OUTLIER-AWARE DICTIONARY LEARNING FOR SPARSE REPRESENTATION

Sajjad Amini, Mostafa Sadeghi*, Mohsen Joneidi*, Massoud Babaie-Zadeh*, Christian Jutten***

*Electrical Engineering Department, Sharif University of Technology, Tehran, IRAN.
**GIPSA-Lab, Grenoble, and Institut Universitaire de France, France.

ABSTRACT

Dictionary learning (DL) for sparse representation has been widely investigated during the last decade. A DL algorithm uses a training data set to learn a set of basis functions over which all training signals can be sparsely represented. In practice, training signals may contain a few outlier data, whose structures differ from those of the clean training set. The presence of these unpleasant data may heavily affect the learning performance of a DL algorithm. In this paper we propose a robust-to-outlier formulation of the DL problem. We then present an algorithm for solving the resulting problem. Experimental results on both synthetic data and image denoising demonstrate the promising robustness of our proposed problem.

Index Terms— Sparse representation, dictionary learning, robustness, outlier data.

1. INTRODUCTION

Sparse representation modelling has received a lot of attention during the last decade [1]. In a sparse representation problem, given a collection of basis vectors, the goal is to decompose natural signals and images as linear combinations of only a few basis vectors. The efficiency of this approach has been extensively investigated in many applications, e.g., image denoising [2], classification tasks [3], and so on.

An important problem in a sparse representation-based application is choosing the appropriate set of vectors over which each data can be sparsely represented. After [4], each vector is called an *atom*, and their collection is called a *dictionary*. One way to construct atoms of the dictionary is to use pre-defined dictionaries such as Fourier, Gabor, Discrete Cosine Transform (DCT), and wavelets. Another method, which is the focus of this paper, is to *learn* a sparsifying dictionary from a set of training signals. This is known as the dictionary learning problem [5], which has been shown to create dictionaries that are more efficient compared to pre-defined ones [1, 2, 5].

This work was partially funded by European project 2012-ERC-AdG-320684 CHESS, and Iran National Science Foundation (INSF) under Contract 91004600.

Consider a training data matrix $\mathbf{Y} \in \mathbb{R}^{n\times L} = [\mathbf{y}_1, \ldots, \mathbf{y}_L]$. A general dictionary learning problem is to factorize $\mathbf{Y}$ as $\mathbf{Y} \simeq \mathbf{DX}$ with $\mathbf{D} \in \mathbb{R}^{n\times K}$ and $\mathbf{X}$ a sparse-column matrix, where $K > n$ and $L \gg K$. This is generally performed by solving the following problem

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{X}\in\mathcal{X}} \sum_{i=1}^{L} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2 = \|\mathbf{Y} - \mathbf{DX}\|_F^2, \qquad (1)$$

where $\|.\|_F$ is the Frobenius norm, and $\mathcal{D}$ and $\mathcal{X}$ are admissible sets of the dictionary and the coefficient matrix, respectively. $\mathcal{D}$ is usually chosen as the set of dictionaries with unit column-norms. $\mathcal{X}$ is the set of matrices, $\mathbf{X}$, with sparse columns. A general approach to solve (1) is to use alternating minimization. This is shown in Algorithm 1.

Algorithm 1 A general approach for dictionary learning

- **Task:** Dictionary learning for training data $\mathbf{Y}$
- **Initialization:** Set $t = 0$, $\mathbf{D}^{(0)}$, $\mathbf{X}^{(0)}$
- **Repeat:**
  1. $\mathbf{X}^{(t+1)} = \text{argmin}_{\mathbf{X}\in\mathcal{X}} \ \|\mathbf{Y} - \mathbf{D}^{(t)}\mathbf{X}\|_F^2$
  2. $\mathbf{D}^{(t+1)} = \text{argmin}_{\mathbf{D}\in\mathcal{D}} \ \|\mathbf{Y} - \mathbf{D}\mathbf{X}^{(t+1)}\|_F^2$
  3. $t \leftarrow t + 1$
- **Until:** stopping criterion is met.

Line (1) in Algorithm 1 is indeed a sparse coding problem for which many algorithms have been proposed [6]. The main difference between various dictionary learning algorithms is their approach for updating the dictionary (line (2)). Two well-known algorithms are Method of Optimal Directions (MOD) [7] and K-Singular Value Decomposition (K-SVD) [8]. MOD finds the unconstrained minimum of the dictionary update problem, which results in a closed-form solution, and then projects it onto $\mathcal{D}$ by normalizing its columns. In this way, MOD updates all atoms at once. K-SVD on the other hand, uses a sequential approach to update the atoms one-by-one.

978-1-4799-3694-6/14/$31.00

The basic assumption of most DL algorithms is that each training signal can be modelled as a sparse linear combination of the atoms of a dictionary plus a Gaussian residual vector, which may correspond to additive Gaussian noise. In real situations, however, this model does not provide a good description of the data. For example, when entries of the training signals are sparsely corrupted, the Gaussian noise is not a good assumption and instead, a Laplace model for the additive noise leads to better performance in many practical applications (see [9, 10]). Such a model has been previously proposed in some works; for example, Lu *et al.* [9] used the ℓ_1 norm[1] function instead of the ℓ_2 norm in (1). Their proposed algorithm, however, is not robust to additive Gaussian noise. To take into account both additive Gaussian noise and sparse corruptions, Chen and Wu [10] proposed a robust DL problem by decomposing the residual vector into two components: a non-sparse component for Gaussian noises and a sparse component for large outliers.

In this paper, we consider a different situation in which a few training signals (named as outliers) present among the training signals that completely violate the basic model of DL algorithms (sparse linear combinations of the atoms plus additive Gaussian noise). Up to our best knowledge, no previous work has studied such a situation. Here, we do this by considering a robust data model which takes into account the outlier data. We then derive a DL algorithm suited to this model. Finally we evaluate the effectiveness of the model and the robust-to-outliers DL algorithm by performing some simulations on synthetic and real data.

The rest of the paper is organized as follows. In Section 2 we introduce our main idea. Our final proposed algorithm is discussed in Section 3. Then, Section 4 presents the simulation results.

2. THE MAIN IDEA

To take into account outlier data, we consider the following data model

$$\mathbf{y}_i = \mathbf{D}\mathbf{x}_i + \mathbf{n}_i + \mathbf{o}_i, \tag{2}$$

where $\mathbf{n}_i$ is a zero-mean Gaussian vector, and $\mathbf{o}_i$ is an outlier vector: $\mathbf{o}_i \neq \mathbf{0}$ if the ith training signal is outlier and $\mathbf{o}_i = \mathbf{0}$ otherwise. It should be mentioned that a similar data model has been proposed and used by Mateos and Giannakis [11] towards robustifying the principal component analysis (PCA) method. Here, however, we target the DL problem, which is generally different from PCA and needs different algorithms to solve.

To derive a robust DL formulation, we use this reasonable assumption that the number of outlier data is much smaller than the total number of training signals. Using this, we end up with the following outlier-aware DL problem

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{X}\in\mathcal{X},\mathbf{O}} \|\mathbf{Y}-\mathbf{D}\mathbf{X}-\mathbf{O}\|_F^2 + \lambda\|\mathbf{O}\|_{2,0} \tag{3}$$

where $\mathbf{O} = [\mathbf{o}_1 \dots \mathbf{o}_L]$ is the matrix of outliers, $\|\mathbf{O}\|_{2,0}$ is the mixed $\ell_{2,0}$ (pseudo) norm of $\mathbf{O}$ which is defined as the ℓ_0 (pseudo) norm of the vector $\mathbf{o} = [\|\mathbf{o}_1\|_2 \dots \|\mathbf{o}_L\|_2]^T$, in which the ℓ_0 norm of a vector simply counts the number of its non-zero entries. Finally, λ is a hyper-parameter whose value will be discussed in Subsection 3.2.

The ℓ_0 norm, however, is discontinuous and hence non-differentiable. This makes the problem (3) difficult to solve. A very popular alternative is the ℓ_1 norm which is the best convex approximation to the ℓ_0 norm [12].

Using this alternative, we reach to the mixed $\ell_{2,1}$ norm, which is defined as the ℓ_1 norm of $\mathbf{o}$. Our DL problem then becomes as follows

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{X}\in\mathcal{X},\mathbf{O}} \|\mathbf{Y}-\mathbf{D}\mathbf{X}-\mathbf{O}\|_F^2 + \lambda\|\mathbf{O}\|_{2,1} \tag{4}$$

Chen and Wu [10] has reached to a similar problem in which $\|\mathbf{O}\|_{2,1}$ is replaced with $\|\mathbf{O}\|_{1,1}$, which is defined as the sum of absolute values of the entries of $\mathbf{O}$. However, as said in Section 1, they aimed to robustify the DL problem relative to additive Gaussian noise and sparse corruptions.

In sparse decomposition literature, a problem of the usage of ℓ_1 norm is that it is not differentiable. However, in the above formulation we do not have such a problem because the entries of $\mathbf{o}$ are non-negative and, hence, its ℓ_1 norm is simply equal to the sum of its entries.

Note also that the previous two problems can be derived from a Maximum A Posteriori (MAP) estimation approach. To do so, assuming that the dictionary is deterministic and known, consider for each $\mathbf{x}_i$ a Laplace prior and for each $\mathbf{o}_i$ a distribution proportional to $\exp(-\alpha\|\mathbf{o}_i\|_2^0)$ (in the case of problem (3)) or $\exp(-\beta\|\mathbf{o}_i\|_2^1)$ (in the case of problem (4)). Then, assuming independence of $\mathbf{x}_i$, $\mathbf{n}_i$, and $\mathbf{o}_i$, to derive the MAP estimates of $\mathbf{x}$ and $\mathbf{o}$, the following problem has to be solved

$$\max_{\mathbf{x},\mathbf{o}} p(\mathbf{x},\mathbf{o}|\mathbf{y}) = \max_{\mathbf{x},\mathbf{o}} \{p(\mathbf{y}|\mathbf{x},\mathbf{o})\cdot p(\mathbf{x})\cdot p(\mathbf{o})\}. \tag{5}$$

where $p(\mathbf{y}|\mathbf{x},\mathbf{o}) \sim \mathcal{N}(\mathbf{D}\mathbf{x}+\mathbf{o},\sigma^2\mathbf{I})$. Simple calculations results in (3) or (4), where the sparsity constraints on $\mathbf{X}$ has been included in $\mathcal{X}$.

3. THE FINAL ALGORITHM

To solve the resulting optimization problem (4), we use alternating minimization. In other words, we iteratively minimize the objective function over just one variable while the other two are kept fixed. We first initialize $\mathbf{D}$ and $\mathbf{O}$, and then begin the iterative optimization by performing minimization over $\mathbf{X}$. Conventional methods for initializing $\mathbf{D}$, include using a

[1] The ℓ_p norm of a vector $\mathbf{x}$ for $p \geq 1$ is defined as $\|\mathbf{x}\|_p \triangleq (\sum_i |x_i|^p)^{\frac{1}{p}}$.

predefined dictionary like DCT, or constructing the atoms of the dictionary by randomly choosing from the training signals themselves [8]. The latter is not a good option in our problem, since some of the outliers may be chosen as the atoms, which in the sequel iterations may affect the overall performance. In order to initialize $\mathbf{O}$, we simply use the zero matrix. In other words, at the first iteration all training signals are treated as not being outliers.

Having initialized $\mathbf{D}$ and $\mathbf{O}$, we iterate between the following three steps:

1. *Sparse coding*:

$$\forall i: \ \mathbf{x}_i \leftarrow \underset{\mathbf{x}_i: \text{ sparse}}{\operatorname{argmin}} \ \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i - \mathbf{o}_i\|_2^2$$

2. *Outlier update*:

$$\forall i: \ \mathbf{o}_i \leftarrow \underset{\mathbf{o}_i}{\operatorname{argmin}} \ \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i - \mathbf{o}_i\|_2^2 + \lambda\|\mathbf{o}_i\|_2$$

3. *Dictionary update*:

$$\mathbf{D} \leftarrow \underset{\mathbf{D}\in\mathcal{D}}{\operatorname{argmin}} \sum_{i=1}^{L} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i - \mathbf{o}_i\|_2^2$$

Remark. A by-product of our proposed DL problem is a robust sparse coding algorithm. In other words, with a fixed dictionary, iterating between stage 1 and 2 of the above procedure corresponds to solving the following outlier-aware sparse coding problem

$$\min_{\mathbf{x},\mathbf{o}} \ \|\mathbf{y} - \mathbf{D}\mathbf{x} - \mathbf{o}\|_2^2 + \lambda\|\mathbf{o}\|_2 \tag{6}$$

So, from this point of view, our outlier-aware DL algorithm consists in iterating between two stages: outlier-aware sparse coding and dictionary update. This two-stage approach has another justification that in some applications e.g. image denoising where in one phase a dictionary is first learned and in the other phase signals are approximated over the learned dictionary, the second phase is actually performed via the above outlier-aware sparse coding. For more details, see Section 4.

In what follows, we discuss these steps in more details.

3.1. Sparse coding

In the *sparse coding* stage, we actually find the sparsest representations of the modified data vectors $\forall i: \ \mathbf{z}_i = \mathbf{y}_i - \mathbf{o}_i$ in the current dictionary. This is a general sparse representation problem and, indeed, any sparse coding algorithm can be used to perform this stage. For example, one may use Orthogonal Matching Pursuit (OMP) algorithm, as we chose in our simulations.

3.2. Outlier update

To update each outlier vector, the following problem has to be solved

$$\min_{\mathbf{o}} \ \|\mathbf{r} - \mathbf{o}\|_2^2 + \lambda\|\mathbf{o}\|_2 \tag{7}$$

where $\mathbf{r} = \mathbf{y} - \mathbf{D}\mathbf{x}$ is the residual vector. Setting the derivative of the objective function equal to zero at the optimal point $\mathbf{o}^*$ results in

$$\mathbf{o}^*(1 + \frac{\lambda}{2\ \|\mathbf{o}^*\|_2}) = \mathbf{r}. \tag{8}$$

So, $\mathbf{o}^*$ is a positive scale of $\mathbf{r}$; that is, $\mathbf{o}^* = \alpha\mathbf{r}$ where $\alpha > 0$. Substituting this into the above equation results in

$$\mathbf{o}^* = \begin{cases} (1 - \frac{\lambda}{2\|\mathbf{r}\|_2})\mathbf{r}, & \text{if } \|\mathbf{r}\|_2 > \frac{\lambda}{2} \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{9}$$

The above expression says that if the residual (the sparse representation error) norm of a training data is above a threshold λ, that point is recognized as an outlier; otherwise, it is recognized as a relevant data vector. From this, we also deduce the role of the trade-off parameter λ; the smaller values for it, the more points being declared as outliers.

3.3. Dictionary update

In the *dictionary update* stage, the following problem has to be solved

$$\min_{\mathbf{D}\in\mathcal{D}} \|\mathbf{A} - \mathbf{D}\mathbf{X}\|_F^2, \tag{10}$$

where $\mathbf{A} = \mathbf{Y} - \mathbf{O}$. Any dictionary update algorithm (see e.g. [5]) can be used to solve the above problem, but we choose to solve it by MOD due to its simplicity. Doing so, we obtain [7]:

$$\mathbf{D} \leftarrow \mathbf{A}\mathbf{X}^\dagger, \tag{11}$$

where $\mathbf{X}^\dagger$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{X}$ defined as $\mathbf{X}^\dagger = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$.

4. EXPERIMENTAL RESULTS

In this section, we compare the performance of our outlier-aware dictionary learning (OADL) algorithm with an ordinary dictionary learning algorithm (here we use MOD) when the training set contains a small number of outliers. At first, we present the comparison of the two algorithms on synthetic data and then we move to a more realistic experiment.

4.1. Synthetic data

In the synthetic data test, we first generated a random 25×50 dictionary whose entries come from an *i.i.d.* $\mathcal{N}(0,1)$ distribution and then scaled its columns to have unit ℓ_2 norm. We randomly selected 3 atoms of dictionary and linearly combined them using 3 *i.i.d.* $\mathcal{N}(0,1)$ coefficients to generate a sample and using this procedure, we formed two sets of training

and test samples of sizes 2500 and 500, respectively. Training samples were then scaled to lave unit ℓ_2 norm and a random vector from $\mathcal{N}(\mathbf{0}, 0.01^2\mathbf{I})$ was added to each of training samples. To generate outliers, we added random vectors from $\mathcal{N}(\mathbf{0}, 0.04^2\mathbf{I})$ to some training samples and considered those samples to be outliers.

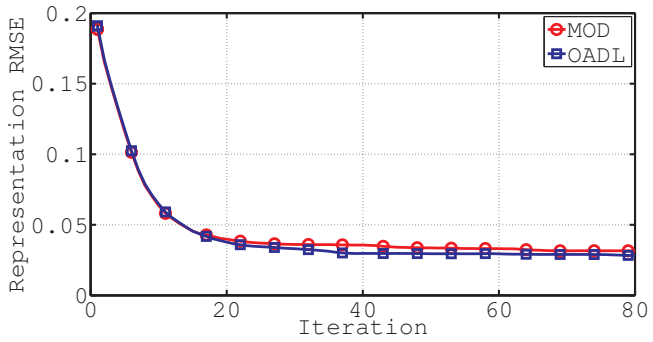

Fig. 1: Representation RMSE versus iteration. About 2% of training samples are outliers.

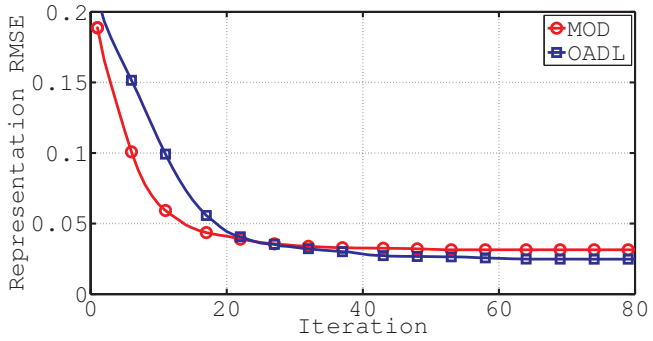

Fig. 2: Representation RMSE versus iteration. About 6% of training samples are outliers.

The training signals were then fed into MOD and OADL to estimate the underlying dictionary (DCT was chosen as the initial dictionary in both algorithms). To compare the resultant dictionaries, we examined their ability to approximate test samples using 3 atoms in the sense of representation root mean square error (RMSE), defined as $\|\mathbf{Y} - \mathbf{DX}\|_F/(n \cdot L)$. We conducted each experiment 20 times and the averaged results are reported.

Figure 1 shows the representation RMSE of test samples along MOD and OADL when 2% of training samples were outliers. In this case, the two algorithms perform nearly the same in both steady state values and convergence speed, but as the number of outliers with respect to training samples increases, OADL shows its superiority. Figure 2 shows the representation RMSE when 6% of the training samples were outliers. As can be seen, OADL outperform MOD significantly in terms of representation RMSE while it converges a little

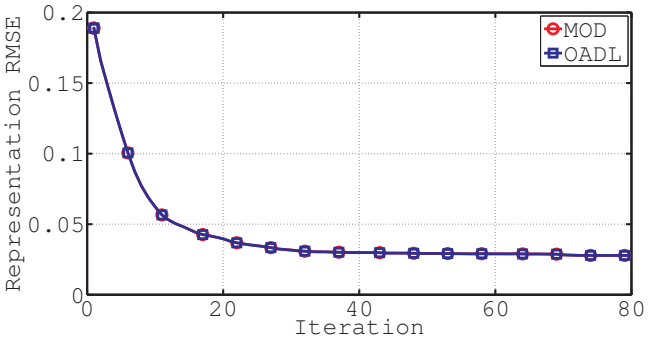

Fig. 3: Representation RMSE versus iteration. About 12% of training samples are outliers.

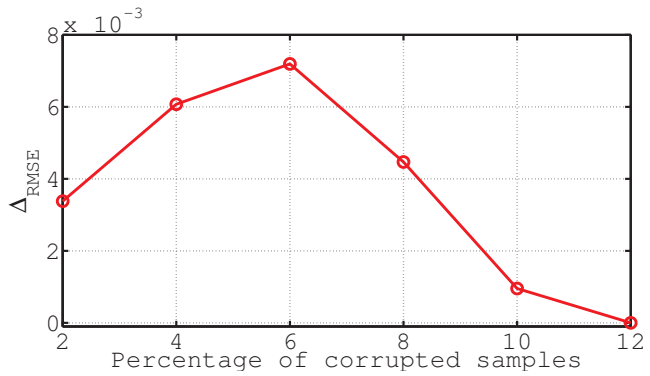

Fig. 4: Difference in steady state values of the representation RMSE between MOD and OADL versus percentage of outliers.

slower than MOD. According to our formulation, as the number of outliers increases, we go to a direction that violates our assumption and as a result we expect a reduction in performance of OADL. Figure 3 shows the representation RMSE when 12% of the training samples were outliers and as we see, OADL does not outperform MOD. This is because in this case, the number of outlier blocks is not small enough compared to to total number of training signals. Figure 4 shows the difference in the steady state representation RMSE between MOD and OADL when the percentage of outliers is varied. As we see, OADL is superior to MOD when the percentage of outliers is less than 12%.

Another important point that should be mentioned is the procedure of choosing λ. As λ increases, we go toward MOD and as it decreases, we go toward a situation that more samples are considered as outliers, so large λ is proper for small percentage of outliers and small λ is suited for large percentage of outliers. For each of outliers percentage, we perform the experiment for different λ between 0.05 and 1 with stepsize of 0.05. Figure 5 shows the best λ (λ_{opt}) that is the lambda that minimizes RMSE of the representation of the test data, for different values of outlier percentage. It is seen, be-

fore $p = 6$ the increase in outlier percentage leads to decrease in λ as we expected, but after $p = 6$, increasing p also increases λ which shows that OADL is going toward MOD. In this situation, MOD is becoming better than our proposed method.

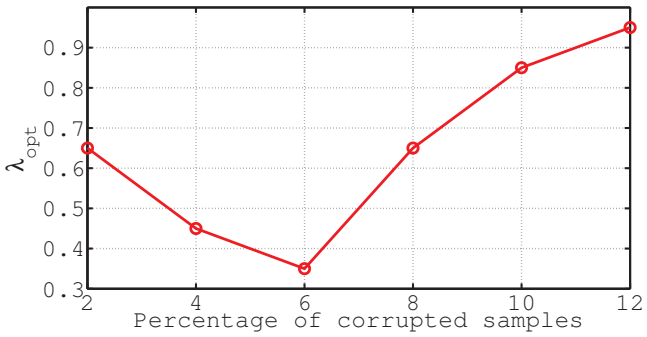

Fig. 5: Best λ, which minimizes test data representation RMSE, versus percentage of outliers.

4.2. Image Denoising

Our second experiment is a denoising scheme using overcomplete dictionary learning [2]. Consider an image contaminated by a low variance *i.i.d.* normal noise and a high variance *i.i.d.* normal noise in small blocks in different parts of the image. A general denoising scheme based on dictionary learning is as follows. First, an overcomplete dictionary is trained using (some of) the noisy image patches. Then, the whole noisy image patches are denoised over the learned dictionary. Some of the training patches may come from noisier parts of the image which can be considered as outliers. In this case, OADL can help us to train a better dictionary. Denoising scheme using OADL is based on outlier-aware sparse coding; see Section 2.

We used six benchmark images (Lena, Peppers, Cameraman, House, Mandril and Pirate) of size 256×256. At first we added a low variance (10^2) *i.i.d.* zero mean normal noise to all pixels of each image and then we defined 9, 36×36 square blocks equidistantly in each image (in 3 columns and 3 rows). In each stage, we added high variance (20^2) *i.i.d.* zero mean normal noise to the pixels of one block. To denoise each image, 40,000 blocks of size 8×8 were randomly chosen as the training signals. We use 20 iterations for dictionary learning and 5 iterations for denoising each image patch.

In order to check the performance of different denoising methods, we use PSNR defined as:

$$\text{PSNR} = 10 \log \frac{255^2}{\text{MSE}} \qquad (12)$$

where MSE is:

$$\text{MSE} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} \{\hat{\mathbf{F}}(i,j) - \mathbf{F}(i,j)\}^2 \qquad (13)$$

where m and n are image size, $\mathbf{F}$ is the original image and $\hat{\mathbf{F}}$ is the denoised image.

Figure 6 shows the averaged PSNR over six images for different number of outlier blocks. We can see the robustness of our algorithm against outliers. As the number of outlier blocks increases, OADL denoising scheme performs better than the denoising schemes based on MOD and DCT. On the other hand, MOD performance becomes superior to DCT which shows that learning of dictionary shows its advantages. To select λ, we perform our experiment for different λ between 5 and 100 with the step-size of 5 and choose the best λ (λ_{opt}), that is, the lambda that maximizes the resultant PSNR. Figure 7 shows λ_{opt} versus number of outlier blocks. As we expected, increasing the number of outlier blocks lead to decrease in λ_{opt} (For MOD and DCT, as $\sigma = 10$ is not optimum, we sweep σ between 10 and 20 with the step-size of 2 and select the best one in terms of maximizing PSNR).

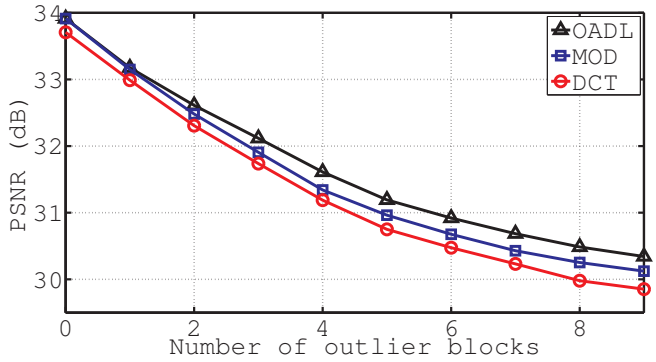

Fig. 6: Averaged PSNR over 6 different test images versus number of outlier blocks.

Table 1 shows the result of denoising, in terms of PSNR, for different number of outlier blocks (the denoising performance using DCT has also been shown as a baseline). The value of λ is chosen according to Fig. 7. As we can see in Table 1, denoising based on OADL has superior performance in most of benchmark images and number of outlier blocks. Just in "Mandril" image, denoising based on MOD outperforms OADL in most of outlier block values which we believe it originates from high frequency components available in the image. Our simulations show that increasing the number of atoms lead to a better performance in OADL than MOD in "Mandril" image but it also increases the complexity and time of simulations.

5. CONCLUSION

In this paper, we proposed a new dictionary learning algorithm which is robust to outliers. Unlike previous robust formulations for dictionary learning which target solely the sparse corruptions in the training signals, we considered an outlier as an unpleasant data vector included in the training

Table 1: Image denoising performance in PSNR for six different benchmark images. In each cell, top left, top right, bottom left and bottom right correspond to the PSNR (in dB) of noisy image, denoised image using DCT, denoised image using MOD and denoised image using OADL, respectively.

# Blocks	Lena		Peppers		Cameraman		House		Mandril		Pirate	
2	27.48	33.05	27.48	33.06	27.48	32.69	27.48	33.57	27.48	29.97	27.48	31.49
	33.21	**33.44**	33.29	**33.55**	32.84	**33.98**	33.87	**34.15**	**30.03**	29.89	31.63	**31.66**
4	26.92	31.77	26.92	31.80	26.92	31.53	26.92	32.18	26.92	29.27	26.92	30.57
	31.93	**32.31**	31.98	**32.38**	31.66	**31.98**	32.39	**32.83**	**29.37**	29.32	30.71	**30.84**
6	26.43	31.10	26.43	31.09	26.43	30.87	26.43	31.58	26.43	28.49	26.43	29.72
	31.33	**31.68**	31.32	**31.76**	31.03	**31.24**	31.87	**32.30**	**28.58**	28.50	29.92	**30.04**
8	26.00	30.54	26.00	30.56	26.00	30.32	26.00	30.99	26.00	**28.17**	26.00	**29.90**
	30.97	**31.27**	30.98	**31.38**	30.57	**30.78**	31.91	**32.08**	27.82	27.94	29.27	29.46

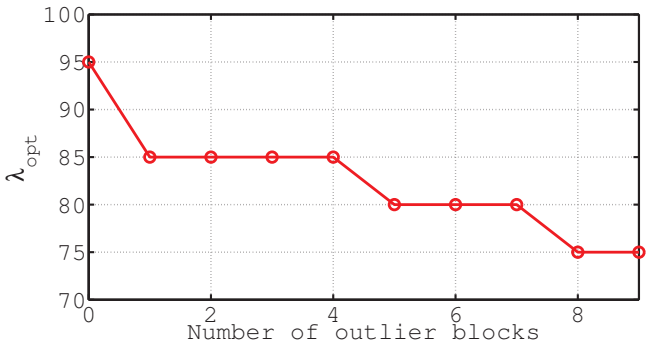

Fig. 7: Best λ, which maximizes PSNR, versus percentage of outliers.

signals whose structure does not match the others. We then propose an iterative algorithm to solve the resultant robust problem. We finally experimentally evaluated the performance of our proposed algorithm and traditional algorithms by performing some simulations on synthetic and real data. Simulations results emphasized on the robustness of the proposed method against outliers. Also, as simulations showed, to have a better performance in our algorithm, the number of outliers should be small in comparison to the total number of training signals (as a rule of thumb, the percentage of outliers being less than about 12% leads to promising results).

6. REFERENCES

[1] M. Elad, *Sparse and Redundant Representations*, Springer, 2010.

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

[3] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.

[4] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Proc.*, vol. 41, no. 12, pp. 3397–3415, 1993.

[5] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[6] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[7] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Proceedings of IEEE ICASSP*, 1999, vol. 5, pp. 2443 – 2446.

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

[9] C. Lu, J. Shi, and J. Jia, "Online robust dictionary learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 415 – 422.

[10] Z. Chen and Y. Wu, "Robust dictionary learning by error source decomposition," in *Proc. IEEE Int'l Conf. on Computer Vision (ICCV'13)*, 2013.

[11] G. Mateos and G. B. Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Trans. on Signal Proc.*, vol. 60, no. 10, pp. 5176–5190, 2012.

[12] S. S. Chen, D. D. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, pp. 129–159, 2001.