On the Equivalence of the LC-KSVD and the D-KSVD Algorithms

Igor Kviatkovsky, Moshe Gabel,
Ehud Rivlin, *Senior Member, IEEE*
and Ilan Shimshoni, *Member, IEEE*

Abstract—Sparse and redundant representations, where signals are modeled as a combination of a few atoms from an overcomplete dictionary, is increasingly used in many image processing applications, such as denoising, super resolution, and classification. One common problem is learning a "good" dictionary for different tasks. In the classification task the aim is to learn a dictionary that also takes training labels into account, and indeed there exist several approaches to this problem. One well-known technique is D-KSVD, which jointly learns a dictionary and a linear classifier using the K-SVD algorithm. LC-KSVD is a recent variation intended to further improve on this idea by adding an explicit *label consistency* term to the optimization problem, so that different classes are represented by different dictionary atoms. In this work we prove that, under identical initialization conditions, LC-KSVD with uniform atom allocation is in fact a reformulation of D-KSVD: given the regularization parameters of LC-KSVD, we give a closed-form expression for the equivalent D-KSVD regularization parameter, assuming the LC-KSVD's initialization scheme is used. We confirm this by reproducing several of the original LC-KSVD experiments.

Index Terms—Discriminative dictionary learning, Label consistent K-SVD, Discriminative K-SVD, equivalence proof.

1 Introduction

Sparse and redundant representations have been successfully applied to solve various problems in image processing and computer vision, such as image denoising [4], image inpainting [5], super resolution [15] and classification [14]. The fundamental idea behind all these works is representing signals using a sparse combination of atoms from large (overcomplete) dictionaries. It was shown that using a dictionary learned from the actual set of data samples rather than building it using a predefined basis, such as redundant Haar, results in an improved performance. The Method of Optimal Directions (MOD) [6] and the K-SVD [1] algorithms address exactly this issue of efficiently learning overcomplete dictionaries from data.

Following the success of signal reconstruction techniques based on dictionary learning, a new direction has emerged in recent years: learning dictionaries which also facilitate classification. Given a set of training signals and associated labels, the aim is to learn a dictionary and a classifier that can accurately predict the label of future test signals. While the classical problem of dictionary learning strives to minimize the signals' reconstruction error, learning algorithms for dictionaries used for classification also optimize for discriminative power. Practically speaking, unlike the classical setting where only the training samples (signals) are used, in a supervised learning setting the class labels corresponding to each signal are also taken into account.

- *I. Kviatkovsky, M. Gabel and E. Rivlin are with the Department of Computer Science, Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel.*
  E-mail: {kviat, mgabel, ehudr}@cs.technion.ac.il
- *I. Shimshoni is with the Department of Information Systems, University of Haifa, Carmel Mount, Rabin building, Haifa 31905, Israel.*
  E-mail: ishimshoni@mis.haifa.ac.il.

1.1 Background

Supervised dictionary learning methods differ in the way they exploit class labels. The most straightforward approach is to learn separate dictionaries using samples corresponding to each class, and then to classify the test signal according to its reconstruction error using each one of these per-class dictionaries. In SRC [14] this strategy is applied for the problem of face recognition, showing promising results.

Rather than using a pure *reconstructive* approach, Mairal et al. [11] propose to learn the per-class dictionaries in a *discriminative* approach by adding a classification loss term to the dictionary learning optimization task. The optimization problem is solved by alternately finding sparse representation given a dictionary, then updating the dictionary by minimizing a weighted combination of both reconstructive and discriminative terms, using the sparse representations (codes). Although this procedure is reminiscent of the classical K-SVD algorithm, they are quite different, since the dictionary update stage includes a discriminative term while the sparse coding stage does not. The classification loss was measured using a hard-to-optimize logistic loss function. Additional drawbacks of this approach are that it does not scale well with the number of classes, and is highly sensitive to the choice of weighting parameters, balancing between the reconstructive and the discriminative terms. On the contrary, Pham and Venkatesh [13] reported competitive results learning a single dictionary for all classes and a much simpler (linear) classifier using a similar iterative procedure. Zhang and Li [17] revised this approach and proposed the Discriminative K-SVD (D-KSVD) algorithm. In D-KSVD the problem of learning a single dictionary for all classes and a linear classifier is formulated as a joint optimization problem, solved using plain K-SVD. The authors showed that D-KSVD outperforms other competing methods including the SRC method.

In order to further improve the discriminative abilities of the learned linear classifier, Jiang et al. [10] proposed incorporating an additional term, called the discriminative sparse-code error, into the D-KSVD problem formulation. The resulting algorithm was named Label Consistent K-SVD (LC-KSVD) and the motivation for adding this term, given in [10], was to "encourage the signals from the same class to have similar sparse codes and those from different classes to have dissimilar sparse codes". While the idea seems reasonable, we show in this work that adding the discriminative sparse-code error term to the formulation of D-KSVD and solving it using the plain K-SVD algorithm, results in exactly the same classifier obtained by solving the original problem formulated in the D-KSVD, using an appropriate regularization parameter. Moreover, this term complicates parameter tuning by introducing an additional unnecessary regularization parameter and needlessly increases the runtime of the training phase due to an increase in the dimensionality of the K-SVD input.

1.2 Our Contribution

D-KSVD and LC-KVSD are commonly treated as two different algorithms. Indeed, many recent publications on face and object recognition (e.g., [2], [12], [16]) evaluate and compare their performance. In this work we correct this common misconception. We prove that LC-KSVD with a uniform allocation of labels to dictionary atoms, as proposed in [10] and commonly used in practice, is in fact exactly equivalent to D-KSVD with a proper choice of the regularization parameter and using the LC-KSVD's initialization scheme. This is further confirmed by reproducing the evaluation in [10] using the same datasets.

An immediate conclusion following from this result is that, although the authors of LC-KSVD were the first to coin the term "label consistency", it is actually an inherent property of previously existing supervised dictionary learning algorithms such as D-KSVD, although not explicitly stated in those terms.

The rest of the paper is organized as follows. Section 2 summarizes the K-SVD, the D-KSVD and the LC-KSVD algorithms. Section 3 presents the proof for the equivalence of D-KSVD and LC-KSVD. Section 4 presents empirical validation and Section 5 concludes with a short discussion and directions for future work.

2 Learning a Linear Classifier with K-SVD

In this section we define the notations and summarize the D-KSVD [17] and the LC-KSVD [10] algorithms. Since both algorithms heavily rely on the well-known K-SVD [1] algorithm, we first present it in a nutshell and describe the most trivial way to use it for classification.

Let $\mathbf{Y} \in \mathbb{R}^{n\times N}$ denote a set of N n-dimensional training signals with a corresponding label matrix $\mathbf{H} \in \mathbb{R}^{m\times N}$, where m is the number of classes. Each column $\mathbf{h}_i$ of the label matrix $\mathbf{H}$ encodes the class label of sample i using the position of the non-zero value. For example, if the label of sample $\mathbf{y}_i$ is 3, then $\mathbf{h}_i = [0, 0, 1, 0, \ldots, 0]^T$.

The original K-SVD algorithm, introduced by Aharon et al. [1] and summarized in Algorithm 1, solves the following optimization problem:

$$\begin{aligned}\langle \mathbf{D}^*, \mathbf{X}^* \rangle = &\underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \\ &s.t.\ \|\mathbf{x}_i\|_0 \le T_0,\ i = 1, \ldots, N,\end{aligned} \tag{1}$$

where T_0 is the sparsity constraint, making sure that each sparse representation $\mathbf{x}_i$ contains not more than T_0 non-zero entries. The dictionary $\mathbf{D} \in \mathbb{R}^{n\times K}$, where $K > n$ is the number of atoms in the dictionary, and the sparse codes $\mathbf{X} \in \mathbb{R}^{K\times N}$, obtained by the K-SVD solution of Eq. 1 minimize the signals' reconstruction error under the sparsity constraint T_0. Our goal, however, is to use the given label matrix, $\mathbf{H}$, to learn a linear classifier $\mathbf{W} \in \mathbb{R}^{m\times K}$ taking in a signal's sparse representation, $\mathbf{x}_i$, and returning the most probable class this signal belongs to. A straightforward approach, mentioned in [13], [17], is to solve the following linear *ridge regression* problem:

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}} \|\mathbf{H} - \mathbf{W}\mathbf{X}^*\|_F^2 + \lambda \|\mathbf{W}\|_F^2, \tag{2}$$

where λ is the regularization parameter. This problem has the following closed form solution:

$$\mathbf{W} = \mathbf{H}\mathbf{X}^{*T}\left(\mathbf{X}^*\mathbf{X}^{*T} + \lambda\mathbf{I}\right)^{-1}. \tag{3}$$

The drawback of this solution is that learning the classifier $\mathbf{W}$ is done independently from learning the dictionary $\mathbf{D}$ and the sparse codes $\mathbf{X}$, and is thus suboptimal: the dictionary learning procedure does not take into account the fact that its output will be used to train a classifier.

Algorithm 1 K-SVD

Input: $\mathbf{Y} \in \mathbb{R}^{n\times N}$, $\mathbf{D}^{(0)} \in \mathbb{R}^{n\times K}$, T_0.
Output: $\mathbf{D}^{(k)} \in \mathbb{R}^{n\times K}$, $\mathbf{X}^{(k)} \in \mathbb{R}^{K\times N}$.
Initialize: Set $k = 1$ and normalize the columns of $\mathbf{D}^{(0)}$.
Main Iteration: Repeat until convergence

1. **Sparse Coding Stage:** Use any pursuit algorithm (e.g., OMP [3]) to compute the representation vectors $\mathbf{x}_i$ for each example $\mathbf{y}_i$, by approximating the solution of
$$\mathbf{x}_i^{(k)} = \underset{\mathbf{x}_i}{\operatorname{argmin}} \left\|\mathbf{y}_i - \mathbf{D}^{(k-1)}\mathbf{x}_i\right\|_2^2,\ s.t.\ \|\mathbf{x}_i\|_0 \le T_0,$$
for $i = 1, 2, \ldots, N$.
2. **K-SVD Dictionary-Update Stage:** Update each column $j_0 = 1, 2, \ldots, K$ in $\mathbf{D} \equiv \mathbf{D}^{(k-1)}$:
   2.1. Define the group of example signals that use the atom $\mathbf{d}_{j_0}$, $\Omega_{j_0} = \left\{i | 1 \le i \le N, \mathbf{x}_i^{(k)}[j_0] \ne 0\right\}$.
   2.2. Compute the overall representation error matrix, $\mathbf{E}_{j_0}$:
   $$\mathbf{E}_{j_0} = \mathbf{Y} - \sum_{j\ne j_0} \mathbf{d}_j \mathbf{x}_T^{(k)j},$$
   where $\mathbf{x}_T^{(k)j}$ are the j'th **rows** of matrix $\mathbf{X}^{(k)}$.
   2.3. Restrict $\mathbf{E}_{j_0}$ by choosing only the columns corresponding to Ω_{j_0}, and obtain $\mathbf{E}_{j_0}^R$.
   2.4. Apply SVD decomposition $\mathbf{E}_{j_0}^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$. Update the dictionary atom $\mathbf{d}_{j_0} = \mathbf{u}_1$, and the representations by $\mathbf{x}_T^{(k)Rj_0} = \mathbf{\Delta}[1,1]\mathbf{v}_1$.
3. $\mathbf{D}^{(k)} \leftarrow \mathbf{D}$, $k \leftarrow k + 1$.

2.1 Discriminative K-SVD (D-KSVD)

To overcome the sub-optimality of the K-SVD algorithm for classification discussed above, [17] proposes to incorporate the classification error term directly into the dictionary learning formulation in Eq. 1, causing the K-SVD algorithm to simultaneously learn the dictionary and the classifier. The authors formulate the joint dictionary-classifier learning problem as follows:

$$\begin{aligned}\langle \mathbf{D}^*, \mathbf{W}^*, \mathbf{X}^* \rangle &= \underset{\mathbf{D},\mathbf{W},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \gamma \|\mathbf{H} - \mathbf{W}\mathbf{X}\|_F^2 \\ &\quad \text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0 \\ &= \underset{\mathbf{D},\mathbf{W},\mathbf{X}}{\operatorname{argmin}} \left\| \begin{bmatrix}\mathbf{Y} \\ \sqrt{\gamma}\mathbf{H}\end{bmatrix} - \begin{bmatrix}\mathbf{D} \\ \sqrt{\gamma}\mathbf{W}\end{bmatrix}\mathbf{X} \right\|_F^2 \\ &\quad \text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0,\end{aligned} \tag{$P_{\text{D-KSVD}}$}$$

where γ is a regularization parameter balancing the contribution of the classification error to the overall objective. The

authors show experimentally that this indeed increases the discriminative properties of the resulting classifier. The D-KSVD algorithm is summarized in Algorithm 2.

Algorithm 2 Discriminative K-SVD

Input: $\mathbf{Y} \in \mathbb{R}^{n\times N}$, $\mathbf{H} \in \mathbb{R}^{m\times N}$, γ, T_0.
Output: $\mathbf{D} \in \mathbb{R}^{n\times K}$, $\mathbf{W} \in \mathbb{R}^{m\times K}$, $\mathbf{X} \in \mathbb{R}^{K\times N}$.

1. **Initialize:**
   1.1. Compute $\mathbf{D}^{(0)}$ using an initialization scheme of choice, e.g., by concatenating class-specific dictionaries found with K-SVD.
   1.2. Compute $\mathbf{X}^{(0)}$ for $\mathbf{Y}$ and $\mathbf{D}^{(0)}$ using sparse coding.
   1.3. Compute $\mathbf{W}^{(0)}$ using Eq. (3) for $\lambda = 1$:
   $$\mathbf{W}^{(0)} = \mathbf{H}\mathbf{z}^{(0)},$$
   where $\mathbf{z}^{(0)} = {\mathbf{X}^{(0)}}^T(\mathbf{X}^{(0)}{\mathbf{X}^{(0)}}^T + \mathbf{I})^{-1}$.
2. **K-SVD:** Solve $P_{\text{D-KSVD}}$; use $\left({\mathbf{D}^{(0)}}^T, \sqrt{\gamma}{\mathbf{W}^{(0)}}^T\right)^T$ to initialize the dictionary.
3. **Normalize:**
   3.1. $\mathbf{D} \leftarrow \left\{\frac{\mathbf{d}_1}{\|d_1\|_2}, \frac{\mathbf{d}_2}{\|\mathbf{d}_2\|_2}, \ldots, \frac{\mathbf{d}_K}{\|\mathbf{d}_K\|_2}\right\}$
   3.2. $\mathbf{W} \leftarrow \left\{\frac{\mathbf{w}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{w}_2}{\|\mathbf{d}_2\|_2}, \ldots, \frac{\mathbf{w}_K}{\|\mathbf{d}_K\|_2}\right\}$

2.2 Label Consistent K-SVD (LC-KSVD)

In a follow-up work by Jiang et al. [10] the authors propose to incorporate a *discriminative sparse-code error* term enforcing label consistency, encouraging similarity among sparse representations of signals belonging to the same class, into the D-KSVD formulation $P_{\text{D-KSVD}}$. The authors claimed that this additional term improves the accuracy of the linear classifier obtained by D-KSVD. The optimization problem posed by LC-KSVD is:

$$\begin{aligned}\langle \mathbf{D}^*, \mathbf{W}^*, \mathbf{A}^*, \mathbf{X}^*\rangle &= \underset{\mathbf{D},\mathbf{W},\mathbf{A},\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \alpha\|\mathbf{Q} - \mathbf{A}\mathbf{X}\|_F^2 \\ &\qquad + \beta\|\mathbf{H} - \mathbf{W}\mathbf{X}\|_F^2 \\ &\text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0 \\ &= \underset{\mathbf{D},\mathbf{W},\mathbf{A},\mathbf{X}}{\operatorname{argmin}} \left\| \begin{bmatrix}\mathbf{Y}\\ \sqrt{\alpha}\mathbf{Q}\\ \sqrt{\beta}\mathbf{H}\end{bmatrix} - \begin{bmatrix}\mathbf{D}\\ \sqrt{\alpha}\mathbf{A}\\ \sqrt{\beta}\mathbf{W}\end{bmatrix}\mathbf{X}\right\|_F^2 \\ &\text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0,\end{aligned} \qquad (P_{\text{LC-KSVD}})$$

where $\mathbf{Q} \in \mathbb{R}^{K\times N}$ is the *discriminative sparse codes* matrix promoting label consistency, $\mathbf{A} \in \mathbb{R}^{K\times K}$ is a linear transformation, and α and β are the regularization parameters balancing the classification and the discriminative sparse-code errors contribution to the overall objective, respectively. The LC-KSVD algorithm is summarized in Algorithm 3.

The authors of [10] proposed to allocate dictionary atoms to classes uniformly – p atoms for each one of the m classes. Assuming that k training samples for each class are provided, the label consistency matrix $\mathbf{Q}$ has the following block structure:

$$\mathbf{Q} = \begin{bmatrix}\mathbf{1} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}\end{bmatrix},$$

where $\mathbf{1} \equiv \mathbf{1}_{p\times k}$ and $\mathbf{0} \equiv \mathbf{0}_{p\times k}$. For example, for $m = 3$, $p = 2$, $k = 2$ ($K = mp = 6$, $N = mk = 6$),

$$\mathbf{Q} = \begin{bmatrix}1&1&0&0&0&0\\1&1&0&0&0&0\\0&0&1&1&0&0\\0&0&1&1&0&0\\0&0&0&0&1&1\\0&0&0&0&1&1\end{bmatrix}.$$

Algorithm 3 Label Consistent K-SVD

Input: $\mathbf{Y} \in \mathbb{R}^{n\times N}$, $\mathbf{Q} \in \mathbb{R}^{K\times N}$, $\mathbf{H} \in \mathbb{R}^{m\times N}$, α, β, T_0.
Output: $\mathbf{D} \in \mathbb{R}^{n\times K}$, $\mathbf{A} \in \mathbb{R}^{K\times K}$, $\mathbf{W} \in \mathbb{R}^{m\times K}$, $\mathbf{X} \in \mathbb{R}^{K\times N}$.

1. **Initialize:**
   1.1. Compute $\mathbf{D}^{(0)}$ using an initialization scheme of choice, e.g., by concatenating class-specific dictionaries found with K-SVD as reported in [10]
   1.2. Compute $\mathbf{X}^{(0)}$ for $\mathbf{Y}$ and $\mathbf{D}^{(0)}$ using sparse coding.
   1.3. Compute $\mathbf{A}^{(0)}$ using Eq. (3) for $\lambda = 1$:
   $$\mathbf{A}^{(0)} = \mathbf{Q}\mathbf{z}^{(0)},$$
   where $\mathbf{z}^{(0)} = {\mathbf{X}^{(0)}}^T(\mathbf{X}^{(0)}{\mathbf{X}^{(0)}}^T + \mathbf{I})^{-1}$.
   1.4. Compute $\mathbf{W}^{(0)}$ using Eq. (3) for $\lambda = 1$:
   $$\mathbf{W}^{(0)} = \mathbf{H}\mathbf{z}^{(0)}.$$
2. **K-SVD:** Solve $P_{\text{LC-KSVD}}$; use $\left({\mathbf{D}^{(0)}}^T, \sqrt{\alpha}{\mathbf{A}^{(0)}}^T, \sqrt{\beta}{\mathbf{W}^{(0)T}}\right)^T$ to initialize the dictionary.
3. **Normalize:**
   3.1. $\mathbf{D} \leftarrow \left\{\frac{\mathbf{d}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{d}_2}{\|\mathbf{d}_2\|_2}, \ldots, \frac{\mathbf{d}_K}{\|\mathbf{d}_K\|_2}\right\}$
   3.2. $\mathbf{A} \leftarrow \left\{\frac{\mathbf{a}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{a}_2}{\|\mathbf{d}_2\|_2}, \ldots, \frac{\mathbf{a}_K}{\|\mathbf{d}_K\|_2}\right\}$
   3.3. $\mathbf{W} \leftarrow \left\{\frac{\mathbf{w}_1}{\|\mathbf{d}_1\|_2}, \frac{\mathbf{w}_2}{\|\mathbf{d}_2\|_2}, \ldots, \frac{\mathbf{w}_K}{\|\mathbf{d}_K\|_2}\right\}$

2.3 Dictionary Initialization

Both D-KSVD and LC-KSVD require an initialization step to set the initial dictionary $\mathbf{D}^{(0)}$ (step 1.1 in Algorithms 2 and 3). D-KSVD learns a single dictionary using data from all m classes while LC-KSVD learns m class-specific dictionaries which are later concatenated into a single dictionary. In both cases the learning is performed using the original K-SVD where the dictionary is initialized with the actual data samples and not randomly as in the original K-SVD. While the initialization steps of these two algorithms is different, and it is not certain whether or not any of them is preferable in general, for the sake of our proof we only assume that both algorithms share the same initialization step. Besides being identical, no other constraints are imposed on the algorithms' initialization steps (for example, our proof holds even when random initialization is used).

In all our experiments, reported in Section 4, we chose to use the initialization step of LC-KSVD.

3 Proof of Equivalence between the LC-KSVD and the D-KSVD Algorithms

We now show that the problem $P_{\text{LC-KSVD}}$ is identical to $P_{\text{D-KSVD}}$ for a proper choice of the regularization parameter γ. We assume that the initialization steps of both algorithms (step 1.1) are identical and that LC-KSVD uses the uniform atom allocation scheme described in [10][1].

Theorem 3.1. *Let us assume that both Algorithms 2 and 3 initialize $\mathbf{D}^{(0)}$ with an identical dictionary $\mathcal{D}$ (step 1.1). Let $\langle \mathbf{D}^*, \mathbf{W}^*, \mathbf{A}^*, \mathbf{X}^* \rangle$ be the solution of $P_{\text{LC-KSVD}}$ (step 2 of Algorithm 3) for $\mathbf{Y} \in \mathbb{R}^{n\times N}$, $\mathbf{Q} \in \mathbb{R}^{K\times N}$, $\mathbf{H} \in \mathbb{R}^{m\times N}$, α, β and T_0, where n is the sample dimension, N is the number of training samples, m is the number of classes, $K = mp$ is the dictionary size while p is the number of dictionary atoms allocated per class, then $\langle \mathbf{D}^*, \mathbf{W}^*, \mathbf{X}^* \rangle$ is the solution of $P_{\text{D-KSVD}}$ (step 2 of Algorithm 2) for $\mathbf{Y}$, $\mathbf{H}$, γ and T_0, where $\gamma = p\alpha + \beta$.*

Proof. First we reformulate $P_{\text{LC-KSVD}}$ into a form facilitating our proof. Let us define a permutation over $\mathbf{Q}$'s row indices, $\pi : \{1, \ldots, mp\} \to \{1, \ldots, mp\}$, as:

$$\pi(i) = ((i-1) \mod m)p + \left\lfloor \frac{i-1}{m} \right\rfloor + 1.$$

We now define the "reshuffled" matrix $\mathbf{Q}$ as $\tilde{\mathbf{Q}} \equiv \mathbf{P}_\pi \mathbf{Q}$, where $\mathbf{P}_\pi$ is the permutation matrix corresponding to permutation π. The purpose of this permutation is to reshuffle the rows of $\mathbf{Q}$ so that it has the form $\tilde{\mathbf{Q}} = \left(\underbrace{\mathbf{H}^T, \ldots, \mathbf{H}^T}_{\times p} \right)^T$. Thus, for the matrix $\mathbf{Q}$ given in the example in Section 2.2:

$$\tilde{\mathbf{Q}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H} \end{bmatrix}.$$

Define the matrix $\mathbf{P} \in \mathbb{R}^{(n+K+m)\times K}$ as:

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_\pi & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_m \end{bmatrix},$$

where $\mathbf{I}_n$ and $\mathbf{I}_m$ are the identity matrices of size n and m, respectively. Since $\mathbf{P}$ is orthonormal, the following holds[2]:

$$\begin{aligned} &\operatorname*{argmin}_{\mathbf{D},\mathbf{W},\mathbf{A},\mathbf{X}} \left\| \begin{bmatrix} \mathbf{Y} \\ \sqrt{\alpha}\mathbf{Q} \\ \sqrt{\beta}\mathbf{H} \end{bmatrix} - \begin{bmatrix} \mathbf{D} \\ \sqrt{\alpha}\mathbf{A} \\ \sqrt{\beta}\mathbf{W} \end{bmatrix} \mathbf{X} \right\|_F^2 = \\ &\text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0 \\ &\operatorname*{argmin}_{\mathbf{D},\mathbf{W},\mathbf{A},\mathbf{X}} \left\| \mathbf{P} \begin{bmatrix} \mathbf{Y} \\ \sqrt{\alpha}\mathbf{Q} \\ \sqrt{\beta}\mathbf{H} \end{bmatrix} - \mathbf{P} \begin{bmatrix} \mathbf{D} \\ \sqrt{\alpha}\mathbf{A} \\ \sqrt{\beta}\mathbf{W} \end{bmatrix} \mathbf{X} \right\|_F^2 = \\ &\text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0 \\ &\operatorname*{argmin}_{\mathbf{D},\mathbf{W},\mathbf{A},\mathbf{X}} \left\| \begin{bmatrix} \mathbf{Y} \\ \sqrt{\alpha}\tilde{\mathbf{Q}} \\ \sqrt{\beta}\mathbf{H} \end{bmatrix} - \begin{bmatrix} \mathbf{D} \\ \sqrt{\alpha}\tilde{\mathbf{A}} \\ \sqrt{\beta}\mathbf{W} \end{bmatrix} \mathbf{X} \right\|_F^2 \\ &\text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0, \end{aligned}$$

where $\tilde{\mathbf{Q}} \equiv \mathbf{P}_\pi \mathbf{Q} = \left(\underbrace{\mathbf{H}^T, \ldots, \mathbf{H}^T}_{\times p} \right)^T$ and $\tilde{\mathbf{A}} \equiv \mathbf{P}_\pi \mathbf{A}$.

We can now reformulate $P_{\text{LC-KSVD}}$ as follows:

$$\begin{aligned} \left\langle \tilde{\mathbf{D}}^*, \mathbf{X}^* \right\rangle = &\operatorname*{argmin}_{\tilde{\mathbf{D}},\mathbf{X}} \left\| \tilde{\mathbf{Y}} - \tilde{\mathbf{D}}\mathbf{X} \right\|_F^2 \\ &\text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0, \end{aligned} \qquad (\tilde{P}_{\text{LC-KSVD}})$$

where

$$\tilde{\mathbf{Y}} = \left(\mathbf{Y}^T, \underbrace{\sqrt{\alpha}\mathbf{H}^T, \ldots, \sqrt{\alpha}\mathbf{H}^T}_{\times p}, \sqrt{\beta}\mathbf{H}^T \right)^T$$

and

$$\tilde{\mathbf{D}} = \left(\mathbf{D}^T, \sqrt{\alpha}\tilde{\mathbf{A}}^T, \sqrt{\beta}\mathbf{W}^T \right)^T.$$

Since the problems $P_{\text{LC-KSVD}}$ and $\tilde{P}_{\text{LC-KSVD}}$ are equivalent, $\langle \mathbf{D}^*, \mathbf{W}^*, \mathbf{A}^*, \mathbf{X}^* \rangle$ is the solution of $P_{\text{LC-KSVD}}$ if and only if $\left\langle \tilde{\mathbf{D}}^*, \mathbf{X}^* \right\rangle$ is the solution of $\tilde{P}_{\text{LC-KSVD}}$, where $\tilde{\mathbf{D}}^* = \left(\mathbf{D}^{*T}, \sqrt{\alpha}\tilde{\mathbf{A}}^{*T}, \sqrt{\beta}\mathbf{W}^{*T} \right)^T$ and $\tilde{\mathbf{A}}^* \equiv \mathbf{P}_\pi \mathbf{A}^*$. From Lemma 3.2 (described below) it follows that

$$\tilde{\mathbf{D}}^* = \left(\mathbf{D}^{*T}, \underbrace{\sqrt{\alpha}\mathbf{W}^{*T}, \ldots, \sqrt{\alpha}\mathbf{W}^{*T}}_{\times p}, \sqrt{\beta}\mathbf{W}^{*T} \right)^T,$$

meaning that $\mathbf{A}$ is a redundant variable in $P_{\text{LC-KSVD}}$. Thus, for the given $\mathbf{Y}$, $\mathbf{Q}$, $\mathbf{H}$, α, β and T_0:

$$\begin{aligned} \langle \mathbf{D}^*, \mathbf{W}^*, \mathbf{X}^* \rangle = &\operatorname*{argmin}_{\mathbf{D},\mathbf{W},\mathbf{X}} \left\| \begin{bmatrix} \mathbf{Y} \\ \sqrt{\alpha}\mathbf{H} \\ \vdots \\ \sqrt{\alpha}\mathbf{H} \\ \sqrt{\beta}\mathbf{H} \end{bmatrix} - \begin{bmatrix} \mathbf{D} \\ \sqrt{\alpha}\mathbf{W} \\ \vdots \\ \sqrt{\alpha}\mathbf{W} \\ \sqrt{\beta}\mathbf{W} \end{bmatrix} \mathbf{X} \right\|_F^2 \\ &\text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0 \\ \overset{\gamma = p\alpha+\beta}{=} &\operatorname*{argmin}_{\mathbf{D},\mathbf{W},\mathbf{X}} \left\| \begin{bmatrix} \mathbf{Y} \\ \sqrt{\gamma}\mathbf{H} \end{bmatrix} - \begin{bmatrix} \mathbf{D} \\ \sqrt{\gamma}\mathbf{W} \end{bmatrix} \mathbf{X} \right\|_F^2 \\ &\text{s.t. } \forall i,\ \|\mathbf{x}_i\|_0 < T_0, \end{aligned}$$

1. Though non-uniform allocation schemes are possible in theory, such extensions are beyond the scope of this work as well as the original LC-KSVD paper [10]. In practice the vast majority of works use LC-KSVD with uniform allocation, as described in the original paper. However, we note that D-KSVD can achieve similar effects as the non-uniform atom allocation of LC-KSVD by replacing the L_2 classification error regularizer γ by Tikhonov regularization matrix $\mathbf{\Gamma}$, assigning different weight to classification errors resulting from instances belonging to different classes. As with non-uniform extensions to LC-KSVD, the classification performance of such schemes must be evaluated. This is beyond the scope of this work.

2. Recall that the Frobenius norm is invariant under unitary transformations: $\|\mathbf{PA}\|_F = \|\mathbf{A}\|_F$.

which is exactly the definition of the $P_{\text{D-KSVD}}$ problem. □

Lemma 3.2. *If* $\tilde{\mathbf{D}}^* = \left(\mathbf{D}^{*T}, \sqrt{\alpha}\tilde{\mathbf{A}}^{*T}, \sqrt{\beta}\mathbf{W}^{*T}\right)^T$ *is the dictionary obtained by solving* $\tilde{P}_{\text{LC-KSVD}}$ *using* $\tilde{\mathbf{D}}^{(0)} = \left(\mathbf{D}^{(0)^T}, \sqrt{\alpha}\tilde{\mathbf{A}}^{(0)T}, \sqrt{\beta}\mathbf{W}^{(0)^T}\right)^T$ *as the initial dictionary where* $\mathbf{D}^{(0)}$ *is set to* $\mathcal{D}$ *using an initialization scheme of choice, and* $\mathbf{A}^{(0)}$ *and* $\mathbf{W}^{(0)}$ *are obtained using steps 1.2–1.4 of Algorithm 3,* $\tilde{\mathbf{A}}^* \equiv \mathbf{P}_\pi\mathbf{A}^*$, $\tilde{\mathbf{A}}^{(0)} \equiv \mathbf{P}_\pi\mathbf{A}^{(0)}$ *and* $\mathbf{P}_\pi$ *is the permutation matrix defined in Theorem 3.1, then* $\tilde{\mathbf{A}}^*$ *has the following structure:*

$$\tilde{\mathbf{A}}^* = \left(\underbrace{\mathbf{W}^{*T}, \ldots, \mathbf{W}^{*T}}_{\times p}\right)^T.$$

Proof. We prove by induction on k, the K-SVD's iteration number, that the dictionary obtained by the K-SVD algorithm (Algorithm 1) is of the form $\tilde{\mathbf{D}}^{(k)} = \left(\mathbf{D}^{(k)^T}, \sqrt{\alpha}\tilde{\mathbf{A}}^{(k)T}, \sqrt{\beta}\mathbf{W}^{(k)^T}\right)^T$, where $\tilde{\mathbf{A}}^{(k)} = \left(\underbrace{\mathbf{W}^{(k)^T}, \ldots, \mathbf{W}^{(k)^T}}_{\times p}\right)^T$.

Basis. For $k = 0$,

$$\tilde{\mathbf{A}}^{(0)} = \mathbf{P}_\pi\mathbf{A}^{(0)} = \mathbf{P}_\pi\mathbf{Q}\mathbf{z}_0 = \tilde{\mathbf{Q}}\mathbf{z}_0 = \\ = \left(\underbrace{(\mathbf{H}\mathbf{z}_0)^T, \ldots, (\mathbf{H}\mathbf{z}_0)^T}_{\times p}\right)^T = \left(\underbrace{\mathbf{W}^{(0)^T}, \ldots, \mathbf{W}^{(0)^T}}_{\times p}\right)^T,$$

due to the initialization step of the LC-KSVD algorithm (see steps 1.3, 1.4 of Algorithm 3).

Iteration Step. Assuming that:

$$\tilde{\mathbf{D}}^{(k-1)} = (\mathbf{D}^{(k-1)^T}, \sqrt{\alpha}\mathbf{W}^{(k-1)^T}, \ldots, \sqrt{\alpha}\mathbf{W}^{(k-1)^T}, \\ \sqrt{\beta}\mathbf{W}^{(k-1)^T})^T$$

we show that:

$$\tilde{\mathbf{D}}^{(k)} = \left(\mathbf{D}^{(k)^T}, \sqrt{\alpha}\mathbf{W}^{(k)^T}, \ldots, \sqrt{\alpha}\mathbf{W}^{(k)^T}, \sqrt{\beta}\mathbf{W}^{(k)^T}\right)^T,$$

by considering the k'th iteration step of the K-SVD algorithm.

Sparse coding step (step 1 in Algorithm 1). Let $\tilde{\mathbf{X}}^{(k)}$ denote the sparse codes obtained using any pursuit algorithm, such as the OMP [3] algorithm:

$$\tilde{\mathbf{X}}^{(k)} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\|\tilde{\mathbf{Y}} - \tilde{\mathbf{D}}^{(k-1)}\mathbf{X}\right\|_F^2, \ s.t.\ \forall i, \ \|\mathbf{x}_i\|_0 \le T.$$

K-SVD Dictionary Update Step (step 2 in Algorithm 1). For $j_0 = 1, 2, \ldots, K$:
Let $\mathbf{E}_{j_0} = \tilde{\mathbf{Y}} - \sum_{j \ne j_0} \tilde{\mathbf{d}}_j^{(k-1)}\tilde{\mathbf{x}}_T^{(k)j}$ where $\tilde{\mathbf{x}}_T^{(k)j}$ is the j'th row of matrix $\tilde{\mathbf{X}}^{(k)}$. Thus,

$$\mathbf{E}_{j_0} = \left(\mathbf{E}_Y{}^T, \underbrace{\sqrt{\alpha}\mathbf{E}_H{}^T, \ldots, \sqrt{\alpha}\mathbf{E}_H{}^T}_{\times p}, \sqrt{\beta}\mathbf{E}_H{}^T\right)^T,$$

where

$$\mathbf{E}_Y = \mathbf{Y} - \sum_{j \ne j_0} \mathbf{d}_j^{(k-1)}\tilde{\mathbf{x}}_j^{(k)T}$$

and

$$\mathbf{E}_H = \mathbf{H} - \sum_{j \ne j_0} \mathbf{w}_j^{(k-1)}\tilde{\mathbf{x}}_j^{(k)T}.$$

Let $\mathbf{E}_{j_0}^R$ denote the restriction (sub-matrix) of $\mathbf{E}_{j_0}$ obtained by step 2.3 of Algorithm 1. Applying SVD decomposition $\mathbf{E}_{j_0}^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$ and using Lemma 3.3 (described below) we get that:

$$\tilde{\mathbf{d}}_{j_0}^{(k)} = \mathbf{u}_1 = \left(\mathbf{d}_{j_0}^{(k)^T}, \underbrace{\sqrt{\alpha}\mathbf{w}_{j_0}^{(k)^T}, \ldots, \sqrt{\alpha}\mathbf{w}_{j_0}^{(k)^T}}_{\times p}, \sqrt{\beta}\mathbf{w}_{j_0}^{(k)^T}\right)^T.$$

Eventually, after updating K dictionary atoms,

$$\tilde{\mathbf{D}}^{(k)} = \left(\mathbf{D}^{(k)^T}, \sqrt{\alpha}\mathbf{W}^{(k)^T}, \ldots, \sqrt{\alpha}\mathbf{W}^{(k)^T}, \sqrt{\beta}\mathbf{W}^{(k)^T}\right)^T,$$

concluding our proof by induction.

We proved that at the end of the final iteration of the K-SVD algorithm, the solution is of the following form:

$$\tilde{\mathbf{D}}^* = \left(\mathbf{D}^{*T}, \sqrt{\alpha}\mathbf{W}^{*T}, \ldots, \sqrt{\alpha}\mathbf{W}^{*T}, \sqrt{\beta}\mathbf{W}^{*T}\right)^T$$

and since we assumed that the solution is of the form $\left(\mathbf{D}^{*T}, \sqrt{\alpha}\tilde{\mathbf{A}}^{*T}, \sqrt{\beta}\mathbf{W}^{*T}\right)^T$, we conclude that[3] :

$$\tilde{\mathbf{A}}^* = \left(\underbrace{\mathbf{W}^{*T}, \ldots, \mathbf{W}^{*T}}_{\times p}\right)^T.$$

□

Lemma 3.3. *Let* $\mathbf{u}_1$ *denote the first left singular vector of matrix* $\mathbf{A} = \left(\mathbf{D}^T, \mathbf{B}^T, a\mathbf{B}^T\right)^T$, *where* $\mathbf{D} \in \mathbb{R}^{n \times N}$, $\mathbf{B} \in \mathbb{R}^{m \times N}$ *and* $a > 0$, *then* $\mathbf{u}_1 = \left(\mathbf{d}^T, \mathbf{u}^T, a\mathbf{u}^T\right)^T$, *where* $\mathbf{d} \in \mathbb{R}^n$ *and* $\mathbf{u} \in \mathbb{R}^m$.

Proof. The proof is based on the *power method* [9]. Let $\mathbf{v}_1^{(0)} \in \mathbb{R}^n$ and $\mathbf{v}_2^{(0)}, \mathbf{v}_3^{(0)} \in \mathbb{R}^m$ denote arbitrary vectors and let $\mathbf{v}^{(0)} \equiv \left(\mathbf{v}_1^{(0)^T}, \mathbf{v}_2^{(0)^T}, \mathbf{v}_3^{(0)^T}\right)^T$. Let us premultiply $\mathbf{v}^{(0)}$ by $\mathbf{A}\mathbf{A}^T$ from the left to obtain $\mathbf{v}^{(1)} \equiv \left(\mathbf{v}_1^{(1)^T}, \mathbf{v}_2^{(1)^T}, \mathbf{v}_3^{(1)^T}\right)^T$:

$$\mathbf{v}^{(1)} = \mathbf{A}\mathbf{A}^T\mathbf{v}^{(0)} = \begin{bmatrix} \mathbf{D}\mathbf{D}^T & \mathbf{D}\mathbf{B}^T & a\mathbf{D}\mathbf{B}^T \\ \mathbf{B}\mathbf{D}^T & \mathbf{B}\mathbf{B}^T & a\mathbf{B}\mathbf{B}^T \\ a\mathbf{B}\mathbf{D}^T & a\mathbf{B}\mathbf{B}^T & a^2\mathbf{B}\mathbf{B}^T \end{bmatrix}\mathbf{v}^{(0)} \\ = \begin{bmatrix} \mathbf{D}\mathbf{D}^T\mathbf{v}_1 + \mathbf{D}\mathbf{B}^T\mathbf{v}_2 + a\mathbf{D}\mathbf{B}^T\mathbf{v}_3 \\ \mathbf{B}\mathbf{D}^T\mathbf{v}_1 + \mathbf{B}\mathbf{B}^T\mathbf{v}_2 + a\mathbf{B}\mathbf{B}^T\mathbf{v}_3 \\ a\left(\mathbf{B}\mathbf{D}^T\mathbf{v}_1 + \mathbf{B}\mathbf{B}^T\mathbf{v}_2 + a\mathbf{B}\mathbf{B}^T\mathbf{v}_3\right) \end{bmatrix}.$$

Note that $\mathbf{v}_3^{(1)} = a\mathbf{v}_2^{(1)}$. Repeating the process to obtain $\mathbf{v}^{(2)} = \mathbf{A}\mathbf{A}^T\mathbf{v}^{(1)}$ preserves this property and therefore $\mathbf{v}_3^{(k)} = a\mathbf{v}_2^{(k)}$ for all values of k. From the power method we know that for $k \to \infty$, $\mathbf{v}^{(k)}$ is the first eigenvector of $\mathbf{A}\mathbf{A}^T$ and therefore is also the first left singular vector of $\mathbf{A}$. Thus, $\mathbf{u}_1 = \left(\mathbf{d}^T, \mathbf{u}^T, a\mathbf{u}^T\right)^T$, where $\mathbf{d} = \mathbf{v}_1^{(k)}$ and $\mathbf{u} = \mathbf{v}_2^{(k)}$. □

Corollary 3.4. *Given D-KSVD that uses the same initialization scheme for* $\mathbf{D}^{(0)}$ *in step 1.1 as reported in the original paper by*

3. Note that the normalization step of the LC-KSVD algorithm (see step 3 in Algorithm 3) scales each one of the atoms independently of each other and therefore does not have any impact on the above analysis.

Jiang et al. [10], i.e., the concatenation of class-specific dictionaries (see Section 2.3), and given that LC-KSVD allocates an equal number of p atoms per class and $\gamma = p\alpha + \beta$, the outputs of D-KSVD and that of the original LC-KSVD [10] are identical.

4 Empirical Validation

To verify the derivation presented in Section 3, we repeated the LC-KSVD and D-KSVD comparison using two of the datasets (the YaleB face recognition dataset [8] and the Caltech101 dataset [7]) and parameter values from [10], with $\gamma = p\alpha + \beta$. Across all experiments D-KSVD obtained the exact same dictionaries and classifiers as LC-KSVD, up to numeric precision (10^{-13}).

LC-KSVD adds $p \times m$ rows to the input matrix of the K-SVD step, compared to D-KSVD. To assess the additional computational burden, caused by this addition of rows, we measured the training phase runtime for dictionaries of various sizes (K) using the code provided by the authors of [10]. Figure 1 presents these results for the Caltech101 dataset. Similarly to [10] we trained dictionaries of sizes $102 \times p$ for $p \in \{5, 10, 15, 20, 25, 30\}$. The runtime was measured on a 2.60GHz Intel Core i7 machine. As can be seen from Figure 1 the advantage of D-KSVD is especially measurable for high values of p, where one can save as much as $40 - 45\%$ of the runtime in the training phase.

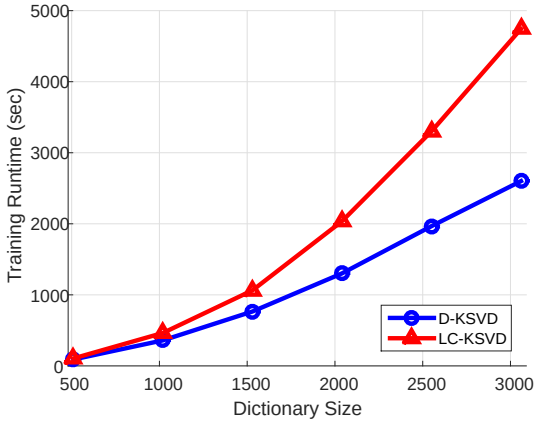

Fig. 1. The runtime of LC-KSVD compared to D-KSVD for dictionaries of various size.

Another advantage of D-KSVD is that only a single regularization parameter, γ, has to be determined by cross-validation as opposed to α and β for LC-KSVD, reducing the computational burden from a 2D grid search to a 1D line search.

5 Conclusions and Future Work

In this work we mathematically proved the equivalence of the LC-KSVD and the D-KSVD algorithms up to a proper choice of regularization parameters, for which we give a closed form expression. Our empirical evaluation validates this result, and shows that D-KSVD has superior run time.

We conclude that the D-KSVD algorithm is preferable due to its simplicity and computational efficiency, compared to the LC-KSVD algorithm. Future work should validate that "label consistency" indeed facilitates learning linear classifiers based on sparse representations, and, if so, develop more effective ways to incorporate the "label consistency" terms into the objective.

Acknowledgements

The authors wish to thank Z. Jiang, Z. Lin, and L. S. Davis for publishing their code and data.

References

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

[2] S. Cai, W. Zuo, L. Zhang, X. Feng, and P. Wang. Support vector guided dictionary learning. In *Proc. Eurpoean Conf. on Computer Vision (ECCV)*, pages IV: 624–639, 2014.

[3] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, 1997.

[4] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[5] M. Elad, J.-L. Starck, P. Querre, and D. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, 19(3):340 – 358, 2005.

[6] K. Engan, S. O. Aase, and J. H. Husøy. Multi-frame compression: Theory and design. *Signal Process.*, 80(10):2121–2140, 2000.

[7] 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. 2004.

[8] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[10] 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*, 35(11):2651–2664, 2013.

[11] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[12] V. Patel, Y. Chen, R. Chellappa, and P. Phillips. Dictionaries for image and video-based face recognition. *Journal of the Optical Society of America A*, 31(5):1090–1103, May 2014.

[13] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[14] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[15] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.

[16] M. Yang, D. Dai, L. Shen, and L. Van Gool. Latent dictionary learning for sparse representation based classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4138–4145, 2014.

[17] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2691–2698, 2010.