Analysis Dictionary Learning based Classification: Structure for Robustness

Wen Tang, Ashkan Panahi, Hamid Krim, and Liyi Dai

Abstract—A discriminative structured analysis dictionary is proposed for the classification task. A structure of the union of subspaces (UoS) is integrated into the conventional analysis dictionary learning to enhance the capability of discrimination. A simple classifier is also simultaneously included into the formulated functional to ensure a more complete consistent classification. The solution of the algorithm is efficiently obtained by the linearized alternating direction method of multipliers. Moreover, a distributed structured analysis dictionary learning is also presented to address large scale datasets. It can group-(class-) independently train the structured analysis dictionaries by different machines/cores/threads, and therefore avoid a high computational cost. A consensus structured analysis dictionary and a global classifier are jointly learned in the distributed approach to safeguard the discriminative power and the efficiency of classification. Experiments demonstrate that our method achieves a comparable or better performance than the state-of-the-art algorithms in a variety of visual classification tasks. In addition, the training and testing computational complexity are also greatly reduced.

Index Terms—Discriminate analysis dictionary learning, distributed analysis dictionary learning, structured mapping, supervised learning.

1 Introduction

Sparse representation has had of great success in dealing with various problems in image processing and computer vision, such as image denoising and image restoration. To obtain such sparse representations with an unknown precise model, Dictionary Learning is one choice because it results in a linear combination of sparse dictionary atoms. There are two different types of dictionary learning methods: Synthesis Dictionary Learning (SDL) and Analysis Dictionary Learning (ADL).

In recent years, SDL has been prevalently and widely studied [1], [2], while ADL has received little attention. SDL supposes that a signal lies in a sparse latent subspace and can be recovered by an associated dictionary. The local structures of the signal are well preserved in the optimal synthesis dictionary [3], [4], [5]. In contrast, ADL assumes that a signal can be transformed into a latent sparse subspace by its corresponding dictionary. In other words, ADL is to produce a sparse representation by applying the dictionary as a transform to a signal. The atoms in an analysis dictionary can be interpreted as local filters, as first mentioned in [6]. Sparse representations can be simply obtained by an inner product operation, when the dictionary is known. Such a fast coding supports ADL more favored than SDL in applications. The contrast of SDL and ADL is shown in Fig. 1.

The success of dictionary learning in image processing problems has shaped much interest in task-driven dictionary learning methods for inference applications, such as image classification. The task of classification aims to assign the correct label to an observed image, which requires a much more discriminative capacity of either the dictionary

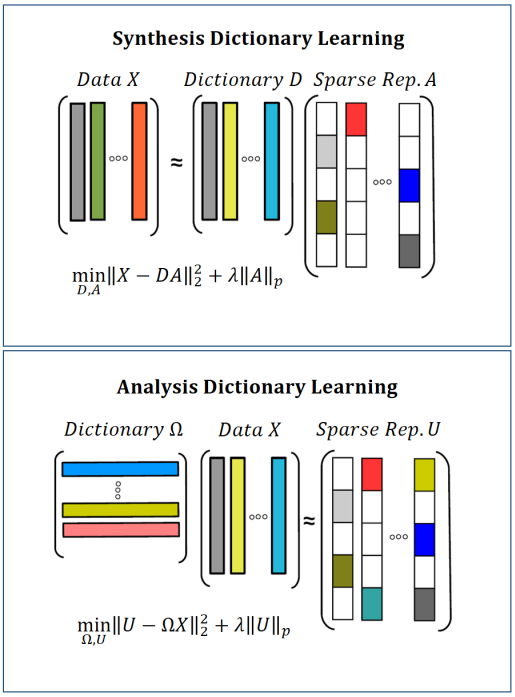

Fig. 1. SDL reconstructs data X by the dictionary D with the sparse representations A. ADL applies the dictionary Ω to data X and results the sparse representations U. $\|\cdot\|_p$ can be either l_1 norm or l_0 norm. If and only if D and Ω are square matrices, SDL and ADL are equivalent to each other.

- *W. Tang, A. Panahi and H. Krim are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606.*
  E-mail: {wtang6, apanahi, ahk}@ncsu.edu
- *L. Dai is with the US Army Research Office, Durham, NC 27703.*
  E-mail:liyi.dai@us.army.mil

This manuscript has been submitted to IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE on July 09 , 2018.

or the sparse representation. Towards addressing this issue, supervised learning is often invoked when using SDL so as to maximize the distances between the sparse representations of each two distinct classes.

There are generally two strategies to address the supervised learning approaches. The first strategy is to learn multiple dictionaries or class-specific dictionaries for different classes [7], [8], [9], [10]. The advantage of learning multiple dictionaries is that these dictionaries characterize specific patterns and structures of each class and enhance the distances between different classes. The minimum reconstruction errors of various dictionaries are subsequently used to assign labels of new incoming images. In [8], Ramirez *et al.* learned class-specific dictionaries with penalty for the common atoms. Yang *et al.* [9] then learned class-specific dictionaries and jointly applied a Fisher criterion to associative sparse representations to thereby enhance the distances between each class. A large-margin method was proposed to increase the divergence of sparse representations for the class-specific dictionaries in [10]. However, as the number of classes increases, it would be too complex and time consuming to train class-specific dictionaries with regularizing distances of each dictionary. Even though a distributed cluster could reduce the time complexity of training dictionaries, it is difficult for the distributed algorithm to communicate with each independent cluster and to compromise with other regularizations for the class-specific dictionary learning.

Another strategy is to learn a shared dictionary for all classes together with a universal classifier [11], [12]. Such a joint dictionary learning enforces more discriminative sparse representations. Compared with class-specific dictionary learning, using this strategy is simpler to learn such a dictionary and classifier, and easier to test the unknown images. In [11], Mairal *et al.* integrated a linear classifier in a sparse representation for a dictionary learning phase. Jiang *et al.* then included a linear classifier and a label consistent regularization term to enforce more consistent of sparse representations in each class [12]. When any large data sets are on hand, memory and computational limitations emerge, and an online learning or distributed solutions are required as a viable strategy.

Although the techniques mentioned above are all based on SDL, ADL has gradually received more attention [13]. To the best of our knowledge, few attempts have been carried out on the task-driven ADL. Both of the analysis K-SVD [14] and the Sparse Null Space (SNS) pursuit [15] have only proposed a solution of learning an analysis dictionary. In [16], Shekhar *et al.* learned an analysis dictionary and then trained SVM for the digital and face recognitions. Their results demonstrated that ADL is more stable than SDL under noise and occlusion, and achieved a competitive performance. Guo *et al.* [17] integrated local topological structures and discriminative sparse labels into the ADL and separately classified images by a k Nearest Neighbor classifier.

Inspired by these past works, and taking advantage of efficient coding by ADL, we propose a supervised ADL with a shared dictionary and a universal classifier. In addition to the classifier, a structured subspace regularization is also included into an ADL model to obtain a more structured discriminative and efficient approach to image classification. We refer to this approach as Structured Analysis Dictionary Learning (SADL). Since Sparse Subspace Clustering [18] has shown that visual data in a class or category can be well captured and localized by a low dimensional subspace, and the sparse representation of the data within a class similarly share a low dimensional subspace, a structured representation is introduced to achieve a distinct representation of each class. This achieves more coherence for within-class sparse representations and more disparity for between-class representations. When sorted by the order of classes, these representations as shown later can be viewed as a block-diagonal matrix. For robustness of the sought sparse representations, we simultaneously learn a one-against-all regression-based classifier. The resulting optimization functional is solved by a linearized alternative direction method (ADM) [19]. This approach leads to a more computationally efficient solution than that of analysis K-SVD [14] and of SNS pursuit [15]. Additionally, a great advantage of our algorithm is its extremely short on-line encoding and classification time for an incoming observed image. It is easy to understand that in contrast to the SDL encoding procedure, ADL obtains a sparse representation by a simple matrix multiplication of the learned dictionary and testing data. Experiments demonstrate that our method achieves an overall better performance than the synthesis dictionary approach. A good accuracy is achieved in the scene and object classification with a simple classifier, and at a remarkably low computational complexity to seek the best performances of facial recognition problems. Moreover, the experiments also shows that our approach has a more stable performance than that of SDL. Even when the dictionary size is reduced to result in memory demand reduction, our performance is still outstanding. To address large datasets, a distributed structured analysis dictionary learning algorithm is also developed while preserving the same properties as those of structured analysis dictionary learning (SADL). Experiments also show that when the dataset is sufficient, a distributed algorithm achieves as high a performance as SADL.

The following represent our main contributions,

- Both a structured representation and a classification error regularization term are introduced in to the conventional ADL formulation to improve classification results. A multiclass classifier and an analysis dictionary are jointly learned.
- The optimal solution provided by the linearized ADM is significantly faster than other existing techniques for non-convex and non-smooth optimization.
- An extremely short classification time is offered by our algorithms, as it entails encoding by a mere matrix multiplication for a simple classification procedure.
- A distributed structured analysis dictionary learning algorithm is also presented.

The balance of this paper is organized as follows: we state and formulate the problem of SADL and its distributed form in Section 2. The resulting solutions to the optimization problems along with the classification procedure are de-

scribed in Sections 3 and 4. In Section 5, we analyze the convergence and complexity of our methods. The experimental comprehensively validation and results are then presented in Section 6. Some comments and future works are finally provided in Section 7.

2 Structured Analysis Dictionary Learning

2.1 Notation

Uppercase and lowercase letters respectively denote matrix and vectors throughout the paper. The transpose and inverse of matrix are represented as the superscripts T and -1, such as A^T and A^{-1}. The identity matrix and all-zero matrix are respectively denoted as I and $\mathbf{0}$. $(a_i)_j$ represents the jth element in the ith column of matrix A.

2.2 Structured ADL Method

2.2.1 ADL Formulation

The conventional ADL problem [14] aims at obtaining a representation frame Ω with a sparse coefficient set U based on the data matrix $X = [x_1, \ldots, x_n] \in \mathbb{R}^{m\times n}$.

$$\begin{aligned} \arg\min_{\Omega,U} \frac{1}{2}\|U - \Omega X\|_2^2 + \lambda_1\|U\|_1 \\ s.t.\ \Omega \in \mathbb{R}^{r\times m} \subset \mathcal{W}, \end{aligned} \tag{1}$$

where $U \in \mathbb{R}^{r\times n}$ and $\mathcal{W}$ is a large class of non-trivial solutions.

2.2.2 Mitigating Inter-Class Feature Interference

The basic idea of our algorithm is to take advantage of the stability to perturbations and of the fast encoding of ADL. Since there is no reconstruction term in the conventional ADL, and to secure an efficient classification, the representation U is used to obtain a classifier in a supervised learning mode. To strengthen the discriminative power of ADL, it is better to minimize the impacts of inter-class common features. We therefore propose two additional constraints on U by way of:

- Minimizing interference of intra-class common features by a structural map of U.
- Minimizing the classification error.

2.2.2.1 Structural Mapping of U: The first constraint is to particularly ensure that the representation of each sample in the same class belong to a subspace defined by a span of the associated coefficients. This imposes the distinction among the classes and improves the identification of each class, and efficiently enhances the divergence between classes. Specifically, we introduce a block-diagonal matrix $H \in \mathbb{R}^{s\times n}$ as shown below,

$$H = \begin{array}{c} \begin{matrix} h_1^1 & h_2^1 & h_3^1 & h_4^2 & h_5^2 & h_6^3 & h_7^3 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array},$$

where $s \geq n$ is the length of the structured representation. Each diagonal block in H represents a subspace of each class to force each one class to remain distinct from another with a consistent intra-class representation. Each column h_i^j is a structured representation for the corresponding data point, which is pre-defined on the training labels. H is not necessarily a uniformly block-diagonal matrix, and the order of samples is not important, so long as the structured representation corresponds to a given class. To mildly relax the constraint, and integrate it into the ADL functional, we write

$$H = QU + \varepsilon_1, \tag{2}$$

where $Q \in \mathbb{R}^{s\times r}$ is a matrix to be learned with Ω and U, ε_1 is the tolerance.

2.2.2.2 Minimal Classification Error: To maintain an audit track on the desired representation, we include a classification error to make the representation QU discriminative and learn an optimal regularization. This is written as

$$Y = W(QU) + \varepsilon_2, \tag{3}$$

where ε_2 is the tolerance, $W \in \mathbb{R}^{c\times s}$ is a linear transform, and the label matrix $Y \in \mathbb{R}^{c\times n}$ is defined as

$$Y_{ij} = \begin{cases} 1 & \text{if image } j \text{ belongs to class } i \\ 0 & \text{otherwise} \end{cases},$$

and c is the number of classes.

2.2.3 Structured ADL Formulation

To account for all these constraints and to avoid overfitting by l_2 regularization arising Q and W, we can rewrite the one, all ADL optimization problem as

$$\begin{aligned} \arg\min_{\substack{\Omega,U,Q,W,\\ \varepsilon_1,\varepsilon_2}} & \frac{1}{2}\|U - \Omega X\|_F^2 + \lambda_1\|U\|_1 \\ & + \frac{\rho_1}{2}\|\varepsilon_1\|_2^2 + \frac{\rho_2}{2}\|\varepsilon_2\|_2^2 \\ & + \frac{\delta_1}{2}\|Q\|_2^2 + \frac{\delta_2}{2}\|W\|_2^2 \\ s.t.\ & H = QU + \varepsilon_1, \\ & Y = W(QU) + \varepsilon_2, \end{aligned} \tag{4}$$

where ω_i^T is the row of Ω, and $\rho_1, \rho_2, \delta_1, \delta_2$ are the penalty coefficients. Recall H is the structured representation, Q is the structuring transformation, Y is the classifier label, W is the linear classifier, and λ_1 is a tuning parameter.

The formulated optimization functional in Eq.(4) provides an analysis dictionary driven by the latent structure of the data yielding an improved discriminative sparse representation among numerous classes.

2.2.4 Distributed Structured ADL Formulation

In order to handle large datasets, we propose a distributed Structured ADL method. Since both the discriminative structure and the efficient classification need to be preserved, we introduce a global analysis dictionary, a global structuring transformation and a global classifier. In pursuing a distributed ADL, we ensure that the global variables share information with each distributed dictionary cluster, thereby ensuring that the global analysis dictionary, the

structured transform and the classifier respectively reach a consensus,

$$\|\Omega-\Omega_t\|^2, \|Q-Q_t\|^2, \|W-W_t\|^2, \forall t=1,\ldots,N. \quad (5)$$

Together with the consensus penalties, the distributed SADL is formulated as

$$\begin{aligned}\arg\min_{\substack{\Omega_t,U_t,\\ Q_t,W_t,\\ \Omega,Q,W,\\ \varepsilon_{1_t},\varepsilon_{2_t}}} &\sum_{t=1}^{N}(\frac{1}{2}\|U_t-\Omega_t X_t\|_F^2+\lambda_1\|U_t\|_1+\frac{\rho_{1_t}}{2}\|\varepsilon_{1_t}\|_2^2\\ &+\frac{\rho_{2_t}}{2}\|\varepsilon_{2_t}\|_2^2+\frac{\xi_{1_t}}{2}\|\Omega-\Omega_t\|_2^2+\frac{\delta_{1_t}}{2}\|Q_t\|_2^2\\ &+\frac{\xi_{2_t}}{2}\|Q-Q_t\|_2^2+\frac{\delta_{2_t}}{2}\|W_t\|_2^2+\frac{\xi_{3_t}}{2}\|W-W_t\|_2^2)\\ s.t.\ &H_t=Q_tU_t+\varepsilon_{1_t},\\ &Y_t=W_t(Q_tU_t)+\varepsilon_{2_t},\\ &\|\omega_i^T\|_2^2=1;\forall i=1,\ldots,r,\\ &\|\omega_{t_i}^T\|_2^2=1;\forall i=1,\ldots,r,\forall t=1,\ldots,N,\end{aligned} \quad (6)$$

where t represents the tth independent cluster, Ω_t, U_t, Q_t and W_t are respectively the local analysis dictionary, sparse representation, structuring transformation and classifier of the tth cluster, and Ω, Q, W are respectively the global analysis dictionary, structuring transformation and classifier. The global variables will be applied to the same efficient classification scheme as the one of SADL.

3 Algorithmic Solution

3.1 SADL Algorithm

Due to the non-convexity of the objective functional in Eq.(4), an augmented Lagrange formulation with dual variables $Z^{(1)}$, $Z^{(2)}$ and μ is adopted to seek an optimal solution. The augmented Lagrangian is then written as,

$$\begin{aligned}&L(\Omega,U,Q,W,Z^{(1)},Z^{(2)},\mu)=\frac{1}{2}\|U-\Omega X\|_F^2+\lambda_1\|U\|_1\\ &+\langle Z^{(1)},H-QU-\varepsilon_1\rangle+\langle Z^{(2)},Y-W(QU)-\varepsilon_2\rangle\\ &+\frac{\mu}{2}\|H-QU-\varepsilon_1\|_2^2+\frac{\mu}{2}\|Y-W(QU)-\varepsilon_2\|_2^2\\ &+\frac{\rho_1}{2}\|\varepsilon_1\|_2^2+\frac{\rho_2}{2}\|\varepsilon_2\|_2^2+\frac{\delta_1}{2}\|Q\|_2^2+\frac{\delta_2}{2}\|W\|_2^2,\end{aligned} \quad (7)$$

where $\lambda_1>0$ is a tuning parameter. To iteratively seek the optimal solution in Eq.(7), the analysis dictionary Ω and two linear transformations Q and W are first randomly initialized, when the sparse representation U is initialized by $U=\mathbf{0}$, the zero matrix. The auxiliary variables η_Q, η_{WQ}, and η_{WU} are introduced to guarantee the convergence of the algorithm. The variable with superscripts which do not include parenthesis is the temporal variable of intermediate step in the calculation. Different variables are alternately updated while fixing the others, resulting in the following steps:

(1) Fix Ω, Q, W, and ε_1, ε_2 update U

$$U_{k+1}=\tau_{\frac{\lambda_1}{\mu\eta_U}}(U_k-\frac{U_k^1+U_k^2+U_k^3}{\mu\eta_U}), \quad (8)$$

where $\tau(\cdot)$ is the element-wise soft thresholding operator, and U_k^1, U_k^2, and U_k^3 are as follows:

$$U_k^1=-(\Omega_kX-U_k), \quad (9)$$

$$U_k^2=-Q_k^T(Z_k^{(1)}+\mu(H-Q_kU_k-\varepsilon_{1_k})), \quad (10)$$

$$U_k^3=-Q_k^TW_k^T(Z_k^{(2)}+\mu(Y-W_kQ_kU_k-\varepsilon_{2_k})). \quad (11)$$

(2) Fix Ω, U W, and ε_1, ε_2 update Q

$$Q_{k+1}=Q_k-\frac{Q_k^1+Q_k^2}{\mu\eta_Q}, \quad (12)$$

$$Q_k^1=-(Z_k^{(1)}+\mu(H-Q_kU_{k+1}-\varepsilon_{1_k}))U_{k+1}^T+\delta_1Q_k, \quad (13)$$

$$Q_k^2=-W_k^T(Z_k^{(2)}+\mu(Y-W_kQ_kU_{k+1}-\varepsilon_{2_k}))U_{k+1}^T. \quad (14)$$

(3) Fix Ω, U Q, and ε_1, ε_2 update W

$$W_{k+1}=W_k-\frac{W_k^1}{\mu\eta_W} \quad (15)$$

$$W_k^1=-(Z_k^{(2)}+\mu(Y-W_kQ_{k+1}U_{k+1}-\varepsilon_{2_k}))U_{k+1}^TQ_{k+1}^T+\delta_2W_k. \quad (16)$$

(4) Fix U, Q W, and ε_1, ε_2 update Ω

$$\Omega_{k+1}^*=\arg\min_{\Omega}\frac{1}{2}\|U_{k+1}-\Omega X\|_F^2. \quad (17)$$

The analytical solution of Eq.(17) can be regularized as

$$\Omega_{k+1}=U_{k+1}X^T(XX^T+\lambda_4I)^{-1}, \quad (18)$$

where λ_4 is also a tuning parameter. It will be chosen by a usual way.

(5) Fix U, Ω, Q W, and ε_2 update ε_1

$$\varepsilon_{1_{k+1}}=\frac{1}{\rho_1-1}(Z_k^{(1)}+\mu(H-Q_{k+1}U_{K+1})). \quad (19)$$

(6) Fix U, Ω, Q W, and ε_1 update ε_2

$$\varepsilon_{2_{k+1}}=\frac{1}{\rho_2-1}(Z_k^{(2)}+\mu(Y-W_{k+1}Q_{k+1}U_{K+1})). \quad (20)$$

And then, the dual variable $Z^{(1)}$, $Z^{(2)}$ and μ are updated as

$$Z_{k+1}^{(1)}=Z_k^{(1)}+\mu(H-Q_{k+1}U_{k+1}), \quad (21)$$

$$Z_{k+1}^{(2)}=Z_k^{(2)}+\mu(Y-W_{k+1}Q_{k+1}U_{k+1}). \quad (22)$$

In contrast to previous ADL techniques, which train a dictionary by iterating a single row of the dictionary, *i.e.*, one atom, to avoid a trivial solution, we proceed to update a set of rows in a single step at each iteration. A fast convergence rate of the algorithm is also guaranteed by linearized ADM [19] and with a closed form solution for the dictionary Ω given in Eq.(18). The proposed SADL algorithm is summarized in Algorithm 1.

Algorithm 1 Structured Analysis Dictionary Learning

Input: Training data $X = [x_1, \ldots, x_n]$, diagonal block matrix H, classes labels Y, penalty coefficients $\rho_1, \rho_2, \delta_1, \delta_2$, parameter λ_1, λ_4 and maximum iteration T;

Output: Analysis dictionary Ω, sparse representation U, and linear transformations Q and W;

Initialize Ω, Q, and W as random matrices, and initialize U as a zero matrix;
while not converged **and** $k < T$ **do**
  k=k+1;
  Update U_k by (8);
  Update Q_k by (12);
  Update W_k by (15);
  Update Ω_k by (18);
  Update ε_{1_k} by (19);
  Update ε_{2_k} by (20);
  Update $Z_k^{(1)}$ by (21);
  Update $Z_k^{(2)}$ by (22);
end while

3.2 Distributed SADL Algorithm

The distributed SADL is similarly expressed in the augmented Lagrangian functional as

$$
\begin{aligned}
&L_d(\Omega_t, U_t, Q_t, W_t, \Omega, Q, W, Z^{(1)}, Z^{(2)}, \mu_k) = \\
&\sum_{t=1}^{N}(\frac{1}{2}\|U_t - \Omega_t X_t\|_F^2 + \lambda_1\|U_t\|_1 + \frac{\delta_{1_t}}{2}\|Q_t\|_2^2 + \frac{\delta_{2_t}}{2}\|W_t\|_2^2 \\
&+ \frac{\xi_{1_t}}{2}\|\Omega - \Omega_t\|_2^2 + \frac{\xi_{2_t}}{2}\|Q - Q_t\|_2^2 + \frac{\xi_{3_t}}{2}\|W - W_t\|_2^2 \\
&+ \frac{\rho_{1_t}}{2}\|\varepsilon_{1_t}\|_2^2 + \frac{\rho_{2_t}}{2}\|\varepsilon_{2_t}\|_2^2 \\
&+ \langle Z_t^{(1)}, H_t Q_t U_t - \varepsilon_{1_t}\rangle + \langle Z_t^{(2)}, Y_t - W_t(Q_t U_t) - \varepsilon_{2_t}\rangle \\
&+ \frac{\mu_k}{2}\|H_t - Q_t U_t - \varepsilon_{1_t}\|_2^2 + \frac{\mu_k}{2}\|Y_t - W_t(Q_t U_t) - \varepsilon_{2_t}\|_2^2).
\end{aligned} \tag{23}
$$

To minimize such an objective functional, each variable is alternatively updated while fixing others. The distributed SADL algorithm is presented in Algorithm 2.

4 Classification Procedure

Both SADL and Distributed SADL have the same classification procedure because the global analysis dictionary Ω, transforming matrix Q and classifier W are obtained from the algorithms. With the analysis dictionary Ω in hand, an observed image x can be quickly sparsely encoded as Ωx. This is in stark contrast to SDL for which a sparse representation is obtained by solving a non-smooth optimization as: $\arg\min_\alpha \|x - D\alpha\|_2^2 + \|\alpha\|_1$, and highlights the marked improvement ADL provides. Our proposed SADL, which naturally enjoys the same encoding properties as ADL, efficiently yields a structured sparse representation $Q(\Omega x)$ of the signal x as well. Figure 2 shows an example of the structured representations obtained from Scene 15 dataset.

As shown, the result reflects the desired block diagonal structure. The ultimate desired classification goal of x is accomplished by $W(Q\Omega x)$. Figure 3 depicts $W(Q\Omega x)$ for the example in Figure 2 where the horizontal axis is image index, and the vertical axis reflects the class labels, which are computed according to,

$$y = \max_j (WQ\Omega x)_j, \tag{24}$$

shown as the brightest ones in Figure 3.

Algorithm 2 Distributed SADL

Input: Training data $X = [x_1, \ldots, x_n]$, diagonal block matrix H, classes labels L, penalty coefficients $\delta_{1_t}, \delta_{2_t}, \xi_{1_t}, \xi_{2_t}, \xi_{3_t}$, parameter λ_1, λ_4 and maximum iteration T;

Output: Analysis dictionary Ω, linear transformations Q and W;

Initialize Ω_t, Q_t, W_t, Ω, Q, and W as random matrices, initialize U_t as a zero matrix, and set X_t as a random subset of X with $\bigcup_{t=1}^{N} X_t = X$;
while not converged **and** $k < T$ **do**
  $k = k + 1$;
  for t=1:N **do** %Here for loop can be parallelized or distributed in different clusters.
    $U_t^{k+1} = \tau_{\frac{\lambda_1}{\mu(\eta_Q+\eta_{WQ})}}\left(U_t^k - \frac{\nabla_U L_d(\Omega_t^k, U_t^k, Q_t^k, W_t^k, \Omega^k, Q^k, W^k, Y_t^{(1)^k}, Y_t^{(2)^k})}{\mu(\eta_Q+\eta_{WQ})}\right)$;
    $Q_t^{k+1} = Q_t^k - \frac{\nabla_Q L_d(\Omega_t^k, U_t^{k+1}, Q_t^k, W_t^k, \Omega^k, Q^k, W^k, Y_t^{(1)^k}, Y_t^{(2)^k})}{\mu(\eta_Q+\eta_{WU})}$;
    $W_t^{k+1} = W_t^k - \frac{\nabla_W L_d(\Omega_t^k, U_t^{k+1}, Q_t^{k+1}, W_t^k, \Omega^k, Q^k, W^k, Y_t^{(1)^k}, Y_t^{(2)^k})}{\mu\eta_{QU}}$;
    $\Omega_t^{k+1} = (U_t^{k+1}X_t^T + \xi_{1_t}\Omega^k)(X_t X_t^T + \xi_{1_t} I)^{-1}$;
    Normalize Ω_t^{k+1} by $\omega_{t_i}^T = \frac{\omega_{t_i}^T}{\|\omega_{t_i}^T\|_2}, \forall i$;
    $Y_{k+1}^{(1)} = Y_k^{(1)} + \mu(H - Q_{k+1}U_{k+1})$;
    $Y_{k+1}^{(2)} = Y_k^{(2)} + \mu(L - W_{k+1}Q_{k+1}U_{k+1})$;
    $\mu_{k+1} = \min\{\rho\mu, \mu_{max}\}$;
    $\xi_{1_{k+1}} = \min\{\rho\xi_{1_k}, \xi_{1_{max}}\}$;
    $\xi_{2_{k+1}} = \min\{\rho\xi_{2_k}, \xi_{2_{max}}\}$;
    $\xi_{3_{k+1}} = \min\{\rho\xi_{3_k}, \xi_{3_{max}}\}$;
  end for
  $\Omega^{k+1} = \frac{1}{N}\sum_t \Omega_t^{k+1}$;
  Normalize Ω^{k+1} by $\omega_i^T = \frac{\omega_i^T}{\|\omega_i^T\|_2}, \forall i$;
  $Q^{k+1} = \frac{1}{N}\sum_t Q_t^{k+1}$;
  $W^{k+1} = \frac{1}{N}\sum_t W_t^{k+1}$;
end while

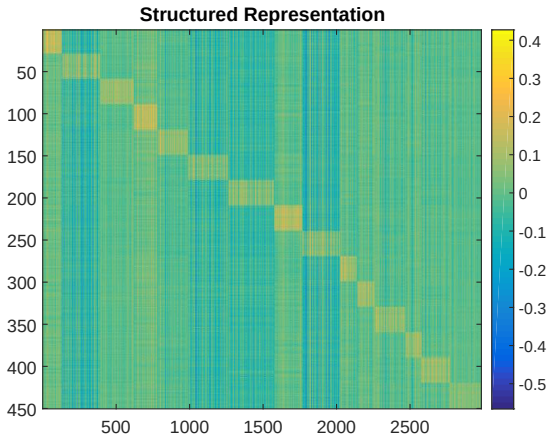

Fig. 2. $Q(\Omega x)$ on Scene 15 Dataset

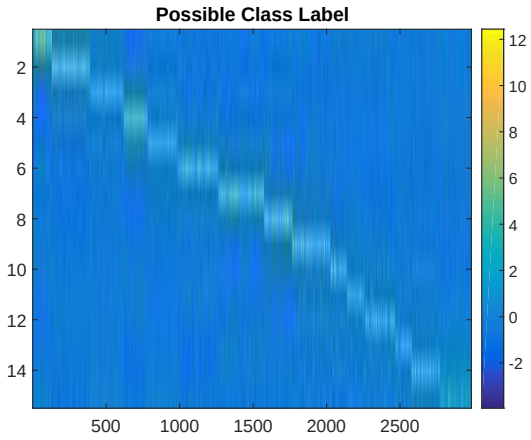

Fig. 3. $WQ(\Omega x)$ on Scene 15 Dataset

5 Convergence

Since we have used linearized ADM method to solve our nonconvex objective functional, η_U, η_Q, η_W are introduced as the auxiliary variables. We additionally have the following

Theorem 1. *Suppose that* $\mu \geq \sqrt{2}\{\rho_1, \rho_2\}$. *There exist positive values* $\eta_U^0, \eta_Q^0, \eta_W^0, R$ *only depending on the initialization such that for* $\eta_U > \eta_U^0, \eta_Q > \eta_Q^0, \eta_W > \eta_W^0$ *the sequence* $\{\Theta_k = (\Omega_k, U_k, Q_k, W_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\}_{k=1}^{\infty}$ *converges to the following set of bounded feasible stationary points of the Lagrangian* [1]*:*

$$S = \{\Theta = (\Omega, U, Q, W, \varepsilon^{(1)}, \varepsilon^{(2)}, Z^{(1)}, Z^{(2)}) \mid$$

$$\|\Theta\| < R, -\nabla L_s \in \lambda \partial \|U\|_1, H = QU + \varepsilon^{(1)}, Y = QUW + \varepsilon^{(2)}\}$$

where L_s *is the smooth part of* L*, i.e.*

$$L = L_s + \lambda_1 \|U\|_1.$$

According to the Theorem 1, if we initialize η_U, η_Q, η_W large enough, Algorithm 1 not only converges, but also generates the variable sequences with a final convergence to the stationary points. The proof of Theorem 1 can be found in the Appendix.

6 Experiments and Results

We now evaluate our proposed SADL method on five popular visual classification datasets which have been widely used in previous works and with known performance benchmarks. They include Extended YaleB [20] face dataset, AR [21] face dataset, Caltech101 [22] object categorization dataset, Caltech256 [23] objective dataset, and Scene15 [24] scene image dataset.

In our experiments, we provide a comparative evaluation of three state of the art techniques and our proposed technique, including a classification accuracy as well as training and testing times. All our experiments and competing algorithms are implemented in Matlab 2015b on the server with 2.30GHz Intel(R) Xeon(R) CPU. For a fair comparison, we measure the performance of each algorithm by repeating the experiment over 10 realizations. We evaluate the performances of all algorithms by using the same dictionary size. The testing time is defined as the average processing time to classify a single image. In our tables, the accuracy in parentheses with the associated citation is that was reported in the original paper. The difference in the accuracy of our approach and of the original one might be caused by different segmentations of the training and testing samples.

6.1 Parameter Settings

In our proposed SADL method, λ_1, λ_4 and maximum iteration T are tuning parameters. λ_1 controls the contribution of the sparsity, and the parameter λ_4 controls the learned analysis dictionary, while T is the maximum iteration number. We found that the result of setting $\varepsilon_1 = 0$ and $\varepsilon_2 = 0$ is almost the same as ones of setting penalty coefficients ρ_1 and ρ_2 to be 10^{10}, we let $\varepsilon_1 = 0$ and $\varepsilon_2 = 0$ in our experiment implementation. We choose for all the experiments λ_1, λ_4 and T by 10-fold cross validation on each dataset. In addition, we also optimally tuned the parameters of all competing methods to ensure their best performance.

6.2 State-of-the-art Methods

We compare our proposed SADL and Distributed SADL (DSADL) with these competing techniques: The first one is a baseline, which uses the ADL method to learn a sparse representation and subsequently trains a Support Vector Machine (SVM) to classify images based on such sparse representations (ADL+SVM) [16]. A penalty term is included to avoid similar atoms and minimize false positives. The second one is the classical Sparse Representation based Classification (SRC) [7]. For this method, we do not need to train a dictionary. Instead, we use the training images as the atoms in the dictionary. In the testing phase, we obtain the sparse coefficients based on such a dictionary. The third technique that we consider in this work is a state-of-the-art dictionary learning method, called Label Consistent K-SVD (LC-KSVD) [12], which forces each category labels to be consistent with classification. We select the LC-KSVD2 in [12] for comparison, because it has a better classification performance.

6.3 Extended YaleB

Fig. 4. Extended YaleB Dataset Examples

1. The norm $\|\Theta\|$ is any norm which is continuous with respect to the two norm of the components, for example their some of two norms. Also, the function $\|U\|_1$ is treated as a (convex) function of Θ, which is constant with respect to other components than U.

The Extended YaleB face dataset contains in total 2414 frontal face images of 38 persons under various illumination and expression conditions, as illustrated in Figure 4. Each person has about 64 images, each cropped to 168×192 pixels. We project each face image onto a n-dimensional random face feature vector. The projection is performed by a randomly generated matrix with a zero mean normal distribution whose rows are l_2 normalized. This procedure is similar to the one in [12]. In our experiment, n is 504, *i.e.,* each Extended YaleB face image is reduced to a 504-dimensional feature vector. Then, we randomly choose half of the images for training, and the rest for testing. The dictionary size is set to 570 atoms, $\lambda_1 = 0.001, \lambda_4 = 0.5$ and $T = 780$.

TABLE 1
Classification Results on Extended YaleB Dataset

Methods	Classification Accuracy(%)	Training Time(s)	Testing Time(s)
ADL+SVM [16]	82.91%	91.78	1.13×10^{-3}
SRC [7]	80.5%	No Need	3.74×10^{-1}
LC-KSVD [12]	94.56% (**95%** [12])	234.67	1.63×10^{-2}
SADL	94.91%	**51.29**	$\mathbf{2.72}\times10^{-6}$

The classification results, training and testing times are summarized in Table 1. Our proposed SADL method achieves the highest classification accuracy. Although the performance of our algorithm is superior by only a small factor, it is substantially more efficient than the others in terms of numerical complexity.

For a more thorough evaluation, we compare SADL with LC-KSVD for different dictionary sizes, and display the classification accuracy and training time in Figure 5 and 6. We ran our experiments for dictionary sizes by 32, 128, 224, 320, 416, 512, 608, 704, 800, 896, 992, and 1216 (all training size). SADL exhibits a more stable accuracy performance than that of LC-KSVD. In particular, the accuracy of LC-KSVD significantly decreases, when the dictionary size approaches the all training sample size. In addition, our method apparently has a much higher classification accuracy than LC-KSVD, when the dictionary size is small. The significant decrease in accuracy may be caused by the trivial solution of dictionary D in SDL. Moreover, in the training phase, the SADL method is also much faster than the LC-KSVD.

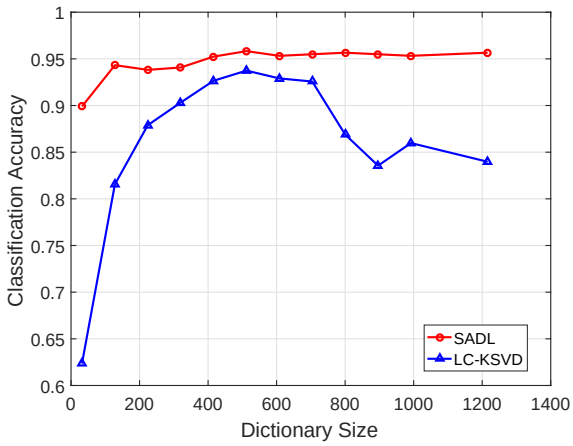

Fig. 5. Classification Accuracy versus Dictionary Size

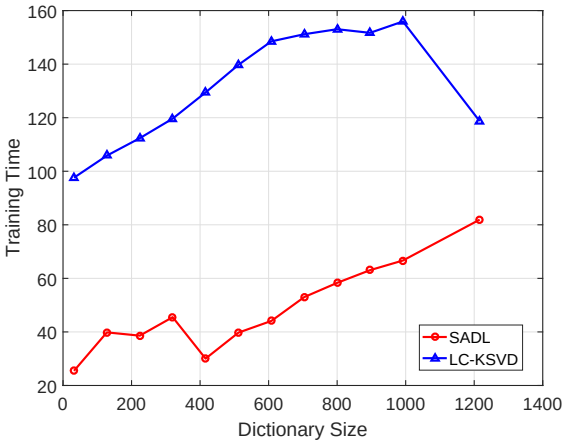

Fig. 6. Training Time versus Dictionary Size

The classification results as well as the training and testing times are summarized in Table 2. Our proposed SADL achieves higher classification accuracy than others. Our method is about 10000 times faster than SRC and LC-KSVD for the testing phase.

6.4 AR Face

The AR Face dateset has 2600 color images of 50 females and 50 males with more facial variations than the Extended YaleB database, such as different illumination conditions, expressions and facial disguises. Each person has about 26 images of size 165×120. Figure 7 shows some sample images of faces with sunglasses or scarves. The features of the AR face image are extracted in the same way as those of the Extended YaleB face image are, but we project it to a 540 dimensional feature vector similarly to the setting in [12]. 20 images of each person are randomly selected as a training set and the other 6 images for testing. The dictionary size of the AR dataset is set to 500 atoms, $\lambda_1 = 0.001$, $\lambda_4 = 0.5$ and $T = 1040$.

Fig. 7. AR Dataset Examples

TABLE 2
Classification Results on AR Dataset

Methods	Classification Accuracy(%)	Training Time(s)	Testing Time(s)
ADL+SVM [16]	90.40%	218.54	9.10×10^{-3}
SRC [7]	66.50%	No Need	5.25×10^{-2}
LC-KSVD [12]	87.78% (93.7% [12])	244.52	1.42×10^{-2}
SADL	**95.08%**	**89.13**	$\mathbf{3.67}\times10^{-6}$

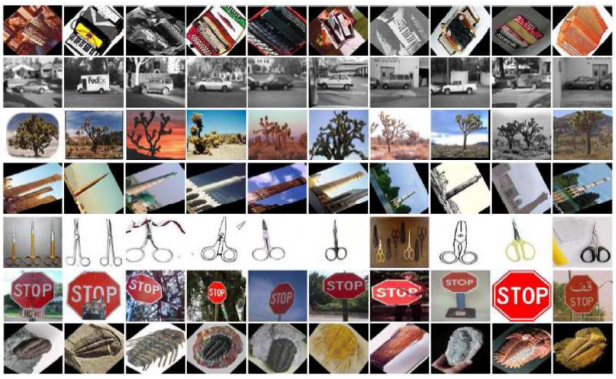

Fig. 8. Caltech101 Dataset Examples

6.5 Caltech101

The Caltech101 dataset has 101 different categories of different objects and one non-object category. Most categories have around 50 images. Figure 8 gives some examples from the Caltech101 dataset. We extract dense Scale-invariant Feature Transform(SIFT) descriptors for each image from 16×16 patches and with a 6 pixels step. Then, we apply a spatial pyramid method [24] to the dense SIFT features with three segmentation sizes 1×1, 2×2, and 4×4 to capture the objects' features at different scales. At the same time, a 1024 size codebook is trained by k-means clustering for spatial pyramid features. Spatial pyramid features of each subregion are then concatenated together as a vector to represent one image. Due to the sparse nature of the spatial pyramid features, we use PCA to reduce each feature to 3000 dimensions. In our experiment, 30 images per class are randomly chosen as training data, and other images are used as testing data. All the steps and settings follow [12]. The dictionary size is set to 510, $\lambda_1 = 0.001, \lambda_4 = 4.6$ and $T = 990$.

TABLE 3
Classification Results on Caltech101 Dataset

Methods	Classification Accuracy(%)	Training Time(s)	Testing Time(s)
ADL+SVM [16]	54.93%	**447.80**	7.75×10^{-3}
SRC [7]	67.70%	No Need	4.34×10^{-1}
LC-KSVD [12]	71.79%	487.61	1.35×10^{-2}
SADL	**72.36%**	773.66	$\mathbf{8.10}\times10^{-6}$
DSADL	**73.49%**	-	$\mathbf{8.10}\times10^{-6}$
ADL+SVM [16]	66.75%	1943.47	1.33×10^{-2}
SRC [7]	70.70%	No Need	4.34×10^{-1}
LC-KSVD [12]	73.67(73.6 [12])%	2144.90	2.49×10^{-3}
SADL	**74.17%**	**1406.68**	$\mathbf{4.76}\times10^{-5}$

The classification results, training and testing times are summarized in Table 3. The dictionary size in the above part of the table is 510, and the one in the below part is 3060 (all training samples). Our proposed SADL still achieves the highest performance of the lot. SADL has again a short encoding time, which is around 10000 times faster than LC-KSVD. For the second part of the Table 3, the dictionary size is increased to 3060 (i.e., all the training sample size), and $\lambda_1 = 0.001, \lambda_4 = 1.5, T = 1110$, our SADL again achieves the highest accuracy with the fastest training and testing time.

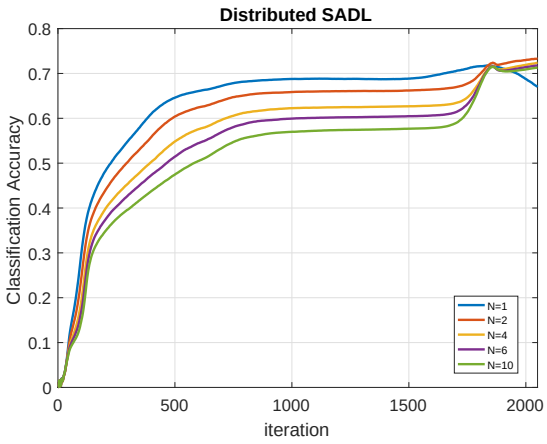

Fig. 9. Distributed SADL on Catech101: N is the number of clusters used. $N = 1$ is centralized. Training set is divided into N groups.

Let the penalty coefficients of the communication cost $\xi_{1_t} = \xi_{2_t} = \xi_{3_t} = 0.1, \forall t$, and other parameters are the same as SADL settings. Distributed SADL also achieves the highest accuracy with the same testing time as SADL, which is also shown in the above Table (3). Figure (9) shows that when the number of groups is increased, the accuracy is actually lower at first because of the smaller training sample size of each independent variable. But after the affects of the communication between global variables and local independent variables are enhanced, the performance rises up very quickly to a high generalized accuracy. Distributed SADL is demonstrated that it can also obtain a very stable and excellent performance even when the number of groups is large.

6.6 Caltech256

The Caltech256 is a relative lager objective dataset, which includes 256 object categories and one clutter. There are totally 30608 images with various object location, pose, and size. Figure 10 shows the examples of Caltech 256, whose each category has at least 80 images. The features of Caltech256 images are extracted by using the output features of the last layer before fully connected layer of ResNet-50 [25] with the weights trained by ImageNet. The dimension of each feature is 2046×1. We randomly sample 15 images from each category for training, and test on the rest of them. To train the Distributed SADL, the dictionary size is set to 3855, dataset is divided into 3 subsets (*i.e.*, $t = 3$ in Algorithm 2), $\lambda_1 = 0.001, \lambda_4 = 0.5$, $\xi_{1_t} = \xi_{2_t} = \xi_{3_t} = 3 \times 10^{-5}, \forall t$ and $T = 4495$.

The Caltech256 are applied by our Distributed SADL, ADL+SVM and LC-KSVD with same dictionary size. Our

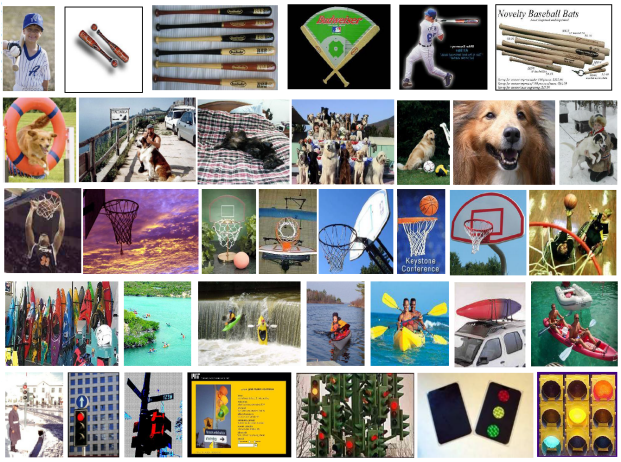

Fig. 10. Caltech256 Dataset Examples

TABLE 4
Classification Results on Caltech256 Dataset

Methods (training samples)	Classification Accuracy(%)	Training Time(s)	Testing Time(s)
ADL+SVM(15) [16]	66.66%	3501.44	7.67×10^{-2}
LC-KSVD(15) [12]	73.37%	3118.76	3.00×10^{-3}
CNN Features(15) [26]	65.70% [26]	-	-
DSADL(15)	**74.38%**	-	$\mathbf{4.46}\times10^{-5}$
ResFeats-50(30) [27]	75.40% [27]		

Distributed SADL also achieves the best performance with a extreme fast testing time, even though the dimension of feautres are increased. For more reference, we also compare our method with two network methods [26], [27]. In [26], Zeiler *et al.* constructed a convnet pertrained by ImageNet, and then learned a convolutional network for Caltech256 based on it. As trained the same training size as our settings, our performance is 10% higher than the CNN result. ResFeats-50 [27] is a most recent convolutional network method. This method is trained by 30 samples of each category with 50 layers. Though ResFeats-50 with double training samples than ours, our result is still very comparable.

6.7 Scene15

Fig. 11. Scene15 Dataset Examples

Scene15 dataset contains a total of 15 categories of different scenes, and each category has around 200 images. The examples are listed in Figure 11. Proceeding as for the Caltech 101 dataset, we compute the spatial pyramid features for scene images. A four-level spatial pyramid (*i.e.*, each image is girded into 1×1, 2×2, 4×4 and 8×8) and a codebook of size 200 are used here. The final features are also obtained by applying PCA to reduce the dimension of spatial pyramid features to 3000. We randomly pick 100 image per class as training data, and use the rest of images as testing data. The settings and steps follow [12]. The dictionary size is set to 450, $\lambda_1 = 0.001, \lambda_4 = 0.001$ and $T = 220$.

TABLE 5
Classification Results on Scene15 Dataset

Methods	Classification Accuracy(%)	Training Time(s)	Testing Time(s)
ADL+SVM [16]	49.35%	**110.47**	1.14×10^{-4}
SRC [7]	91.80%	No Need	4.06×10^{-1}
LC-KSVD [12]	**98.83%** (92.7% [12])	270.93	1.26×10^{-2}
SADL	98.16%	121.02	$\mathbf{9.23}\times10^{-6}$

The classification results, training and testing time are summarized in Table 5. Our performance is slightly lower than LC-KSVD, but is still higher than SRC, ADL+SVM and the LC-KSVD reported accuracy. However, the testing phase is superior to the others. Note that, the testing time is 10 thousand times faster than LC-KSVD.

7 Conclusion

We proposed an image classification method referred to as structured analysis dictionary learning (SADL). To obtain SADL, we constrain a structured subspace (cluster) model in the enhanced ADL method, where each class is represented by a structured subspace. The enhancement of ADL is realized by constraining the learning by a classification fidelity term on the sparse coefficients. Our formulated optimization problem was efficiently solved by the linearized ADM method, in spite of its non-convexity due to bilinearity. Taking advantage of analysis dictionary, our method achieves a significantly faster testing time. Furthermore, a Distributed SADL (DSADL) was also developed to address the scalability problem. Both discriminative structure and fast testing phase are well preserved in the DSADL. Even though the algorithm was run by many multi-clusters, the performance was still stable and comparable to the centralized SADL.

Our experiments demonstrate that our approach has at least a comparable, and often a better performance than state-of-the-art techniques on five well known datasets and achieves superior training and testing times by orders of magnitude.

A possible future direction for improving our method could be to leverage the discriminative nature of the synthesis dictionary and the efficiency of the analysis dictionary together. This can achieve a more discriminative power and high efficiency.

Acknowledgments

We gratefully acknowledge the generous support of ARO under grant W911NF-16-2-0005.

APPENDIX A
PROOF OF OPTIMIZATION TRANSFORM

As mentioned in Section 2 of the paper, the primal problem of our work is

$$\begin{aligned}\arg \min_{\substack{\Omega,U,Q,W,\\ \varepsilon_1,\varepsilon_2}} \ & \frac{1}{2}\|U-\Omega X\|_F^2+\lambda_1\|U\|_1 \\ & +\frac{\rho_1}{2}\|\varepsilon_1\|_2^2+\frac{\rho_2}{2}\|\varepsilon_2\|_2^2 \\ & +\frac{\delta_1}{2}\|Q\|_2^2+\frac{\delta_2}{2}\|W\|_2^2 \\ s.t.\ & H=QU+\varepsilon_1, \\ & L=WQU+\varepsilon_2, \\ & \|\omega_i^T\|_2^2=1; \forall i=1,\ldots,r,\end{aligned} \tag{25}$$

Then, the augmented Lagrangian dual optimization of Eq. (25) is expressed as:

$$\max_{\substack{Z^{(1)},Z^{(2)},\\ \gamma_1,\gamma_2}} \min_{\substack{\Omega,U,Q,W,\\ \varepsilon_1,\varepsilon_2}} L(\Omega,U,Q,W,\varepsilon_1,\varepsilon_2,Z^{(1)},Z^{(2)},\gamma_1,\gamma_2),$$

where

$$\begin{aligned}&L(\Omega,U,Q,W,\varepsilon_1,\varepsilon_2,Z^{(1)},Z^{(2)},\gamma_1,\gamma_2)= \\ &\frac{1}{2}\|U-\Omega X\|_F^2+\lambda_1\|U\|_1+\frac{\rho_1}{2}\|\varepsilon_1\|_2^2+\frac{\rho_2}{2}\|\varepsilon_2\|_2^2 \\ &+\langle Z^{(1)},H-QU-\varepsilon_1\rangle+\langle Z^{(2)},L-WQU-\varepsilon_2\rangle \\ &+\frac{\gamma_1}{2}\|H-QU-\varepsilon_1\|_2^2+\frac{\gamma_2}{2}\|L-WQU-\varepsilon_2\|_2^2 \\ &+\frac{\delta_1}{2}\|Q\|_2^2+\frac{\delta_2}{2}\|W\|_2^2.\end{aligned} \tag{26}$$

By minimizing the ε_1 and ε_2, we obtain

$$\frac{\partial L}{\partial \varepsilon_1}=\rho_1\varepsilon_1-Z^{(1)}-\gamma_1(H-QU-\varepsilon_1)=0,$$

$$\varepsilon_1=\frac{1}{\gamma_1+\rho_1}Z^{(1)}+\frac{\gamma_1}{\gamma_1+\rho_1}(H-QU). \tag{27}$$

Similarly,

$$\varepsilon_2=\frac{1}{\gamma_2+\rho_2}Z^{(2)}+\frac{\gamma_2}{\gamma_2+\rho_2}(L-WQU). \tag{28}$$

Substituting Eqs. (27) and (28) into Eq. (35), we obtain

$$\begin{aligned}&L(\Omega,U,Q,W,Z^{(1)},Z^{(2)},\gamma_1,\gamma_2)= \\ &\frac{1}{2}\|U-\Omega X\|_F^2+\lambda_1\|U\|_1+ \\ &\frac{\rho_1}{2}\|\frac{1}{\gamma_1+\rho_1}Z^{(1)}+\frac{\gamma_1}{\gamma_1+\rho_1}(H-QU)\|_2^2+ \\ &\frac{\rho_2}{2}\|\frac{1}{\gamma_2+\rho_2}Z^{(2)}+\frac{\gamma_2}{\gamma_2+\rho_2}(L-WQU)\|_2^2+ \\ &\langle Z^{(1)},H-QU-(\frac{1}{\gamma_1+\rho_1}Z^{(1)}+\frac{\gamma_1}{\gamma_1+\rho_1}(H-QU))\rangle+ \\ &\langle Z^{(2)},L-WQU-(\frac{1}{\gamma_2+\rho_2}Z^{(2)}+\frac{\gamma_2}{\gamma_2+\rho_2}(L-WQU))\rangle \\ &+\frac{\gamma_1}{2}\|H-QU-\frac{1}{\gamma_1+\rho_1}Z^{(1)}-\frac{\gamma_1}{\gamma_1+\rho_1}(H-QU)\|_2^2 \\ &+\frac{\gamma_2}{2}\|L-WQU-\frac{1}{\gamma_2+\rho_2}Z^{(2)}-\frac{\gamma_2}{\gamma_2+\rho_2}(L-WQU)\|_2^2 \\ &+\frac{\delta_1}{2}\|Q\|_2^2+\frac{\delta_2}{2}\|W\|_2^2.\end{aligned} \tag{29}$$

After careful manipulations of Eq. (29), we have

$$\begin{aligned}&L(\Omega,U,Q,W,Z^{(1)},Z^{(2)},\gamma_1,\gamma_2)= \\ &\frac{1}{2}\|U-\Omega X\|_F^2+\lambda_1\|U\|_1+\frac{\delta_1}{2}\|Q\|_2^2+\frac{\delta_2}{2}\|W\|_2^2 \\ &+\frac{\rho_1}{\gamma_1+\rho_1}\langle Z^{(1)},H-QU\rangle \\ &+\frac{\rho_2}{\gamma_2+\rho_2}\langle Z^{(2)},L-WQU\rangle \\ &+\frac{\gamma_1\rho_1}{2(\gamma_1+\rho_1)}\|H-QU\|_2^2+\frac{\gamma_2\rho_2}{2(\gamma_2+\rho_2)}\|L-WQU\|_2^2 \\ &-\frac{1}{2(\gamma_1+\rho_1)}\|Z^{(1)}\|_2^2-\frac{1}{2(\gamma_2+\rho_2)}\|Z^{(2)}\|_2^2 \\ &+\frac{\delta_1}{2}\|Q\|_2^2+\frac{\delta_2}{2}\|W\|_2^2.\end{aligned} \tag{30}$$

The last two terms in Eq. (30) are not crucial in the optimization algorithm, and can be removed to obtain the Lagrangian equation in Eq. (5) in the paper. To see this, notice that updating the dual variables in augmented Lagrangian is by $Z^{(1)}=Z^{(1)}+\alpha\bigtriangledown_{Z^{(1)}}L$ and $Z^{(2)}=Z^{(2)}+\alpha\bigtriangledown_{Z^{(2)}}L$, where α is the learning step. This can be written as:

$$\begin{aligned}Z^{(1)}\leftarrow & Z^{(1)}-\frac{\alpha}{\gamma_1+\rho_1}Z^{(1)}+\frac{\alpha\rho_1}{\gamma_1+\rho_1}(H-QU), \\ Z^{(1)}\leftarrow & \left(1-\frac{\alpha}{\gamma_1+\rho_1}\right)Z^{(1)}+\frac{\alpha\rho_1}{\gamma_1+\rho_1}(H-QU).\end{aligned}$$

As γ_1, ρ_1 and α are all constants, when ρ_1 is big enough, $\frac{1}{\gamma_1+\rho_1}$ can be extremely small, and $\left(1-\frac{\alpha}{\gamma_1+\rho_1}\right)$ approximates to 1. So that, by omitting the scalar $\left(1-\frac{\alpha}{\gamma_1+\rho_1}\right)$ and replacing the scalar $\frac{\alpha\rho_1}{\gamma_1+\rho_1}$ by the learning rate μ, we estimate the updating of our dual variable $Z^{(1)}$ by

$$Z^{(1)}\leftarrow Z^{(1)}+\mu(H-QU). \tag{31}$$

Similarly, the updating of our dual variable $Z^{(2)}$ is estimated by

$$Z^{(2)}\leftarrow Z^{(2)}+\mu(L-WQU). \tag{32}$$

We observe that removing the last two terms in Eq. (29) leads to similar iterations, so that we can write $L(\Omega,U,Q,W,Z^{(1)},Z^{(2)},\gamma_1,\gamma_2)$ in Eq. (30) as:

$$\begin{aligned}&L(\Omega,U,Q,W,Z^{(1)},Z^{(2)},\gamma_1,\gamma_2)= \\ &\frac{1}{2}\|U-\Omega X\|_F^2+\lambda_1\|U\|_1+\frac{\delta_1}{2}\|Q\|_2^2+\frac{\delta_2}{2}\|W\|_2^2 \\ &+\frac{\rho_1}{\gamma_1+\rho_1}\langle Z^{(1)},H-QU\rangle \\ &+\frac{\rho_2}{\gamma_2+\rho_2}\langle Z^{(2)},L-WQU\rangle \\ &+\frac{\gamma_1\rho_1}{2(\gamma_1+\rho_1)}\|H-QU\|_2^2+\frac{\gamma_2\rho_2}{2(\gamma_2+\rho_2)}\|L-WQU\|_2^2\end{aligned} \tag{33}$$

Let $\lambda_2=\frac{\rho_1}{\gamma_1+\rho_1}$, $\lambda_3=\frac{\rho_2}{\gamma_2+\rho_2}$ and $\mu=\frac{\gamma_1\rho_1}{\gamma_1+\rho_1}=\frac{\gamma_2\rho_2}{\gamma_2+\rho_2}$, we have the new augmented Lagrangian as follows:

$$\begin{aligned}&L(\Omega,U,Q,W,Z^{(1)},Z^{(2)},\mu)= \\ &\frac{1}{2}\|U-\Omega X\|_F^2+\lambda_1\|U\|_1+\lambda_2\langle Z^{(1)},H-QU\rangle \\ &+\lambda_3\langle Z^{(2)},L-WQU\rangle+\frac{\mu}{2}\|H-QU\|_2^2 \\ &+\frac{\mu}{2}\|L-WQU\|_2^2+\frac{\delta_1}{2}\|Q\|_2^2+\frac{\delta_2}{2}\|W\|_2^2,\end{aligned} \tag{34}$$

which is the Eq. (5) in Section 3 of the paper.

APPENDIX B

Take the Lagrangian function

$$\begin{aligned} &L(\Omega, U, Q, W, \varepsilon^{(1)}, \varepsilon^{(2)}, Z^{(1)}, Z^{(2)}) = \\ &\frac{1}{2}\|U - \Omega X\|_F^2 + \lambda_1\|U\|_1 + \frac{\rho_1}{2}\|\varepsilon_1\|_2^2 + \frac{\rho_2}{2}\|\varepsilon_2\|_2^2 \\ &+ \langle Z^{(1)}, H - QU - \varepsilon_1\rangle + \langle Z^{(2)}, Y - WQU - \varepsilon_2\rangle \\ &+ \frac{\mu}{2}\|H - QU - \varepsilon_1\|_2^2 + \frac{\mu}{2}\|Y - WQU - \varepsilon_2\|_2^2 \\ &+ \frac{\delta_1}{2}\|Q\|_2^2 + \frac{\delta_2}{2}\|W\|_2^2. \end{aligned} \tag{35}$$

Our algorithm can be written as the one in Alg. 3. Let us

Algorithm 3 ADMM for Structured Analysis Dictionary Learning

At each iteration $k+1$, compute:

$$U_{k+1} = \tau_{\frac{\lambda_1}{\mu\eta_U}}\left(U_k - \tfrac{1}{\mu\eta_U}\bigtriangledown_U L_s(U_k, Q_k, W_k, \Omega_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\right), \tag{36}$$

$$Q_{k+1} = Q_k - \tfrac{1}{\mu\eta_Q}\bigtriangledown_Q L(U_{k+1}, Q_k, W_k, \Omega_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)}), \tag{37}$$

$$W_{k+1} = W_k - \tfrac{1}{\mu\eta_W}\bigtriangledown_W L(U_{k+1}, Q_{k+1}, W_k, \Omega_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)}), \tag{38}$$

$$\Omega_{k+1} = \arg\min_\Omega L(U_{k+1}, Q_{k+1}, W_{k+1}, \Omega, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)}), \tag{39}$$

$$\varepsilon_{k+1}^{(1)} = \arg\min_{\varepsilon^{(1)}} L(U_{k+1}, Q_{k+1}, W_{k+1}, \Omega_{k+1}, \varepsilon^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)}), \tag{40}$$

$$\varepsilon_{k+1}^{(2)} = \arg\min_{\varepsilon^{(2)}} L(U_{k+1}, Q_{k+1}, W_{k+1}, \Omega_{k+1}, \varepsilon_{k+1}^{(1)}, \varepsilon^{(2)}, Z_k^{(1)}, Z_k^{(2)}), \tag{41}$$

$$Z_{k+1}^{(1)} = Z_k^{(1)} + \mu(H - Q_{k+1}U_{k+1} - \varepsilon_{k+1}^{(1)}), \tag{42}$$

$$Z_{k+1}^{(2)} = Z_k^{(2)} + \mu(Y - W_{k+1}Q_{k+1}U_{k+1} - \varepsilon_{k+1}^{(2)}), \tag{43}$$

proceed by introducing two simple lemmas:

Lemma 2. *Consider a differentiable function f with an $L-$Lipschitz continuous derivative and another arbitrary convex function g. For any arbitrary point x define*

$$x^+ = \text{prox}_{\tau g}(x - \tau\nabla f(x)),$$

where $\tau > 0$ is a step size and

$$\text{prox}_{\tau g}(y) = \arg\min_x \frac{1}{2}\|x - y\|^2 + \tau g(x).$$

Then, we have

$$F(x^+) - F(x) \leq \left(\frac{L}{2} - \frac{1}{\tau}\right)\|x - x^+\|^2,$$

where $F(x) = f(x) + g(x)$.

Proof. Notice that by the definition of the proximal operator prox, there exists a subgradient $\xi \in \partial g(x^+)$ such that

$$x^+ = x^- - \tau\xi$$

where $x^- = x - \tau\nabla f(x)$. Then, we have

$$g(x) \geq g(x^+) + \langle x - x^0, \xi\rangle$$

On the other hand,

$$f(x) \geq f(x^+) + \langle x - x^+, \nabla f(x)\rangle - \frac{L}{2}\|x - x^+\|^2$$

Adding the two inequalities yields

$$F(x) \geq F(x^+) + \langle x - x^+, \nabla f(x) + \xi\rangle - \frac{L}{2}\|x - x^+\|^2$$

Now noticing that $\tau(\nabla f(x) + \xi) = x - x^+$ completes the proof. □

Lemma 3. *Take a differentiable function f and a convex function g and suppose that a point x satisfies*

$$\text{prox}_{\tau g}(x - \tau\nabla f(x)) = x$$

Then, x is a stationary point of $F = f + g$, i.e. $-\nabla f(x) \in \partial g(x)$.

Proof. From the definition of the proximal operator there exists a vector $\xi \in \partial g(x)$ such that $x = x - \tau\nabla f(x) - \tau\xi$. We conclude that $-\nabla f(x) = \xi$, which completes the proof. □

Next, we make a simple but crucial observation about our algorithm:

Lemma 4. *For Algorithm 3 the following holds in every iteration k:*

$$Z_{k+1}^{(1)} = \rho_1\varepsilon_{k+1}^{(1)},$$
$$Z_{k+1}^{(2)} = \rho_2\varepsilon_{k+1}^{(2)},$$

and as a result,

$$\|Z_{k+1}^{(1)} - Z_k^{(1)}\| = \rho_1\|\varepsilon_{k+1}^{(1)} - \varepsilon_k^{(1)}\|, \tag{44}$$

$$\|Z_{k+1}^{(2)} - Z_k^{(2)}\| = \rho_2\|\varepsilon_{k+1}^{(2)} - \varepsilon_k^{(2)}\|. \tag{45}$$

Proof. From the $\varepsilon^{(1)}$ update rule (40),we have the following optimality condition

$$\rho_1\varepsilon_{k+1}^{(1)} - Z_k^{(1)} - \mu(H - Q_{k+1}U_{k+1} - \varepsilon_{k+1}^{(1)}) = 0. \tag{46}$$

Combined with dual variable $Z_{k+1}^{(1)}$ update rule (42), we obtain

$$Z_{k+1}^{(1)} = \rho_1\varepsilon_{k+1}^{(1)}. \tag{47}$$

The result for $Z_{k+1}^{(2)}$ is similarly obtained. □

We take $L_k = L(\Omega_k, U_k, Q_k, W_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)}, \mu_k)$ for $k = 0, 1, 2, \ldots$ and notice that the change in L_k can be controlled by the following result:

Lemma 5.

$$\begin{aligned} &L_{k+1} - L_k \\ &\leq \left(\tfrac{\alpha_{k,U}}{2} - \mu\eta_U\right)\|U_{k+1} - U_k\|^2 + \left(\tfrac{\alpha_{k,Q}}{2} - \mu\eta_Q\right)\|Q_{k+1} - Q_k\|^2 \\ &+ \left(\tfrac{\alpha_{k,W}}{2} - \mu\eta_W\right)\|W_{k+1} - W_k\|^2 - \tfrac{m_\Omega}{2}\|\Omega_{k+1} - \Omega_k\|^2 \\ &+ \left(\tfrac{\rho_1^2}{\mu} - \tfrac{m_{\varepsilon^{(1)}}}{2}\right)\|\varepsilon_{k+1}^{(1)} - \varepsilon_k^{(1)}\|^2 + \left(\tfrac{\rho_2^2}{\mu} - \tfrac{m_{\varepsilon^{(2)}}}{2}\right)\|\varepsilon_{k+1}^{(2)} - \varepsilon_k^{(2)}\|^2, \end{aligned} \tag{48}$$

where

$$\alpha_{k,U} = 1 + \mu\|Q_k^TQ_k + Q_k^TW_k^TW_kQ_k\|_*$$
$$\alpha_{k,Q} = \delta_1 + \mu\|W_k^TW_k\|_*\|U_{k+1}U_{k+1}^T\|_*$$
$$\alpha_{k,W} = \delta_2 + \mu\|Q_{k+1}U_{k+1}U_{k+1}^TQ_{k+1}^T\|_*$$
$$m_\Omega = \sigma_{\min}(XX^T)$$
$$m_{\varepsilon^{(1)}} = \rho_1 + \mu, \quad m_{\varepsilon^{(2)}} = \rho_2 + \mu,$$

Proof. Respectively denote by $\Delta L_{k,U}, \Delta L_{k,Q}, \Delta L_{k,W}, \Delta L_{k,\Omega}, \Delta L_{k,\varepsilon^{(i)}}, \Delta L_{k,Y^{(i)}}$ for

$i = 1, 2$, the change in L corresponding to the update of $U, Q, W, \Omega, \varepsilon^{(i)}$ and $Y^{(i)}$ in Eq. (2-9). Notice that

$$L_{k+1} - L_k = \Delta L_{k,U} + \Delta L_{k,Q} + \Delta L_{k,W} + \Delta L_{k,\Omega} + \Delta L_{k,\varepsilon^{(1)}} + \Delta L_{k,\varepsilon^{(2)}} + \Delta L_{k,Z^{(1)}} + \Delta L_{k,Z^{(2)}}$$

Notice that by taking $f(U) = L_s(\Omega_k, U, Q_k, W_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})$, $g(U) = \lambda_1 \|U\|_1$ and $\tau = 1/\mu\eta_U$, and recalling Lemma 2, we have

$$\Delta L_{k,U} \leq \left(\frac{\alpha_{k,U}}{2} - \mu\eta_U\right) \|U_{k+1} - U_k\|^2 \tag{49}$$

where we use the fact that $f(U)$ is quadratic, hence possessing $\alpha_{k,U}-$Lipschitz derivatives with $\alpha_{k,U}$ being the largest singular value of the Hessian. Similarly, by taking $f(Q) = L(\Omega_k, U_{k+1}, Q, W_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)}), g(Q) = 0, \tau = 1/\mu\eta_Q$ and $f(W) = L(\Omega_k, U_{k+1}, Q_{k+1}, W, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)}), g(W) = 0, \tau = 1/\mu\eta_U$ and utilizing Lemma 2, we respectively obtain

$$\Delta L_{k,Q} \leq \left(\frac{\alpha_{k,Q}}{2} - \mu\eta_Q\right) \|Q_{k+1} - Q_k\|^2 \tag{50}$$

$$\Delta L_{k,W} \leq \left(\frac{\alpha_{k,W}}{2} - \mu\eta_W\right) \|W_{k+1} - W_k\|^2 \tag{51}$$

Next, notice that the function $f(\Omega) = L(\Omega, U_{k+1}, Q_{k+1}, W_{k+1}, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})$ is quadratic and $m_\Omega-$strongly convex, where m_Ω is the smallest singular value of Hessian. Hence,

$$\Delta L_{k,\Omega} = f(\Omega_k) - \min_{\Omega} f(\Omega) \leq -\frac{m_\Omega}{2} \|\Omega_{k+1} - \Omega_k\|^2 \tag{52}$$

Similarly, taking $f(\varepsilon^{(1)}) = L(\Omega_{k+1}, U_{k+1}, Q_{k+1}, W_{k+1}, \varepsilon^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})$ and $f(\varepsilon^{(2)}) = L(\Omega_{k+1}, U_{k+1}, Q_{k+1}, W_{k+1}, \varepsilon_{k+1}^{(1)}, \varepsilon^{(2)}, Z_k^{(1)}, Z_k^{(2)})$, yield

$$\Delta L_{k,\varepsilon^{(i)}} \leq -\frac{m_{\varepsilon^{(i)}}}{2} \left\|\varepsilon_{k+1}^{(i)} - \varepsilon_k^{(i)}\right\|^2, \quad i = 1, 2 \tag{53}$$

Finally, notice that

$$\begin{aligned}\Delta L_{k,Z^{(1)}} &= \left\langle Z_{k+1}^{(1)} - Z_k^{(1)}, H - Q_{k+1}U_{k+1} - \varepsilon_{k+1}^{(1)} \right\rangle \\ &= \left\langle Z_{k+1}^{(1)} - Z_k^{(1)}, \frac{1}{\mu}\left(Z_{k+1}^{(1)} - Z_k^{(1)}\right) \right\rangle = \frac{1}{\mu}\left\|Z_{k+1}^{(1)} - Z_k^{(1)}\right\|^2 \\ &= \frac{\rho_1^2}{\mu}\left\|\varepsilon_{k+1}^{(1)} - \varepsilon_k^{(1)}\right\|^2\end{aligned}$$

Similarly, we obtain

$$\Delta L_{k,Z^{(2)}} = \frac{\rho_2^2}{\mu}\left\|\varepsilon_{k+1}^{(2)} - \varepsilon_k^{(2)}\right\|^2 \tag{54}$$

Summing the inequalities in Eq. (49), Eq. (50), Eq. (51), Eq. (52), Eq. (53),Eq. (5) and Eq. (54) completes the proof. □

Now, we have the following theorem:

Theorem 6. *Suppose that $\mu \geq \sqrt{2}\{\rho_1, \rho_2\}$. There exist positive values $\eta_U^0, \eta_Q^0, \eta_W^0$ only depending on the initial values such that for $\eta_U > \eta_U^0, \eta_Q > \eta_Q^0, \eta_W > \eta_W^0$ the sequence $\{L_k\}_{k=1}^\infty$ is positive and decreasing, hence convergent.*

Proof. First define

$$L_{k,e}(\Omega, U, Q, W) = L(\Omega, U, Q, W, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})$$

observe that according to Lemma 4, for $k = 1, 2, \ldots$ we have that

$$\begin{aligned}L_{k,e} &= \frac{1}{2}\|U - \Omega X\|_F^2 + \lambda_1\|U\|_1 \\ &+\rho_1\langle\varepsilon_k^{(1)}, H - QU - \varepsilon_k^{(1)}\rangle + \tfrac{\mu}{2}\|H - QU - \varepsilon_k^{(1)}\|_2^2 + \tfrac{\rho_1}{2}\|\varepsilon_k^{(1)}\|_2^2 \\ &+\rho_2\langle\varepsilon_k^{(2)}, Y - WQU - \varepsilon_k^{(2)}\rangle + \tfrac{\mu}{2}\|Y - WQU - \varepsilon_k^{(2)}\|_2^2 + \tfrac{\rho_2}{2}\|\varepsilon_k^{(2)}\|_2^2 \\ &+ \frac{\delta_1}{2}\|Q\|_2^2 + \frac{\delta_2}{2}\|W\|_2^2. \\ &= \frac{1}{2}\|U - \Omega X\|_F^2 + \lambda_1\|U\|_1 \\ &+ \frac{\mu}{2}\left\|H - QU - \left(1 - \frac{\rho_1}{\mu}\right)\varepsilon_k^{(1)}\right\|_2^2 + \frac{\rho_1}{2}\left(1 - \frac{\rho_1}{\mu}\right)\|\varepsilon_k^{(1)}\|_2^2 \\ &+ \frac{\mu}{2}\left\|Y - WQU - \left(1 - \frac{\rho_2}{\mu}\right)\varepsilon_k^{(2)}\right\|_2^2 + \frac{\rho_2}{2}\left(1 - \frac{\rho_2}{\mu}\right)\|\varepsilon_k^{(2)}\|_2^2 \\ &+ \frac{\delta_1}{2}\|Q\|_2^2 + \frac{\delta_2}{2}\|W\|_2^2.\end{aligned} \tag{55}$$

Hence, for $\mu > \max\{\rho_1, \rho_2\}$, we $L_{k,e} \geq 0$. In particular, we obtain that $L_k = L_{k,e}(\Omega_k, U_k, Q_k, W_k) \geq 0$. Now, we use complete (strong) indiction to show that $L_{k+1} \geq L_k$ for $k = 1, 2, \ldots$. Suppose that this holds for $k = 1, 2, \ldots, t$. We conclude that $L_t \leq L_1$. Now, notice that from (55) and the fact that $L_t = L_{t,e}(\Omega_t, U_t, Q_t, W_t)$ we obtain for $\mu > \max\{\rho_1, \rho_2\}$ that

$$\|Q_t\|^2 \leq \frac{2L_1}{\delta_1}, \quad \|W_t\|^2 \leq \frac{2L_1}{\delta_2}$$

which leads to the following:

$$\alpha_{t,U} \leq 1 + \mu\left(\|Q_t\|^2 + \|Q_t\|^2\|W_t\|^2\right) \leq 1 + \frac{2L_1\mu}{\delta_1}\left(1 + \frac{2L_1}{\delta_2}\right)$$

Now, from (49), we observe that by selecting $\eta_U > \left[2 + \frac{2L_1\mu}{\delta_1}\left(1 + \frac{L_1}{\delta_2}\right)\right]/2\mu$ we have that

$$\Delta L_{t,U} \leq -\frac{1}{2}\|U_{t+1} - U_t\|^2 \tag{56}$$

which subsequently yields,

$$L_{t,e}(\Omega_t, U_{t+1}, Q_t, W_t) \leq L_t \leq L_1$$

Then according to (55) for $\mu > \max\{\rho_1, \rho_2\}$, we have that

$$\|U_{t+1}\|_1 \leq \frac{L_1}{\lambda_1}$$

We conclude that

$$\alpha_{t,Q} \leq \delta_1 + \mu\|W_t\|^2\|U_{t+1}\|_1^2 \leq \delta_1 + \mu\frac{2L_1^2}{\lambda_1\delta_2}$$

Now, by taking $\eta_U > \left[1 + \delta_1 + \frac{2\mu L_1^2}{\lambda_1\delta_2}\right]/2\mu$ in (50) we have that

$$\Delta L_{t,Q} \leq -\frac{1}{2}\|Q_{t+1} - Q_t\|^2 \tag{57}$$

This also results in

$$L_{t,e}(\Omega_t, U_{t+1}, Q_{t+1}, W_t) \leq L_{t,e}(\Omega_t, U_{t+1}, Q_t, W_t) \leq L_t \leq L_1$$

which using (55) for $\mu > \max\{\rho_1, \rho_2\}$ leads to

$$\|Q_{t+1}\|^2 \leq \frac{2L_1}{\delta_1}$$

and hence

$$\alpha_{t,W} \leq \delta_2 + \mu\|Q_{t+1}\|^2\|U_{t+1}\|_1^2 \leq \delta_2 + \frac{2\mu L_1^2}{\delta_1\lambda_1}$$

Now, we can also chose $\eta_W \geq \left[1 + \delta_2 + \frac{2\mu L_1^2}{\lambda_1\delta_1}\right]/2\mu$ we conclude from (51) that

$$\Delta L_{t,W} \leq -\frac{1}{2}\left\|W_{t+1} - W_t\right\|^2. \tag{58}$$

Finally, by choosing $\mu > \sqrt{2}\max\{\rho_1, \rho_2\}$, we obtain from Lemma 5 that

$$\begin{aligned} &L_{t+1} - L_t \leq -\tfrac{1}{2}\|U_{t+1} - U_t\|_2^2 - \tfrac{1}{2}\|Q_{t+1} - Q_t\|_2^2 - \tfrac{1}{2}\|W_{t+1} - W_t\|_2^2 \\ &-\tfrac{m_\Omega}{2}\|\Omega_{t+1} - \Omega_t\|_2^2 - \tfrac{\rho_1}{2}\|\varepsilon_{t+1}^{(1)} - \varepsilon_t^{(1)}\|_2^2 - \tfrac{\rho_2}{2}\|\varepsilon_{t+1}^{(2)} - \varepsilon_t^{(2)}\|_2^2 \end{aligned} \tag{59}$$

We conclude that $L_{t+1} \leq L_t$ which completes the proof. □

We finally obtain the following corollary which clarifies the statement and gives the proof of our main result in Theorem 1:

Corollary 1. Suppose that $\mu \geq \sqrt{2}\{\rho_1, \rho_2\}$. There exist positive values $\eta_U^0, \eta_Q^0, \eta_W^0, R$ only depending on the initialization such that for $\eta_U > \eta_U^0, \eta_Q > \eta_Q^0, \eta_W > \eta_W^0$ the sequence $\{\Theta_k = (\Omega_k, U_k, Q_k, W_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\}_{k=1}^{\infty}$ satisfies the following:

1) The parameters for $k = 0, 1, 2, ...$ are bounded by R, i.e

$$\|\Theta_k\| = \max\left\{\|\Omega_k\|, \|U_k\|, \|Q_k\|, \|W_k\|, \|\varepsilon_k^{(1)}\|, \|\varepsilon_k^{(2)}\|, \|Z_k^{(1)}\|, \|Z_k^{(2)}\|\right\} < R.$$

   Hence, the are confined in a compact set.
2) Any convergence subsequence of $\{\Theta_k\}$ converges to a point $\Theta^* \in S$.
3) $\mathrm{dist}(\Theta_k, S)$ converges to zero, where

$$\mathrm{dist}(\Theta, S) = \min_{\Theta' \in S}\|\Theta' - \Theta\|$$

Proof. Part a is simply obtained by noticing (55) and the fact that $L_{k,e}(\Omega_k, U_k, Q_k, W_k) = L_k \leq L_1$, since $\{L_k\}$ is decreasing. For part b, note that since the sequence $\{L_k\}$ is convergent, we have $\lim_{k\to\infty} L_{k+1} - L_k = 0$, which according to (59) yields

$$\begin{aligned} &\lim_{k\to\infty}\|U_{k+1} - U_k\|_2^2 = \lim_{k\to\infty}\|Q_{k+1} - Q_k\|_2^2 = \lim_{k\to\infty}\|W_{k+1} - W_k\|_2^2 \\ &= \lim_{k\to\infty}\|\Omega_{k+1} - \Omega_k\|_2^2 = \|\varepsilon_{k+1}^{(i)} - \varepsilon_k^{(i)}\|_2^2 = 0 \end{aligned}$$

for $i = 1, 2$. Also from Lemma 4 we have that

$$\lim_{k\to\infty}\|Z_{k+1}^{(i)} - Z_k^{(i)}\|_2^2 = 0$$

We conclude that

$$\lim_{k\to\infty}\left\|\tau_{\frac{\lambda_1}{\mu\eta_U}}\left(U_k - \tfrac{1}{\mu\eta_U}\nabla_U L_s(U_k, Q_k, W_k, \Omega_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\right) - U_k\right\|_2^2 = 0$$

$$\lim_{k\to\infty}\left\|\nabla_Q L(U_{k+1}, Q_k, W_k, \Omega_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\right\|_2^2 = 0$$

$$\lim_{k\to\infty}\left\|\nabla_W L(U_{k+1}, Q_{k+1}, W_k, \Omega_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\right\|_2^2 = 0$$

$$\lim_{k\to\infty}\left\|H - Q_{k+1}U_{k+1} - \varepsilon_{k+1}^{(1)}\right\|_2^2 = 0$$

$$\lim_{k\to\infty}\left\|Y - W_{k+1}Q_{k+1}U_{k+1} - \varepsilon_{k+1}^{(2)}\right\|_2^2 = 0$$

Moreover, note that the Lagrangian L is $L_\Omega-$second order Lipschitz with respect to Ω (fixing the rest) with $L_\Omega = \|XX^T\|_*$. We obtain that

$$\begin{aligned} &\left\|\nabla_\Omega L(U_{k+1}, Q_{k+1}, W_{k+1}, \Omega_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\right\|_2^2 \\ &\leq L_\Omega^2\|\Omega_{k+1} - \Omega_k\|_2^2 \end{aligned}$$

which yields

$$\lim_{k\to\infty}\left\|\nabla_\Omega L(U_{k+1}, Q_{k+1}, W_{k+1}, \Omega_k, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\right\|_2^2 = 0$$

Similarly, we obtain

$$\lim_{k\to\infty}\left\|\nabla_{\varepsilon^{(1)}} L(U_{k+1}, Q_{k+1}, W_{k+1}, \Omega_{k+1}, \varepsilon_k^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\right\|_2^2 = 0$$

$$\lim_{k\to\infty}\left\|\nabla_{\varepsilon^{(2)}} L(U_{k+1}, Q_{k+1}, W_{k+1}, \Omega_{k+1}, \varepsilon_{k+1}^{(1)}, \varepsilon_k^{(2)}, Z_k^{(1)}, Z_k^{(2)})\right\|_2^2 = 0$$

Now, take a subsequence of $\{\Theta_k\}$ converging to a point $\Theta_* = (\Omega_*, U_*, Q_*, W_*, \varepsilon_*^{(1)}, \varepsilon_*^{(2)}, Z_*^{(1)}, Z_*^{(2)})$. Since the argument of the above limits are continuous we obtain

$$\tau_{\frac{\lambda_1}{\mu\eta_U}}\left(U_* - \frac{1}{\mu\eta_U}\nabla_U L_s(\Theta_*)\right) - U_* = 0$$

$$\nabla_Q L(\Theta_*) = 0, \quad \nabla_W L(\Theta_*) = 0, \quad \nabla_{\varepsilon^{(i)}} L(\Theta_*) = 0$$

$$\nabla_{Z^{(1)}} L(\Theta_*) = H - Q_*U_* - \varepsilon_*^{(1)} = 0,$$

$$\nabla_{Z^{(2)}} L(\Theta_*) = Y - W_*Q_*U_* - \varepsilon_*^{(2)}$$

According to Lemma 3, we conclude that $\Theta_* \in S$. For part c, suppose that the claim is not true. Then, according to part a there exists a convergent subsequence of $\{\Theta_k\}$ which is $\varepsilon-$distant from S. Then, the convergence point is also $\varepsilon-$distant from S which contradicts part b and completes the proof.

□

REFERENCES

[1] B. A. Olshausen *et al.*, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[2] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.

[3] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on image processing*, vol. 17, no. 1, pp. 53–69, 2008.

[4] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2272–2279.

[5] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 689–696.

[6] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.

[7] 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, 2009.

[8] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3501–3508.

[9] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 543–550.

[10] Z. Wang, J. Yang, N. Nasrabadi, and T. Huang, "A max-margin perspective on sparse representation-based classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1217–1224.

[11] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, 2009, pp. 1033–1040.

[12] 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.

[13] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.

[14] R. Rubinstein, T. Peleg, and M. Elad, "Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model," *Signal Processing, IEEE Transactions on*, vol. 61, no. 3, pp. 661–677, 2013.

[15] X. Bian, H. Krim, A. Bronstein, and L. Dai, "Sparsity and nullity: Paradigms for analysis dictionary learning," *SIAM Journal on Imaging Sciences*, vol. 9, no. 3, pp. 1107–1126, 2016.

[16] S. Shekhar, V. M. Patel, and R. Chellappa, "Analysis sparse coding models for image-based classification," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 5207–5211.

[17] J. Guo, Y. Guo, X. Kong, M. Zhang, and R. He, "Discriminative analysis dictionary learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[18] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[19] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in neural information processing systems*, 2011, pp. 612–620.

[20] 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.

[21] A. Martinez and R. Benavente, "The ar face database," *CVC Technical Report*, no. 24, June 1998.

[22] 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," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[23] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[24] 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)*, vol. 2. IEEE, 2006, pp. 2169–2178.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[27] A. Mahmood, M. Bennamoun, S. An, and F. Sohel, "Resfeats: Residual network based features for image classification," *arXiv preprint arXiv:1611.06656*, 2016.

[28] X. Bian, H. Krim, A. Bronstein, and L. Dai, "Sparse null space basis pursuit and analysis dictionary learning for high-dimensional data analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. ICASSP, 2015.

[29] G. Peyré and J. M. Fadili, "Learning analysis sparsity priors," in *Sampta'11*, 2011, pp. 4–pp.

[30] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse problems*, vol. 23, no. 3, p. 947, 2007.

[31] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 524–531.

[32] W. Tang, I. R. Otero, H. Krim, and L. Dai, "Analysis dictionary learning for scene classification," in *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. IEEE, 2016, pp. 1–5.

[33] A. Koppel, G. Warnell, and E. Stump, "Task-driven dictionary learning in distributed online settings," in *Signals, Systems and Computers, 2015 49th Asilomar Conference on*. IEEE, 2015, pp. 1114–1118.

[34] W. Tang, A. Panahi, H. Krim, and L. Dai, "Structured analysis dictionary learning for image classification," *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference*, 2018. [Online]. Available: https://arxiv.org/abs/1805.00597