairs, while the second, penalizes only pairs of too-close atoms.

The problem is solved by following a block coordinate descent procedure for a single atom $\boldsymbol{d}_j$ while the rest are fixed. The update is made using a gradient descent method

$$\boldsymbol{d}_j \leftarrow \boldsymbol{d}_j - \gamma_k g_j(\boldsymbol{d}_j), \tag{10}$$

where

$$\begin{aligned} g_j(\boldsymbol{d}_j) = {} & \boldsymbol{F}\boldsymbol{x}_j + \bar{\boldsymbol{D}}_j \boldsymbol{W}_j^2 \bar{\boldsymbol{D}}_j^T \boldsymbol{d}_j \\ & + \lambda \left[\sum_{\|\boldsymbol{d}_i - \boldsymbol{d}_j\| \leq M} (\boldsymbol{d}_i - \boldsymbol{d}_j) + \sum_{\|\boldsymbol{d}_i + \boldsymbol{d}_j\| \leq M} (-\boldsymbol{d}_i - \boldsymbol{d}_j) \right], \end{aligned} \tag{11}$$

where $\boldsymbol{F} = \left[\boldsymbol{Y}_i - \sum_{\ell \neq j} \boldsymbol{d}_\ell \boldsymbol{x}_\ell^\top\right]_{\mathcal{I}_j}$ is the representation error; $\mathcal{I}_j$ denotes the indices of the nonzero positions on the jth row of coefficient matrix $\boldsymbol{X}_i$; see the AK-SVD method [3] for details on the gradient of the error term.

Note that this strategy ensures a global incoherence between dictionaries from different classes. On the other hand, it is well-known that local incoherence at a dictionary level can improve the representation capabilities. The proposed method can be enhanced by combining the global barrier (between dictionaries of different classes) with a local barrier (between atoms of the same dictionary).

3.2. Incoherent Dictionary Learning via triplet distance barrier

In this subsection, we present a different approach for the distance barrier function, inspired from an idea used before in the triplet loss cost function [22]. We rewrite the incoherent dictionary learning objective function (4) as follows

$$\sum_{i=1}^{C} \|\boldsymbol{Y}_i - \boldsymbol{D}_i \boldsymbol{X}_i\|_F^2 + \gamma \sum_{i=1}^{C} \sum_{j \neq i} \tilde{b}(\boldsymbol{D}_i, \boldsymbol{D}_j) \tag{12}$$

where the new distance barrier function $\tilde{b}(\boldsymbol{D}_i, \boldsymbol{D}_j)$ is defined as

$$\begin{aligned} \sum_{(\boldsymbol{d}^{(a)}, \boldsymbol{d}^{(p)}, \boldsymbol{d}^{(n)})} \Big[& \max(0, M + \|\boldsymbol{d}^{(a)} - \boldsymbol{d}^{(p)}\|^2 - \|\boldsymbol{d}^{(a)} - \boldsymbol{d}^{(n)}\|^2) \\ & + \max(0, M + \|\boldsymbol{d}^{(a)} - \boldsymbol{d}^{(p)}\|^2 - \|\boldsymbol{d}^{(a)} + \boldsymbol{d}^{(n)}\|^2) \\ & + \max(0, M + \|\boldsymbol{d}^{(a)} + \boldsymbol{d}^{(p)}\|^2 - \|\boldsymbol{d}^{(a)} - \boldsymbol{d}^{(n)}\|^2) \\ & + \max(0, M + \|\boldsymbol{d}^{(a)} + \boldsymbol{d}^{(p)}\|^2 - \|\boldsymbol{d}^{(a)} + \boldsymbol{d}^{(n)}\|^2) \Big]. \end{aligned}$$

The sum goes over all triplets of atoms $(\boldsymbol{d}^{(a)}, \boldsymbol{d}^{(p)}, \boldsymbol{d}^{(n)})$ with $\boldsymbol{d}^{(a)}, \boldsymbol{d}^{(p)} \in \boldsymbol{D}_i$ and $\boldsymbol{d}^{(n)} \in \boldsymbol{D}_j$. The distance margin M ensures a minimum distance between a positive pair of atoms ($\boldsymbol{d}^{(a)}$, $\boldsymbol{d}^{(p)}$) (an anchor and a positive atom) and a negative pair ($\boldsymbol{d}^{(a)}$, $\boldsymbol{d}^{(n)}$) (the anchor and a negative atom). In the context of an atom update, we consider the updated atom as the anchor and any other atom from the same dictionary class as a positive atom. In contrast, any other atoms from different dictionary classes are considered negative atoms. Note that this time we do not use the quadratic term used before in IDB. Since we use triplets of atoms to build the distance barrier, we name it the Incoherent Triplet Distance Barrier (ITDB); by extension, we also name (12) ITDB. By following this strategy, we intend to obtain a more compressed structure of atoms in the same dictionary. In contrast, the rest of the atoms of different classes should be far away, ideally at distances larger than the margin M. Notice that the optimization of each atom is made similarly to IDB. The gradient of the first norm inside the $max(\cdot)$ function ensures that the optimization is made in the direction that brings positive atoms closer together; the second norm ensures that the direction that brings closer negative pairs of atoms is neglected.

Compared to the IDB method, the ITDB problem implies expensive computations for the distance barrier function since it uses triplets of atoms instead of pairs of atoms. To overcome this bottleneck, we propose two solutions that use a small percentage of the possible triplets. A natural way to do that is to use a portion of the negative atoms to form triplets. Since the selection is made randomly, we might not use the closest negative atoms to the current anchor. However, we expect all the negative atoms also to have a compressed representation; few of them may be representative of their general direction. The second solution is to compute all triplet

Dataset name	# Samples	# Dim	# Classes
YaleB	2414	504	38
AR	2600	540	100
CMU PIE	11554	256	68
15 Scene	4485	3000	15
Caltech101	9144	3000	102
HMDB51	6766	5000	51
UCF50	6680	5000	50

Table 1. Dataset summary

distances $M + \|\boldsymbol{d}^{(a)} \pm \boldsymbol{d}^{(p)}\|^2 - \|\boldsymbol{d}^{(a)} \pm \boldsymbol{d}^{(n)}\|^2$ and use a percentage of the pairs that have the highest values. In our implementation, we have chosen to select a percentage of the negative atoms to form triplets.

Remark 2. *To better understand the role of the barrier function $\tilde{b}(\boldsymbol{D}_i, \boldsymbol{D}_j)$ and its soft margin, we expand the first term of the sum, for a single triplet $(\boldsymbol{d}^{(a)}, \boldsymbol{d}^{(p)}, \boldsymbol{d}^{(n)})$ of atoms. We consider here that the distance barrier is not respected; hence we drop off the max function.*

$$\begin{aligned} & M + \|\boldsymbol{d}^{(a)} - \boldsymbol{d}^{(p)}\|^2 - \|\boldsymbol{d}^{(a)} - \boldsymbol{d}^{(n)}\|^2 \\ & = M + (2 - 2(\boldsymbol{d}^{(a)})^\top \boldsymbol{d}^{(p)}) - (2 - 2(\boldsymbol{d}^{(a)})^\top \boldsymbol{d}^{(n)}) \\ & = M + 2(\boldsymbol{d}^{(a)})^\top (\boldsymbol{d}^{(n)} - \boldsymbol{d}^{(p)}) \end{aligned} \tag{13}$$

We can see that the soft margin determines a coherence boundary between the anchor atom and the direction that brings closer atoms of the same class but does not bring the anchor closer to the negative atoms.

4. EXPERIMENTS

In this section, we present the experiment details alongside the obtained results. For our tests, we used seven datasets. We employed our methods in classification tasks, such as face recognition (*Yale B Face*, *AR Face*, *CMU PIE Face*), scene category recognition (*15 Scene*), object recognition (*Caltech101*) and action recognition (*UCF50 action*, *HMDB51 action*). The datasets summary is shown in Table 1.

The *YaleB Face* dataset (Extended Yale Face Database B) is a facial recognition dataset with 2.414 grayscale images of 38 different persons. All the registrations are obtained under different lighting conditions and poses. *AR Face* is a more extensive dataset containing facial images. This dataset consists of 4.000 color images of 126 individuals (70 men and 56 women). Each person has 26 distinct images with various expressions, light conditions, and occlusions. For the experiments, we followed the same procedure in [16] and used only 2.600 of available registrations. The last facial image dataset is *CMU PIE Face* (Carnegie Mellon University Pose, Illumination, and Expression). This dataset contains 41.368 images of 68 subjects. Each individual is captured under 13 different poses, 43 illumination conditions, and 4 facial expressions. The three datasets were chosen to test the performance and robustness of our methods underneath various poses, illuminations, and expression conditions. We used random face features for the face recognition tasks provided in [23].

15 Scene is a dataset designed for scene recognition problems. This dataset contains 4.485 images from 15 categories. The images have different sizes and resolutions. The *Caltech101* dataset is a well-known dataset used in object recognition problems. It consists of 9.144 images of 101 distinct objects and an additional background class. Each category contains between 40 and 800 images of different sizes, poses, and light conditions. For both datasets, scenes, and objects, we used the spatial pyramid matching (SPM) features [13], [24]. The image features are obtained by concatenating multiple histograms of bags of feature representations from four different pyramid levels. The final obtained features are reduced to 3.000 for both datasets using PCA.

The last used datasets are *HMDB51* and *UCF50* action bank, which are video datasets designed for action types recognition. The *HMDB51* dataset consists of 6.766 registrations of actions corresponding to 51 different human activities. The *UCF50* dataset contains 6.680 registrations from 50 different action categories. We used the action bank features for the classification problem, available in [15]. The action bank features were reduced to 5.000 by PCA.

All the experiments were performed on a Desktop PC with Ubuntu 20.04 as the operating system. The available hardware resources include an Intel i9 processor with 36 cores, 256 GB RAM, and an NVIDIA RTX3090 video card. The implementations were written in Matlab and are available on our research project website. We measured the performance of all methods in terms of accuracy, training time, and testing time. The results are obtained over 10 independent rounds with different initializations and dataset split seeds. We compared the IDB and ITDB methods with the standard IDL problem and three additional competitors, Projective Dictionary Pair Learning (DPL) [16], Self-Expressive Locality-Adaptive Latent Dictionary Pair Learning (SLatDPL) [17], and Low-rank Shared Dictionary Learning (LRSDL) [18]. For the DPL[1], SLatDPL[2] and LRSDL[3] methods, we used the original code provided by the authors. We followed the experimental details provided in [16]. For the proposed experiments, we split each dataset for training and testing. Depending on the number of available samples per class, we used a fixed number of samples for training, while the rest were used for tests. For the training stage, we used 32 samples per class for the *YaleB* dataset and 20 samples for the *AR Face* dataset, while for the rest of the datasets, only 30 training samples were used.

For datasets used in the DPL paper, we used the hyperparameters provided by the authors. For the rest of the datasets,

[1]http://www4.comp.polyu.edu.hk/ cslzhang/code/DPL_NIPS14.zip
[2]https://github.com/Daitu/SLatDPL
[3]https://github.com/tiepvupsu/DICTOL

	IDL	IDB	ITDB	DPL	SLatDPL	LRSDL
YaleB	94.28	94.35	94.47	97.65	97.25	91.58
AR	93.13	93.07	93.92	98.43	98.12	96.83
CMU PIE	90.56	90.60	91.98	93.25	91.24	90.41
15 Scene	95.62	94.17	92.39	95.89	94.57	73.07
Caltech101	69.16	72.18	70.42	70.72	62.89	73.61
HMDB51	29.74	29.89	30.81	25.57	19.40	18.40
UCF50	59.94	60.14	60.63	59.87	55.77	60.15

Table 2. Accuracy results (percentages)

	IDL	IDB	ITDB	DPL	SLatDPL	LRSDL
YaleB	13.97	11.20	81.35	1.35	6.59	36.33
	31.46	32.24	31.63	0.28	0.06	0.81
AR	42.70	127.40	1361.58	3.50	21.36	180.13
	41.24	42.27	41.78	0.33	0.15	1.61
CMU PIE	23.30	17.20	56.02	1.44	4.37	439.09
	420.62	419.93	420.81	0.50	1.25	69.16
15 Scene	9.14	22.15	28.192	5.27	200.84	0.22
	43.13	43.12	42.62	0.54	3.64	2.63
Caltech101	803.31	883.59	2544.06	40.80	1610.80	337.51
	474.96	471.19	467.60	3.29	71.58	78.93
HMDB51	273.60	93.13	1038.41	59.12	1958.49	309.73
	240.77	241.53	240.05	2.31	27.17	32.07
UCF50	328.36	100.27	1066.39	55.98	1955.90	307.16
	225.09	223.93	222.07	2.16	26.10	31.12

Table 3. Execution times of the algorithms

we performed an additional hyperparameter search and used $\tau = 0.05$ and $\lambda = 0.005$ (and $\lambda = 0.05$ for the HMDB51 dataset). For the SLatDPL method, since the hyperparameters are not provided in [17], we computed a custom hyperparameters search on our own. We fixed $\gamma = 0.5$ and proceed with a grid search with the α and β parameters over $\{0.005, 0.05, 0.5, 1, 5, 50, 5000\}$. The dictionary sizes and the number of iterations were taken as in the original papers. For the LRSDL method, we followed the instruction provided in the paper for three of the used datasets. For the rest of them, we used 20 atoms per class dictionary and 40 shared atoms; the three λ parameters were decided based on a hyperparameter tuning procedure over $\{0.005, 0.05, 0.5, 5, 50, 500, 5000\}$.

For the IDL, IDB, and ITDB methods, we only used dictionaries of size $n = 40$ with a sparsity constraint of $s = 20$. All the training stages were performed over 10 iterations. The experiments show that 10 iterations are enough for the algorithms to converge. For the hyperparameter tuning process, we conducted experiments using a grid search strategy; the evaluated margins are $M \in \{1.2, 1.4, 1.6, 1.8\}$, for the γ and λ parameters, we used combinations of values from $\{0.005, 0.05, 0.5, 5, 50, 500, 5000\}$. For the ITDB method tests, we only used $perc = 5\%$ and $perc = 10\%$ (depending on the number of training samples) of negative atoms to construct atom triplets. We provide the used hyperparameters in Table 4.

We summarize the main results in Table 2, where we present the obtained accuracies over all the datasets. Moreover, Table 3 gives the measured times over the training and testing stage. The results show the good behavior of the proposed methods. The IDB and ITDB methods generally overperform the accuracy of the IDL problem, but also that of the DPL, SLatDPL, and LRSDL methods for several datasets (Caltech101, CMU PIE, HMDB51, UCF50 for SLatDPL; the last two for DPL; the last three for LRSDL). ITDB is better than IDB in all databases but 15 Scene and Caltech101. However, a disadvantage of the proposed methods is the running time, which is larger. In training, IDB is clearly faster than ITDB, sometimes up to 10 times; however, testing times are similar because the testing procedure (3) is identical for the two methods.

5. CONCLUSIONS

This paper proposes two new formulations for the Incoherent Dictionary Learning (IDL) problem. We introduce two distance barrier functions that ensure incoherence between dictionaries from different classes. The results demonstrate good behavior of our methods compared with the standard IDL method and three other state-of-the-art methods, over several datasets.

6. REFERENCES

[1] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar conference on signals, systems and computers*. IEEE, 1993, pp. 40–44.

[2] B. Dumitrescu and P. Irofti, *Dictionary learning algorithms and applications*, Springer, 2018.

[3] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Tech. Rep., Computer Science Department, Technion, 2008.

[4] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[5] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2008.

[6] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *2010 IEEE Computer society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2691–2698.

Datasets	IDL	IDB			ITDB		DPL		SLatDPL			LRSDL		
	γ	M	γ	λ	M	γ	τ	γ	α	β	γ	λ_1	λ_2	λ_3
YaleB	0.005	1.4	0.05	500	1.8	500	0.05	0.003	0.005	0.005	0.5	0.5	0.005	5000
AR	5000	1.2	5000	0.5	1.4	500	0.05	0.005	0.005	0.05	0.5	500	0.005	0.05
CMU PIE	5	1.4	0.5	500	1.4	500	0.05	0.005	0.005	0.05	0.5	50	0.005	0.5
15 Scene	0.005	1.4	0.005	0.005	1.4	0.005	0.05	0.005	0.5	0.005	0.5	0.05	500	0.005
Caltech101	0.005	1.2	0.005	0.5	1.4	0.005	0.05	0.0001	0.005	0.005	0.5	0.005	50	5000
HMDB51	50	1.2	500	5000	1.6	5000	0.05	0.05	0.005	0.5	0.5	50	0.005	0.5
UCF50	5	1.2	500	500	1.6	5000	0.01	0.01	0.5	0.5	0.5	50	0.05	5000

Table 4. Algorithms hyperparameters

[7] Z. Jiang, Z. Lin, and L.S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[8] A. Seghouane and A. Iqbal, "Consistent adaptive sequential dictionary learning," *Signal Processing*, vol. 153, pp. 300–310, 2018.

[9] S. Ubaru, A. Seghouane, and Y. Saad, "Improving the incoherence of a learned dictionary via rank shrinkage," *Neural computation*, vol. 29, no. 1, pp. 263–285, 2016.

[10] A.S. Georghiades, P.N. Belhumeur, and D.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*, vol. 23, no. 6, pp. 643–660, 2001.

[11] A.M. Martinez, "The AR face database," *CVC Technical Report24*, 1998.

[12] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression (pie) database," in *Proceedings of fifth IEEE international conference on automatic face gesture recognition*. IEEE, 2002, pp. 53–58.

[13] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. IEEE, 2006, vol. 2, pp. 2169–2178.

[14] L. Fei-Fei, R. Fergus, and P. 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*. IEEE, 2004, pp. 178–178.

[15] S. Sadanand and J.J. Corso, "Action bank: A high-level representation of activity in video," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1234–1241.

[16] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Projective dictionary pair learning for pattern classification," *Advances in neural information processing systems*, vol. 27, 2014.

[17] Z. Zhang, Y. Sun, Y. Wang, Z. Zhang, H. Zhang, G. Liu, and M. Wang, "Twin-incoherent self-expressive locality-adaptive latent dictionary pair learning for classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 947–961, 2020.

[18] T.H. Vu and V. Monga, "Fast low-rank shared dictionary learning for image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5160–5175, 2017.

[19] I. Ramirez, P. Sprechmann, and G. 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*. IEEE, 2010, pp. 3501–3508.

[20] D.C. Ilie-Ablachim and B. Dumitrescu, "Incoherent frames design and dictionary learning using a distance barrier," *Signal Processing*, vol. 209, pp. 109019, 2023.

[21] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, 2006, vol. 2, pp. 1735–1742.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[23] Z. Jiang, Z. Lin, and L.S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[24] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *2009 IEEE Conference on computer vision and pattern recognition*. IEEE, 2009, pp. 1794–1801.