# Dictionary Learning Using Rank-One Projection

Cheng Cheng and Wei Dai
*Dept. of Electrical and Electronic Engineering*, *Imperial College London*, London, United Kingdom
c.cheng17@imperial.ac.uk, wei.dai1@imperial.ac.uk

***Abstract*—Dictionary learning aims to find a dictionary that can sparsely represent the training data. Methods in the literature typically solve the problem by alternating between two stages: sparse coding and dictionary update. In this paper, we propose a novel dictionary learning algorithm using rank-one projection (ROP). The key contribution is that we cast dictionary learning as an optimization with respect to a single variable which is a set of rank one matrices. The resulting algorithm is hence single-stage. An alternating direction method of multipliers (ADMM) is derived to solve the optimization problem and a lower bound of penalty parameter is computed to guarantee a global convergence despite non-convexity of the optimization formulation. From practical point of view, ROP reduces the number of tuning parameters required in benchmark algorithms. Numerical tests demonstrate that ROP outperforms benchmark algorithms for both synthetic and real data.**

***Index Terms*—ADMM, dictionary learning, non-convex optimization, rank-one projection, single image super-resolution**

## I. Introduction

Sparse signal representation has drawn extensive attention due to its various applications in signal denoising [1], [2], restoration [3], [4], source separation [5], [6], recognition [7], [8], and image super-resolution [9], [10]. The basic idea of it is that an observed signal can be approximated as a linear combination of a few atoms picked from a dictionary. Compared with choosing a basis from predefined dictionaries such as Fourier and wavelet transforms, a dictionary trained from the data itself can attain sparser representations [11]. Therefore massive interests have been attracted to find a dictionary that can sparsely represent the training data.

A typical dictionary learning algorithm is an iterative process alternating between two stages: sparse coding and dictionary update [12]–[17]. As dictionary learning involves two unknown variables, the general principle is to fix one variable and optimizing the other. The purpose of sparse coding is to find the sparse coefficients based on a given dictionary. This optimization problem can be solved using two different strategies: greedy algorithms such as matching pursuit (MP) [18], orthogonal matching pursuit (OMP) [19], [20], subspace pursuit (SP) [21], CoSaMP [22] that select the support set from the sparse coefficients sequentially, and Basis Pursuit (BP) [23] that convexifies the problem by replacing $\ell_0$ pseudo-norm with $\ell_1$ norm. The other stage dictionary update aims to refine the atoms of the dictionary using the sparse coefficients obtained from the previous stage. Method of optimal directions (MOD) [12] is one of the earliest two-stage methods, where the whole sparse coefficient matrix is fixed and the stage is cast as a least squares problem. In many other methods including K-SVD [13], SimCO [16], and Blotless [17], only the sparsity pattern (the positions of non-zeros) of sparse coefficients is fixed, and both the dictionary and the sparse coefficients are updated. Specifically, K-SVD updates one column of the dictionary and the corresponding row of sparse coefficients, while fixing all other dictionary atoms and the corresponding sparse coefficients. SimCO updates the whole dictionary and the whole sparse coefficient matrix by viewing coefficients as a function of dictionary and performing a gradient descent with respect to dictionary. Blotless updates a block of the dictionary and the corresponding sparse coefficients using a total least squares approach.

In this paper, a novel dictionary learning algorithm that uses rank-one projection (ROP) is proposed. The key novelty in ROP is to formulate dictionary learning as an optimization problem involving only one unknown variable, i.e., a set of rank-one matrices. More specifically, dictionary learning is cast as representing training data as the sum of rank-one matrices, each with only a few non-zero columns. With this formulation, the two-stage optimization procedure in the literature is replaced by a single-stage process. Then alternating direction method of multipliers (ADMM) is adapted to solve the ROP formulation. Note that ROP involves a constrained optimization with non-smooth objective function and a non-convex constraint (the set of rank-one matrices is non-convex). Nevertheless, recent advance in optimization theory [24] shows that the ADMM solver of ROP enjoys a global convergence guarantee if the penalty parameter of augmented Lagrangian is carefully chosen.

The single variable formulation ROP brings significant benefits. Firstly, it reduces the burden of parameter tuning. In the sparse coding stage of benchmark algorithms, one typically needs by trial-and-error to choose either the maximum sparsity level for greedy algorithms or a regularization constant for a Lasso type of formulation. By comparison, there is no parameter to tune in ROP in generating all the simulations in this paper. Secondly, our numerical results demonstrate that ROP outperforms other benchmark algorithms for the tests involving both synthetic data and real data. The results show that ROP can train good dictionaries with much less training data compared with other benchmark algorithms. In the tests of real data, the performance improvement of ROP is demonstrated using examples of single image super-resolution.

## II. Background

Let $\boldsymbol{Y} \in \mathbb{R}^{M\times N}$, where $M \in \mathbb{N}$ and $N \in \mathbb{N}$ denote the dimension and the number of training vectors, respectively.

Dictionary learning can be written as

$$\min_{\boldsymbol{D},\boldsymbol{X}} \sum_n \|\boldsymbol{X}_{:,n}\|_0 \text{ s.t. } \boldsymbol{Y} \approx \boldsymbol{D}\boldsymbol{X}, \tag{1}$$

where $\boldsymbol{D} \in \mathbb{R}^{M\times K}$ denotes the unknown dictionary, and $\boldsymbol{X} \in \mathbb{R}^{K\times N}$ are the sparse representation coefficients, $\boldsymbol{X}_{:,n}$ is the $n$-th column of $\boldsymbol{X}$, and $\|\cdot\|_0$ is the $\ell_0$ pseudo-norm. The constraint $\boldsymbol{Y} \approx \boldsymbol{D}\boldsymbol{X}$ can be rewritten as $\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F \le \epsilon$ when the noise energy in the training data can be roughly estimated, where $\|\cdot\|_F$ denotes the Frobenius norm and $\epsilon > 0$ is a constant chosen based on the noise energy. In dictionary learning problems, it is typical that $M < K$, i.e., the dictionary is over-complete. To make dictionary learning feasible, in the literature extra constraints are imposed and suboptimal algorithms are designed [12]–[17]. Note the scaling ambiguity that $\boldsymbol{D}_{:,k}\boldsymbol{X}_{k,:} = (a\boldsymbol{D}_{:,k})(\frac{1}{a}\boldsymbol{X}_{k,:})$. It is common to assume unit $\ell_2$-norm of columns of $\boldsymbol{D}$.

A popular approach is assuming that the sparse coefficients of each training vector in $\boldsymbol{Y}$ has at most $S$ many non-zeros, where $S \ll M$ is a predefined constant carefully chosen by trial-and-error. The optimization problem (1) then becomes

$$\begin{aligned} &\min_{\boldsymbol{D},\boldsymbol{X}} \ \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \\ &\text{s.t. } \|\boldsymbol{D}_{:,k}\|_2 = 1,\ \|\boldsymbol{X}_{:,n}\|_0 \le S,\ \forall n \in [N], \forall k \in [K], \end{aligned} \tag{2}$$

where $[N] := 1, 2, \cdots, N$. The problem (2) is typically solved by alternating between two stages: sparse coding and dictionary update. In the sparse coding stage, one fixes the dictionary $\boldsymbol{D}$ and updates the coefficients $\boldsymbol{X}$ by

$$\min_{\boldsymbol{X}_{:,n}} \|\boldsymbol{Y}_{:,n} - \boldsymbol{D}\boldsymbol{X}_{:,n}\|_2^2, \text{ s.t. } \|\boldsymbol{X}_{:,n}\|_0 \le S,\ \forall n \in [N]. \tag{3}$$

The non-convex problem (3) can be solved by many pursuit algorithms [18]–[22]. In the dictionary update stage, one updates the dictionary by fixing either the sparse coefficients, for example MOD [12], or the sparsity pattern, for example K-SVD [13], SimCO [16], and Blotless [17].

These two-stage algorithms have the same issue. The performance of the two stages are coupled together and the optimal tuning of one stage may not lead to the optimal performance of the overall dictionary learning. Furthermore, few performance guarantees have been obtained in the literature for the general dictionary learning problem.

## III. Dictionary Learning Via ROP

This section derives the ROP formulation for dictionary learning which avoids alternating between two stages.

We start with the constraint set in the original dictionary learning problem (1). It is straightforward to see that $\boldsymbol{Y} \approx \boldsymbol{D}\boldsymbol{X} = \sum_k \boldsymbol{D}_{:,k}\boldsymbol{X}_{k,:} = \sum_k \boldsymbol{Z}_k$ where $\boldsymbol{Z}_k := \boldsymbol{D}_{:,k}\boldsymbol{X}_{k,:}$ is a rank-one matrix for all $k \in [K]$. Define the set of rank-one matrices of proper size $\mathcal{R}1 := \left\{\boldsymbol{Z} \in \mathbb{R}^{M\times N} : \text{rank}(\boldsymbol{Z}) = 1\right\}$. Then the constraint set in (1) can be written as $\boldsymbol{Y} \approx \sum_k \boldsymbol{Z}_k,\ \boldsymbol{Z}_k \in \mathcal{R}1,\ \forall k \in [K]$.

The objective function is adapted accordingly. It is clear that a zero entry in $\boldsymbol{X}$, say $X_{k,n}$, results in a zero column in $\boldsymbol{Z}_k$, i.e., $(\boldsymbol{Z}_k)_{:,n} = \boldsymbol{D}_{:,k}X_{k,n} = \boldsymbol{0}$. The objective function is designed to promote zero columns in $\boldsymbol{Z}_k$, that is, $\sum_k \|\boldsymbol{Z}_k\|_{2,0}$, where

$$\|\boldsymbol{Z}_k\|_{2,0} := \|[\|(\boldsymbol{Z}_k)_{:,1}\|_2, \|(\boldsymbol{Z}_k)_{:,2}\|_2 \cdots, \|(\boldsymbol{Z}_k)_{:,N}\|_2]^T\|_0$$

counts the number of non-zero columns of $\boldsymbol{Z}_k$. In practice, the non-convex $\ell_0$ pseudo-norm is replaced with convex $\ell_1$-norm, resulting in the convex objective function

$$\sum_k \|\boldsymbol{Z}_k\|_{2,1} := \sum_k \sum_n \|(\boldsymbol{Z}_k)_{:,n}\|_2. \tag{4}$$

Then dictionary learning is cast as

$$\min_{\boldsymbol{Z}_k} \sum_k \|\boldsymbol{Z}_k\|_{2,1} \text{ s.t.} \boldsymbol{Y} \approx \sum_k \boldsymbol{Z}_k,\ \boldsymbol{Z}_k \in \mathcal{R}1,\ \forall k \in [K]\,. \tag{5}$$

After solving (5), the dictionary atoms $\boldsymbol{D}_{:,k}$ and the corresponding coefficients $\boldsymbol{X}_{k,:}$ can be obtained using singular value decomposition (SVD) of $\boldsymbol{Z}_k$. In practice, the constraint $\boldsymbol{Y} \approx \sum_k \boldsymbol{Z}_k$ can be replaced with data fidelity target, e.g., $\|\boldsymbol{Y} - \sum_k \boldsymbol{Z}_k\|_F \le \epsilon$.

The solutions of (5) are invariant under the projection onto the set of rank-one matrices. Hence we term this formulation *Rank-One Projection (ROP)*. It is a non-convex optimization as the set $\mathcal{R}1$ is non-convex. There is a permutation ambiguity in the solution of ROP. It is noteworthy that all dictionary learning methods in the literature are non-convex and permutation ambiguous.

Now it becomes clear the key difference between ROP and benchmark algorithms. Benchmark algorithms involve two unknown variables, i.e., the set of dictionary atom $\boldsymbol{D}_{:,k}$ and the set of sparse coefficient vector $\boldsymbol{X}_{k,:}$, and optimize over them alternatively (in sparse coding and dictionary learning stages respectively) with different strategies. The analysis of alternating optimization can be very complicated: [16] shows that the optimization process may converge to singular points rather than the commonly thought local minimum points. Also hyper-parameters have to be introduced and well-tuned even when the data fidelity target is well set. By contrast, ROP involves only one unknown variable the set of rank-one matrix $\boldsymbol{Z}_k$. It avoids two-stage optimization, the existence of singular points, or extra hyper-parameters to tune except the data fidelity target which is necessary for all dictionary learning methods.

### A. An ADMM Solver for ROP

In this subsection, ADMM technique is adapted to solve (5). As we show later, although the problem (5) is non-convex, the ADMM procedure will converge.

For compositional convenience, we derive ADMM version of ROP by replacing the constraint $\boldsymbol{Y} \approx \sum_k \boldsymbol{Z}_k$ in (5) with $\boldsymbol{Y} = \sum_k \boldsymbol{Z}_k$. The extension to the constraint $\|\boldsymbol{Y} - \sum_k \boldsymbol{Z}_k\|_F \le \epsilon$ is similar and omitted here. Towards this end, we define the following indicator function for the set of rank-one matrices as

$$\mathbb{1}_{\mathcal{R}1}(\boldsymbol{Z}) = \begin{cases} 0, & \text{if } \boldsymbol{Z} \in \mathcal{R}1, \\ +\infty, & \text{otherwise.} \end{cases} \tag{6}$$

Further, we introduce auxiliary variables $\boldsymbol{P}_k \in \mathbb{R}^{M\times N}$ and $\boldsymbol{Q}_K \in \mathbb{R}^{M\times N}$. An equivalent form to (5) is given by

$$\begin{aligned}&\min_{\boldsymbol{P}_k,\boldsymbol{Q}_k,\boldsymbol{Z}_k} \sum_k \|\boldsymbol{P}_k\|_{2,1} + \sum_k \mathbb{1}_{\mathcal{R}1}(\boldsymbol{Q}_k)\\ &\text{s.t. } \boldsymbol{Y} = \sum_k \boldsymbol{Z}_k,\ \boldsymbol{P}_k = \boldsymbol{Z}_k,\ \boldsymbol{Q}_k = \boldsymbol{Z}_k,\ \forall k \in [K]. \end{aligned} \tag{7}$$

For readability, we use the notations in (7) instead of standard ADMM form, and we derive detailed ADMM iteration steps as follow. As there are $MN + 2MNK$ many equality constraints in (7), we denote the corresponding *scaled* Lagrange multipliers (see [25, §3.1.1] for details) by $\boldsymbol{\Lambda}_0 \in \mathbb{R}^{M\times N}$, $\boldsymbol{\Lambda}_{1,k} \in \mathbb{R}^{M\times N}$, and $\boldsymbol{\Lambda}_{2,k} \in \mathbb{R}^{M\times N}$, corresponding to the equality constraints $\boldsymbol{Y} = \sum_k \boldsymbol{Z}_k$, $\boldsymbol{P}_k = \boldsymbol{Z}_k$, and $\boldsymbol{Q}_k = \boldsymbol{Z}_k$, respectively. Then the augmented Lagrangian is given by

$$\begin{aligned}&\mathcal{L}_\rho\left(\{\boldsymbol{P}_k,\boldsymbol{Q}_k,\boldsymbol{Z}_k,\boldsymbol{\Lambda}_{1,k},\boldsymbol{\Lambda}_{2,k}\}_{k=1}^K,\boldsymbol{\Lambda}_0\right)\\ =&\sum_k\left(\|\boldsymbol{P}_k\|_{2,1} + \frac{\rho}{2}\|\boldsymbol{Z}_k - \boldsymbol{P}_k + \boldsymbol{\Lambda}_{1,k}\|_F^2 - \frac{\rho}{2}\|\boldsymbol{\Lambda}_{1,k}\|_F^2\right)\\ +&\sum_k\left(\mathbb{1}_{\mathcal{R}1}(\boldsymbol{Q}_k) + \frac{\rho}{2}\|\boldsymbol{Z}_k - \boldsymbol{Q}_k + \boldsymbol{\Lambda}_{2,k}\|_F^2 - \frac{\rho}{2}\|\boldsymbol{\Lambda}_{2,k}\|_F^2\right)\\ +&\frac{\rho}{2}\|\sum_k \boldsymbol{Z}_k - \boldsymbol{Y} + \boldsymbol{\Lambda}_0\|_F^2 - \frac{\rho}{2}\|\boldsymbol{\Lambda}_0\|_F^2 \end{aligned} \tag{8}$$

where $\rho > 0$ denotes the penalty parameter. Then the ADMM iterations are given by

$$\boldsymbol{P}_k^{l+1} = \arg\min_{\boldsymbol{P}_k} \|\boldsymbol{P}_k\|_{2,1} + \frac{\rho}{2}\|\boldsymbol{Z}_k^l - \boldsymbol{P}_k + \boldsymbol{\Lambda}_{1,k}^l\|_F^2, \tag{9}$$

$$\boldsymbol{Q}_k^{l+1} = \arg\min_{\boldsymbol{Q}_k} \mathbb{1}_{\mathcal{R}1}(\boldsymbol{Q}_k) + \frac{\rho}{2}\|\boldsymbol{Z}_k^l - \boldsymbol{Q}_k + \boldsymbol{\Lambda}_{2,k}^l\|_F^2, \tag{10}$$

$$\begin{aligned}&(\cdots,\boldsymbol{Z}_k^{l+1},\cdots) = \arg\min_{\cdots,\boldsymbol{Z}_k,\cdots} \|\sum_k \boldsymbol{Z}_k - \boldsymbol{Y} + \boldsymbol{\Lambda}_0^l\|_F^2 + \sum_k \|\\ &\boldsymbol{Z}_k - \boldsymbol{P}_k^{l+1} + \boldsymbol{\Lambda}_{1,k}^l\|_F^2 + \sum_k \|\boldsymbol{Z}_k - \boldsymbol{Q}_k^{l+1} + \boldsymbol{\Lambda}_{2,k}^l\|_F^2, \end{aligned} \tag{11}$$

$$\boldsymbol{\Lambda}_0^{l+1} = \boldsymbol{\Lambda}_0^l + \sum_k \boldsymbol{P}_k^{l+1} - \boldsymbol{Y}, \tag{12}$$

$$\boldsymbol{\Lambda}_{1,k}^{l+1} = \boldsymbol{\Lambda}_{1,k}^l + \boldsymbol{Z}_k^{l+1} - \boldsymbol{P}_k^{l+1}, \tag{13}$$

$$\boldsymbol{\Lambda}_{2,k}^{l+1} = \boldsymbol{\Lambda}_{2,k}^l + \boldsymbol{Z}_k^{l+1} - \boldsymbol{Q}_k^{l+1}, \tag{14}$$

where $l$ denotes the iteration number.

Each iteration of ADMM involves three optimization problems (9-11) that are conceptually easy to solve. The optimization problem (9) is convex but involves a non-differential term $\|\cdot\|_{2,1}$ in its objective function. The closed form of the optimal solution of (9) can be obtained by setting the sub-gradient of the objective function to zero. Define $\hat{\boldsymbol{P}}_k := \boldsymbol{Z}_k^l + \boldsymbol{\Lambda}_{1,k}^l$. Then

$$(\boldsymbol{P}_k^{l+1})_{:,n} = \left(1 - \frac{1}{\rho\|(\hat{\boldsymbol{P}}_k)_{:,n}\|_2}\right)_+ (\hat{\boldsymbol{P}}_k)_{:,n}, \tag{15}$$

where $(x)_+ := \max(0,x)$.

The optimization problem (10) is non-convex. Fortunately by Eckart-Young-Mirsky theorem, it can be solved by using singular value decomposition (SVD). Define $\hat{\boldsymbol{Q}}_k = \boldsymbol{Z}_k^l + \boldsymbol{\Lambda}_{2,k}^l$. Considering the SVD of the matrix $\hat{\boldsymbol{Q}}_k$, denote its largest singular value by $\sigma_1$ and the corresponding right and left singular vectors by $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ respectively. Then $\boldsymbol{Q}_k^{l+1} = \sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^T$.

The optimization problem (11) is a quadratic programming. In principle, it can be solved by many commercially available optimization toolkits. However, this quadratic programming involves a linear map of huge dimensions, which results in large run-time when using standard solvers. To address this problem, we design and implement a conjugate gradient (CG) procedure which uses the structures in (11) to simplify the computations substantially. In our simulation part, our CG procedure cuts the run-time in orders of magnitude. The details are omitted here due to the length constraint of this paper.

### *B. Convergence of ROP-ADMM*

ROP involves a non-convex ADMM with a non-smooth objective function. It is important to ensure its convergence before using it in practice.

We extend results in [24] and show that ROP enjoys global convergence guarantee when the penalty parameter $\rho > 4$. To proceed, the following definition is needed.

**Definition 1.** (Restricted prox-regularity) [24, Definition 2] For a lower semi-continuous function $f$, let $J \in \mathbb{R}_+$, $f : \mathbb{R}^N \to \mathbb{R}\cup\{\infty\}$, and define the exclusion set

$$S_J := \{x \in \text{domain}(f) : \|d\| > J \text{ for all } d \in \partial f(x)\} \tag{16}$$

$f$ is called *restricted prox-regular* if, for any $J > 0$ and bounded set $T \subseteq \text{domain}\, f$, there exists $\gamma > 0$ such that

$$\begin{aligned}&f(y) + \frac{\gamma}{2}\|x-y\|^2 \geq f(x) + \langle d, y-x\rangle,\\ &\forall x \in T\backslash S_J, y \in T, d \in \partial f(x), \|d\| \leq J. \end{aligned} \tag{17}$$

Consider the ADMM formulation of ROP in (7). It can be verified that the first term in the objective function $\sum_k \|\boldsymbol{P}_k\|_{2,1}$ is restricted prox-regular. The second term in the objective function is an indicator function of rank-one matrices, which is lower semi-continuous. The reason is that the indicator function of a closed set is lower semi-continuous, and the set of rank-one matrices is closed. Then we prove that when $\rho > 4$, the augmented Lagrangian (8) descends during each updating step. The ADMM process starting from any point $\left(\{\boldsymbol{P}_k^0,\boldsymbol{Q}_k^0,\boldsymbol{Z}_k^0,\boldsymbol{\Lambda}_{1,k}^0,\boldsymbol{\Lambda}_{2,k}^0\}_{k=1}^K,\boldsymbol{\Lambda}_0^0\right)$ converges to a stationary point, where $\boldsymbol{0} \in \partial\mathcal{L}_\rho\left(\{\boldsymbol{P}_k^*,\boldsymbol{Q}_k^*,\boldsymbol{Z}_k^*,\boldsymbol{\Lambda}_{1,k}^*,\boldsymbol{\Lambda}_{2,k}^*\}_{k=1}^K,\boldsymbol{\Lambda}_0^*\right)$. The detailed analysis is omitted here due to the space constraint.

## IV. Numerical Tests

This section compares the numerical performance of ROP with other benchmark dictionary learning algorithms including MOD, K-SVD, and BLOTLESS. The comparison in Section IV-A is based on synthetic data while real data is involved in Section IV-B.

### A. Dictionary learning for synthetic data

For synthetic data tests, we adopt the typical setting for data generation. We assume that the training data $\boldsymbol{Y}$ are generated from a ground-truth dictionary $\boldsymbol{D}^0$ and a ground-truth sparse coefficient matrix $\boldsymbol{X}^0$ via $\boldsymbol{Y} = \boldsymbol{D}^0\boldsymbol{X}^0$. The dictionary $\boldsymbol{D}^0$ is generated by first filling it with independent realizations of the standard Gaussian variable and then normalizing its columns to have unit $\ell_2$-norm. The sparse coefficients in $\boldsymbol{X}^0$ is generated as follows. Assume that the number of nonzero coefficients in the $n$-th column of $\boldsymbol{X}^0$ is $S_n$. The index set of the nonzero coefficients are randomly generated from the uniform distribution on $\binom{[K]}{S_n}$ and the values of the nonzero coefficients are independently generated from the standard Gaussian distribution. In our simulations, we set $S_n = S \in \mathbb{N}$, $\forall n \in [N]$.

Given the synthetic data, different dictionary learning algorithms are tested. OMP [19] is used for the sparse coding stage of MOD, K-SVD, and BLOTLESS, with the prior knowledge of $S$. Note that different from other benchmark algorithms, ROP does not require such prior information.

Dictionary learning algorithms are compared using dictionary recovery error. Considering the permutation ambiguity of the trained dictionary, we define the dictionary recovery error as

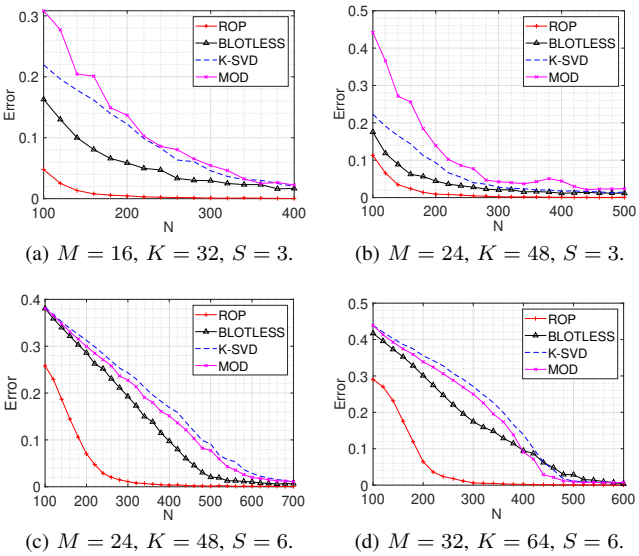
$$\text{Error} := \frac{1}{K}\sum_{k=1}^{K}(1 - \mid \hat{\boldsymbol{D}}_{:,k}^T \boldsymbol{D}_{:,i_k}^0 \mid), \tag{18}$$

where $i_k := \arg\max_{i\in\mathcal{I}_k}(\hat{\boldsymbol{D}}_{:,k}^T\boldsymbol{D}_{:,i}^0)$, $\mathcal{I}_k := [K]\backslash\{i_1,\cdots,i_{k-1}\}$, $\hat{\boldsymbol{D}}_{:,k}$ denotes the $k$-th column of estimated dictionary, and $\boldsymbol{D}_{:,i_k}^0$ represents the $i_k$-th column of ground truth dictionary which has largest correlation with $\hat{\boldsymbol{D}}_{:,k}$. The use of $\mathcal{I}_k$ is to avoid repetitions in $i_k$, $\forall k \in [K]$.

Fig. 1 compares the performance of dictionary learning algorithms. The results are averages of 100 random trials, and in each trial the maximum number of iterations is set to 500. The results in Fig. 1 clearly show that ROP outperforms all other tested benchmark algorithms. The number of training samples required for ROP for a good recovery is the least. More importantly, when the number of training samples is relatively large, ROP is the only algorithm that has no visible error floor while all other algorithms suffer from non-negligible error floors.

### B. Single image super-resolution using dictionary learning

This subsection focuses on the performance comparison of dictionary learning algorithms when applied for single image super-resolution problem. We follow the approach by Yang et al. in [9]. The basic idea is that given pairs of low- and high-resolution images as training data, a pair of dictionaries are learned so that sparse approximations of each pair of low/high-resolution images share the same coefficients. For a test image of low-resolution, one first finds its sparse representation under the low-resolution dictionary, and then apply the corresponding sparse coefficients to the high-resolution dictionary to generate a high-resolution image.




(a) $M = 16$, $K = 32$, $S = 3$. (b) $M = 24$, $K = 48$, $S = 3$.

(c) $M = 24$, $K = 48$, $S = 6$. (d) $M = 32$, $K = 64$, $S = 6$.

Fig. 1: Comparison of dictionary learning methods for the noise-free cases. Results are averages of 100 trials.

Our simulations are based on MNIST dataset which contains images for digits from 0 to 9. Each image is of $28 \times 28$ pixels. We generate low-resolution images of size $14 \times 14$ by grouping adjacent $2 \times 2$ pixels from original images and taking their average as one pixel.

The training data used for dictionary learning is patch based. Patches of size $3 \times 3$ are extracted from the low-resolution images with 2 pixel overlap in either direction for adjacent patches. Find the corresponding patches of size $6 \times 6$ from the high-resolution images. Stack each pair of low- and high-resolution patches to form a column in the training data, i.e., $\boldsymbol{Y}_{:,n} = \left[\text{vect}(\boldsymbol{P}_L)_n^T, \text{vect}(\boldsymbol{P}_H)_n^T\right]^T$, where $\boldsymbol{P}_L$ and $\boldsymbol{P}_H$ are low/high-resolution patches respectively. In the simulations, we use 144 patches and hence the training sample matrix $\boldsymbol{Y}$ is of size $45 \times 144$. Usually, it is suggested to use some techniques to extract the features from the patches before training the dictionaries, such as first- and second-order derivatives and principal component analysis (PCA). While, here we use the pure patches instead of features, as such preprocessing techniques ease the difficulty of training dictionary, which may influence the comparison of dictionary learning algorithms.

We then apply different algorithms for dictionary learning. Denote the acquired dictionary by $\boldsymbol{D} = \left[\boldsymbol{D}_L^T, \boldsymbol{D}_H^T\right]^T$, where $\boldsymbol{D}_L$ and $\boldsymbol{D}_H$ are the sub-dictionaries corresponding to low- and high-resolution patches respectively. Here we set $K = 128$. Given a low-resolution image for the test, extract $3 \times 3$ patches with overlap of 2 pixels between adjacent patches in either direction. For each patch, a sparse representation coefficient vector $\boldsymbol{\alpha}$ is obtained so that $\boldsymbol{P}_L \approx \boldsymbol{D}_L\boldsymbol{\alpha}$ using sparse coding technique for example OMP. The corresponding high-resolution patches are generated via $\boldsymbol{D}_H\boldsymbol{\alpha}$ and the high-resolution image is generated by aligning the patches and taking average of overlapped pixels across patches.

TABLE I: Comparison of single image super-resolution via dictionary learning.

| High-resolution ground truth | Low-resolution samples | Methods | | | |
|---|---|---|---|---|---|
| | | ROP | BLOTLESS | K-SVD | MOD |
|  |  |  |  |  |  |
| PSNR of digit 5 | 19.1722 | 22.6663 | 13.8718 | 14.1980 | 12.5625 |
| PSNR of digit 0 | 19.2372 | 28.1114 | 14.1279 | 12.2549 | 12.1283 |
| PSNR of digit 9 | 18.9517 | 23.3359 | 14.3899 | 13.8016 | 13.3339 |
| PSNR of digit 2 | 19.2001 | 24.6123 | 13.6228 | 13.3721 | 12.7610 |

The simulation results are presented in Table I. In numerical comparison, peak signal-to-noise ratio (PSNR) is used as the performance criterion, which is formulated as

$$\mathrm{PSNR} = 10\log_{10}\frac{N_e}{\|\hat{\boldsymbol{I}} - \boldsymbol{I}^0\|_F^2}, \tag{19}$$

where $\boldsymbol{I}^0$ and $\hat{\boldsymbol{I}}$ are the 'ground-truth' high-resolution image and a high-resolution image generated using the learned dictionary respectively, and $N_e$ denotes the number of entries in $\boldsymbol{I}^0$. Simulation results demonstrate the significant improvement of ROP in both the numerical error and the visual effect. Without preprocessing techniques, only ROP boosts the resolution of testing images.

## V. Conclusion

In this paper, we propose a novel dictionary learning algorithm using rank-one projection (ROP), where the problem is cast as an optimization with respect to a single variable. Practically ROP reduces the number of tuning parameters required in other benchmark algorithms. An ADMM is derived to solve the optimization problem and guarantees a global convergence. The test results show that ROP outperforms other benchmark algorithms for both synthetic data and real data especially when the number of sample is small.

## References


[1] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[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] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2013.

[5] Y. Li, S.-I. Amari, A. Cichocki, D. W. Ho, and S. Xie, "Underdetermined blind source separation based on sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 423–437, 2006.

[6] V. Abolghasemi, S. Ferdowsi, and S. Sanei, "Blind separation of image sources via adaptive dictionary learning," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 2921–2930, 2012.

[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] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition," in *Computer vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 471–478.

[9] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[10] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.

[11] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, p. 607, 1996.

[12] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 5. IEEE, 1999, pp. 2443–2446.

[13] M. Aharon, M. Elad, A. Bruckstein *et al.*, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, p. 4311, 2006.

[14] K. Engan, K. Skretting, and J. H. Husøy, "Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation," *Digital Signal Processing*, vol. 17, no. 1, pp. 32–49, 2007.

[15] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.

[16] W. Dai, T. Xu, and W. Wang, "Simultaneous codeword optimization (SimCO) for dictionary update and learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6340–6353, 2012.

[17] Q. Yu, W. Dai, Z. Cvetkovic, and J. Zhu, "Bilinear dictionary update via linear least squares," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7923–7927.

[18] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," Courant Institute of Mathematical Sciences New York United States, Tech. Rep., 1993.

[19] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. IEEE, 1993, pp. 40–44.

[20] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[21] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[22] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[23] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[24] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers, Inc., 2011, vol. 3, no. 1.