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.

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-KSVD are commonly treated as two different algorithms. Indeed, many recent publications on this subject concludes that the major 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 learning formulation in Eq. 1, causing the K-SVD algorithm to simultaneously learn the dictionary and the classifier. Indeed, many recent publications on face classification error term directly into 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: $Y \in \mathbb{R}^{n \times N}$, $D^{(0)} \in \mathbb{R}^{n \times K}$, $T_0$.
Output: $D^{(k)} \in \mathbb{R}^{n \times K}$, $X^{(k)} \in \mathbb{R}^{n \times N}$.
Main Iteration: Repeat until convergence

1. Sparse Coding Stage: Use any pursuit algorithm (e.g., OMP [3]) to compute the representation vectors $x_i$ for each example $y_i$, by approximating the solution of

$$
\hat{x}_i = \arg\min_{x_i} \|y_i - D^{(k-1)} x_i\|_2^2, \text{ s.t. } \|x_i\|_0 \leq T_0,
$$

for $i = 1, 2, \ldots, N$.

2. K-SVD Dictionary-Update Stage: Update each column $j_0 = 1, 2, \ldots, K$ in $D$ by $D^{(k-1)}$:

$$
D^{(k)} j_0 = \arg\min_{j_0} \|Y_{j_0} - D^{(k-1)} X\|_F^2, \text{ s.t. } \Omega_{j_0} \neq 0.
$$

2.1. Define the group of example signals that use the atom $d_{j_0}$, $\Omega_{j_0} = \{i | 1 \leq i \leq N, x_i^{(k)}(j_0) \neq 0\}$.

2.2. Compute the overall representation error matrix, $E_{j_0}$:

$$
E_{j_0} = Y - \sum_{j \neq j_0} d_j x_i^{(k)}(j),
$$

where $x_i^{(k)}(j)$ are the $j$'th rows of matrix $X^{(k)}$.

2.3. Restrict $E_{j_0}$ by choosing only the columns corresponding to $\Omega_{j_0}$, and obtain $E_{j_0}^r$.

2.4. Apply SVD decomposition $E_{j_0}^r = U \Delta V^T$. Update the dictionary atom $d_{j_0} = u_1$, and the representations by $x_i^{(k)}(j_0) = \Delta [1, 1] v_1$.

3. $D^{(k)} \leftarrow D$, $k \leftarrow k + 1$.

2.1 Discriminative K-SVD (D-KSV)

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:

$$
\langle D^*, W^*, X^* \rangle = \underset{D, W, X}{\arg\min} \|Y - DX\|_F^2 + \gamma \|H -WX\|_F^2,
$$

s.t. $\forall i$, $\|x_i\|_0 < T_0$.

$$
= \underset{D, W, X}{\arg\min} \|Y \sqrt{\gamma} H - \left( \frac{D}{\sqrt{\gamma}} \right) X\|_F^2,
$$

s.t. $\forall i$, $\|x_i\|_0 < T_0$.

(Problem $P_{D-KSVD}$)

where $\gamma$ is a regularization parameter balancing the contribution of the classification error to the overall objective. The
where $Q$ is the discriminative sparse codes matrix promoting label consistency, $A \in \mathbb{R}^{K \times N}$ is a linear transformation, and $\alpha$ and $\beta$ 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 $Q$ has the following block structure:

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

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

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

**Theorem 3.1**. Let us assume that both Algorithms 2 and 3 initialize $D^{(0)}$ with an identical dictionary $D$ (step 1.1). Let $(D^*, W^*, A^*, X^*)$ be the solution of $P_{\text{LC-KSVD}}$ (step 2 of Algorithm 3) for $Y \in \mathbb{R}^{n \times N}$, $Q \in \mathbb{R}^{K \times N}$, $H \in \mathbb{R}^{m \times N}$, $\alpha$, $\beta$, 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 $(D^*, W^*, A^*, X^*)$ is the solution of $P_{\text{D-KSVD}}$ (step 2 of Algorithm 2) for $Y$, $H$, $\gamma$ 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 $Q$’s row indices, $\pi : \{1, \ldots, mp\} \rightarrow \{1, \ldots, mp\}$, as:

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

We now define the “reshuffled” matrix $Q$ as $\tilde{Q} \equiv P_{\pi}Q$, where $P_{\pi}$ is the permutation matrix corresponding to permutation $\pi$. The purpose of this permutation is to reshuffle the rows of $Q$ so that it has the form

$$\tilde{Q} = \left( H^T, \ldots, H^T \right)^T_{xp}.$$

Thus, for the example given in Section 2.2:

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

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

$$P = \begin{bmatrix} I_n & 0 & 0 \\ 0 & P_{\pi} & 0 \\ 0 & 0 & I_m \end{bmatrix},$$

where $I_n$ and $I_m$ are the identity matrices of size $n$ and $m$, respectively. Since $P$ is orthonormal, the following holds:

$$\begin{align*}
\text{argmin}_{D, W, A, X} & \quad \left\| \begin{bmatrix} Y \sqrt{\alpha Q} \\ \sqrt{\beta H} \end{bmatrix} - \begin{bmatrix} D \sqrt{\alpha A} \\ \sqrt{\beta W} \end{bmatrix} X \right\|^2_F \\
\text{s.t.} & \quad \forall i, \|x_i\|_0 < T_0
\end{align*}$$

$$\begin{align*}
\text{argmin}_{D, W, A, X} & \quad \left\| \begin{bmatrix} Y \sqrt{\alpha Q} \\ \sqrt{\beta H} \end{bmatrix} - \begin{bmatrix} D \sqrt{\alpha A} \\ \sqrt{\beta W} \end{bmatrix} X \right\|^2_F \\
\text{s.t.} & \quad \forall i, \|x_i\|_0 < T_0
\end{align*}$$

and

$$\begin{align*}
\text{argmin}_{D, W, A, X} & \quad \left\| \begin{bmatrix} Y \sqrt{\alpha Q} \\ \sqrt{\beta H} \end{bmatrix} - \begin{bmatrix} D \sqrt{\alpha A} \\ \sqrt{\beta W} \end{bmatrix} X \right\|^2_F \\
\text{s.t.} & \quad \forall i, \|x_i\|_0 < T_0,
\end{align*}$$

where $Q = P_{\pi}Q = \left( H^T, \ldots, H^T \right)^T_{xp}$ and $A = P_{\pi}A$.

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

$$\begin{align*}
\langle \tilde{D}^*, \tilde{X}^* \rangle = & \quad \text{argmin}_{\tilde{D}, \tilde{X}} \left\| \tilde{Y} - \tilde{D} \tilde{X} \right\|^2_F \\
\text{s.t.} & \quad \forall i, \|x_i\|_0 < T_0,
\end{align*}$$

where

$$\tilde{Y} = \left( Y^T, \sqrt{\alpha H^T}, \ldots, \sqrt{\alpha H^T}, \sqrt{\beta H^T} \right)^T_{xp}$$

and

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

Since the problems $P_{\text{LC-KSVD}}$ and $\tilde{P}_{\text{LC-KSVD}}$ are equivalent, $(D^*, W^*, A^*, X^*)$ is the solution of $P_{\text{LC-KSVD}}$ if and only if $(\tilde{D}^*, \tilde{X}^*)$ is the solution of $\tilde{P}_{\text{LC-KSVD}}$, where

$$\tilde{D}^* = \left( D^{* T}, \sqrt{\alpha A^{* T}}, \sqrt{\beta W^{* T}} \right)^T_{xp}$$

and $A^* \equiv P_{\pi}A$. From Lemma 3.2 (described below) it follows that

$$\tilde{D}^* = \left( D^{* T}, \sqrt{\alpha A^{* T}}, \sqrt{\beta W^{* T}} \right)^T_{xp},$$

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

$$\begin{align*}
\langle D^*, W^*, X^* \rangle = & \quad \text{argmin}_{D, W, X} \left\| \begin{bmatrix} Y \sqrt{\alpha H} \\ \sqrt{\beta H} \end{bmatrix} - \begin{bmatrix} D \sqrt{\alpha W} \\ \sqrt{\beta W} \end{bmatrix} X \right\|^2_F \\
\text{s.t.} & \quad \forall i, \|x_i\|_0 < T_0
\end{align*}$$

and

$$\gamma = p\alpha + \beta = \text{argmin}_{D, W, X} \left\| \begin{bmatrix} Y \sqrt{\alpha H} \\ \sqrt{\beta H} \end{bmatrix} - \begin{bmatrix} D \sqrt{\alpha W} \\ \sqrt{\beta W} \end{bmatrix} X \right\|^2_F$$

s.t. $\forall i, \|x_i\|_0 < T_0$.

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 $\gamma$ by Tikhonov regularization matrix $\Gamma$, 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: $\|PA\|_F = \|A\|_F$. 
which is exactly the definition of the $P_{D-KSVD}$ problem. □

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

$$
A^* = \left( \begin{array}{c} 
W^* T, \ldots, W^* T 
\end{array} \right)_x. 
$$

**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

$$
\tilde{A}^{(k)} = \left( \begin{array}{c} 
\tilde{W}^{(k)} T, \ldots, \tilde{W}^{(k)} T 
\end{array} \right)_x,
$$

where $\tilde{W}^{(k)} = \left( W^{(k)} T, \ldots, W^{(k)} T \right)_x$.

**Basis.** For $k = 0$,

$$
\tilde{A}^{(0)} = P_x A^{(0)} = P_x Q z_0 = Q z_0 = 
$$

$$
= \left( (H z_0) T, \ldots, (H z_0) T \right)_x = \left( W^{(0)} T, \ldots, W^{(0)} T \right)_x,
$$

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

**Iteration Step.** Assuming that:

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

we show that:

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

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

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

$$
\tilde{X}^{(k)} = \arg \min \left\{ \| \tilde{Y} - \tilde{D}^{(k-1)} X \|_F^2 \right\}, \text{ s.t. } \forall i, \| x_i \|_0 \leq T.
$$

**K-SVD Dictionary Update Step (step 2 in Algorithm 1).**

For $j_0 = 1, 2, \ldots, K$:

Let $E_{j_0} = \tilde{Y} - \sum_{j \neq j_0} d^{(k-1)}_{X_{j}^{(k)}} \tilde{X}_{j}^{(k)}$ where $X_{j}^{(k)}$ is the $j$'th row of matrix $X^{(k)}$. Thus,

$$
E_{j_0} = \left( E_Y T, \sqrt{\alpha E_H T}, \ldots, \sqrt{\alpha E_H T}, \sqrt{\beta E_H T} \right)_x,
$$

where

$$
E_Y = \tilde{Y} - \sum_{j \neq j_0} d^{(k-1)}_{X_{j}^{(k)}} \tilde{X}_{j}^{(k)} T.
$$

and

$$
E_H = H - \sum_{j \neq j_0} w^{(k-1)}_{X_{j}^{(k)}} x^{(k)}_{j} T.
$$

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

$$
d^{(k)}_{j_0} = u_1 = \left( d^{(k)}_{j_0} T, \sqrt{\alpha w^{(k)}_{j_0} T}, \ldots, \sqrt{\alpha w^{(k)}_{j_0} T}, \sqrt{\beta w^{(k)}_{j_0} T} \right)_x.
$$

Eventually, after updating $K$ dictionary atoms,

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

and since we assumed that the solution is of the form $\left( D^*, \sqrt{\alpha A^* T}, \sqrt{\beta A^* T} \right)_x$, we conclude that

$$
A^* = \left( W^* T, \ldots, W^* T \right)_x.
$$

□

**Lemma 3.3.** Let $u_1$ denote the first left singular vector of matrix $A = (D^T, B^T, a B^T)$, where $D \in \mathbb{R}^{n \times N}$, $B \in \mathbb{R}^{m \times N}$ and $a > 0$, then $u_1 = (d^T, u^T, a u^T)^T$, where $d \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$.

**Proof.** The proof is based on the power method [9]. Let $v_1^{(0)} \in \mathbb{R}^n$ and $v_2^{(0)}, v_3^{(0)} \in \mathbb{R}^m$ denote arbitrary vectors and let

$$
v^{(0)} = (v_1^{(0)} T, v_2^{(0)} T, v_3^{(0)} T)^T.
$$

Let us premultiply $v^{(0)}$ by $A A^T$ from the left to obtain:

$$
v^{(1)} = (v_1^{(1)} T, v_2^{(1)} T, v_3^{(1)} T)^T :
$$

$$
v^{(1)} = A A^T v^{(0)} = \left[ \begin{array}{c} 
D D^T + a D B^T \\
B D^T + a B B^T \\
\alpha a B B^T 
\end{array} \right] v^{(0)}
$$

$$
= \left[ \begin{array}{c} 
D D^T v_1 + a B D^T v_2 + \alpha a B B^T v_3 \\
B D^T v_1 + a B B^T v_2 + \alpha a B B^T v_3 \\
\alpha a B B^T v_1 + \alpha a B B^T v_2 + \alpha a B B^T v_3 
\end{array} \right].
$$

Note that $v_3^{(1)} = a v_1^{(1)}$. Repeating the process to obtain $v^{(2)} = A A^T v^{(1)}$ preserves this property and therefore $v_3^{(k)} = a v_1^{(k)}$ for all values of $k$. From the power method we know that for $k \to \infty$, $v^{(k)}$ is the first eigenvector of $A A^T$ and therefore is also the first left singular vector of $A$. Thus, $u_1 = (d^T, u^T, a u^T)^T$, where $d = v_1^{(1)}$ and $u = v_2^{(1)}$. □

**Corollary 3.4.** Given D-KSVD that uses the same initialization scheme for $D^{(0)}$ 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.

Another advantage of D-KSVD is that only a single regularization parameter, $\gamma$, has to be determined by cross-validation as opposed to $\alpha$ and $\beta$ 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