LOW MUTUAL AND AVERAGE COHERENCE DICTIONARY LEARNING USING CONVEX APPROXIMATION

Javad Parsa[1] *Mostafa Sadeghi*[1] *Massoud Babaie-Zadeh*[1] *Christian Jutten*[2]

[1]Electrical Engineering Department, Sharif University of Technology, Tehran, Iran
[2]GIPSA-lab, Univ. Grenoble Alpes, CNRS, Grenoble INP and Inst. Univ. de France.

ABSTRACT

In dictionary learning, a desirable property for the dictionary is to be of low mutual and average coherences. Mutual coherence is defined as the maximum absolute correlation between distinct atoms of the dictionary, whereas the average coherence is a measure of the average correlations. In this paper, we consider a dictionary learning problem regularized with the average coherence and constrained by an upper-bound on the mutual coherence of the dictionary. Our main contribution is then to propose an algorithm for solving the resulting problem based on convexly approximating the cost function over the dictionary. Experimental results demonstrate that the proposed approach has higher convergence rate and lower representation error (with a fixed sparsity parameter) than other methods, while yielding similar mutual and average coherence values.

Index Terms— Compressed sensing, sparse coding, mutual coherence, average coherence, dictionary learning

1. INTRODUCTION

1.1. Dictionary learning

Dictionary learning (DL) has been extensively utilized in a wide range of machine learning and signal processing applications, including image/signal enhancement and reconstruction [1, 2], and pattern recognition and classification [3]. A lot of algorithms have been proposed for this problem. To formally define it, given a training dataset $\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l]$, $\mathbf{y}_i \in \mathbb{R}^m$, a dictionary $\mathbf{D} \triangleq [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_n]$, $\mathbf{d}_i \in \mathbb{R}^m$, is learned in such a way that it provides sparse coefficients for $\mathbf{y}_i$'s. That is, the representations $\mathbf{X} \triangleq [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l]$, $\mathbf{x}_i \in \mathbb{R}^n$ are sufficiently sparse. To achieve this, the DL problem is usually formulated as follows [4]

$$(\mathbf{D}^*, \mathbf{X}^*) = \underset{\mathbf{D}\in\mathcal{D}, \mathbf{X}\in\mathcal{X}}{\operatorname{argmin}} \ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \tag{1}$$

in which, $\mathcal{D}$ and $\mathcal{X}$ are defined as $\mathcal{D} = \{\mathbf{D} : \ \forall i, \|\mathbf{d}_i\|_2^2 \leq 1\}$ and $\mathcal{X} = \{\mathbf{X} : \ \forall i, \|\mathbf{x}_i\|_1 \leq \tau\}$, where, $\|.\|_1$ denotes ℓ_1 norm. To solve (1), many dictionary learning algorithms have been introduced [2, 4, 5, 6, 7], which are mainly based on alternating minimization on $\mathbf{D}$ and $\mathbf{X}$. Some methods impose additional constraints on the dictionary $\mathbf{D}$ which can improve the performance [1, 8, 9]. Two important properties are reviewed in the next section.

1.2. Mutual and average coherences

One of the important properties of a dictionary is the maximum correlation between the columns of the dictionary which is called mutual coherence and denoted by $\mu(\mathbf{D})$ [10]. Another important property of a dictionary is the average correlation of dictionary columns, which is called average coherence and denoted by $\mu_{avg}(\mathbf{D})$. For a dictionary $\mathbf{D}$, these two parameters are respectively defined as:

$$\begin{aligned} \mu(\mathbf{D}) &= \max_{i\neq j} \frac{|\mathbf{d}_i^T \mathbf{d}_j|}{\|d_i\|_2 \|d_j\|_2}, \\ \mu_{avg}(\mathbf{D}) &= \sqrt{\frac{\|\mathbf{D}^T\mathbf{D} - \mathbf{I}\|_F^2}{n(n-1)}}. \end{aligned} \tag{2}$$

In dictionary learning, it is usually desired that the mutual coherence of the learned dictionary is small. This is because of two main reasons: On the one hand, it has been shown in [11] that a dictionary with low mutual coherence has well-conditioned sub-matrices. On the other hand, a signal with a sparse representation $\mathbf{x}$ with sparsity parameter s, i.e. with s nonzero coefficient s, can be recovered from $\mathbf{y} = \mathbf{D}\mathbf{x}$ through ℓ_1 minimization when [12]:

$$s \leq \frac{1}{2}(1 + \frac{1}{\mu(\mathbf{D})}). \tag{3}$$

According to (3), dictionaries with low mutual coherence are better for high s. However, the mutual coherence is lower bounded [13], and it can be shown that:

$$\mathbf{D} \in \mathbb{R}^{m\times n} \rightarrow \mu_{welch} \leq \mu(\mathbf{D}) \leq 1, \tag{4}$$

where the μ_{welch} is the Welch bound [13], defined as:

$$\mu_{welch} \triangleq \sqrt{\frac{n-m}{m(n-1)}}.$$

Furthermore, dictionaries with low average coherence are favorable in compressed sensing applications [14].

During recent years, many dictionary learning algorithms have been proposed trying to reduce mutual coherence [8, 9, 15]. A recent approach, called Gradient-based ISDL [16] (GSD), has been proposed in [17], which minimizes the following cost function:

$$\min_{\mathbf{D}\in\mathcal{D}, \mathbf{X}\in\mathcal{X}} \ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \frac{\lambda}{2}\|\mathbf{D}^T\mathbf{D} - \mathbf{H}\|_F^2, \tag{5}$$

where

$$\mathcal{H} \triangleq \left\{ \mathbf{H} \in \mathbb{R}^{n \times n} : \mathbf{H} = \mathbf{H}^T, h_{ii} = 1, \ \forall i \max_{i \neq j} |h_{ij}| \leq \mu_0 \right\},$$

in which $\mu_0 \geq \mu_{welch}$. To solve the above problem, an alternative minimization approach has been used in [17], while this problem does not yield closed form solution for updating the dictionary. In other words, the gradient of the cost function, $F(\mathbf{D})$, over $\mathbf{D}$ is computed as follows:

$$F(\mathbf{D}) \triangleq \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \frac{\lambda}{2}\|\mathbf{D}^T\mathbf{D} - \mathbf{H}\|_F^2 \Rightarrow \tag{6}$$

$$\nabla_{\mathbf{D}} F(\mathbf{D}) = 2(\mathbf{D}\mathbf{X} - \mathbf{Y})\mathbf{X}^T + 2\lambda \mathbf{D}(\mathbf{D}^T\mathbf{D} - \mathbf{H}). \tag{7}$$

Then solving $F(\mathbf{D})$ does not result to a closed-form solution, because (7) is non-linear over $\mathbf{D}$. For this reason, [17] uses gradient descent to update the dictionary, that is,

$$\mathbf{D}_{k+1} = \mathbf{D}_k - \alpha \nabla_{\mathbf{D}} F(\mathbf{D}_k). \tag{8}$$

Then, $\mathbf{H}$ is updated using the following formula, in which k denotes the iteration number:

$$h_{ij}^k = \begin{cases} \mu & i \neq j, |\mu| \leq \mu_0 \\ \operatorname{sgn}(\mu)\mu_0 & i \neq j, |\mu| \geq \mu_0 \\ 1 & i = j \end{cases} \tag{9}$$

in which, $\mu = u_{ij}^k$, the (i, j) entry of $\mathbf{U}_k = \mathbf{D}_k^T \mathbf{D}_k$.

In some older papers, e.g. [8, 9, 15], a cost function similar to (6) has been used by having identity matrix $\mathbf{I}$ instead of $\mathbf{H}$. And in [18], a combination of these two cost function have been used. In all of these papers, the cost function is non-linear over $\mathbf{D}$, so they all use an update equation of the form (8).

In this paper, we propose a new approach that approximates these non-convex cost functions over the dictionary by a convex function. This leads to a closed-form solution for the dictionary. As our simulations will confirm, the new algorithm results in improved performance in dictionary recovery.

The rest of the paper is organized as follows. Section 2 presents the main idea of our algorithm and related discussions. Then, the new algorithm is experimentally evaluated in Section 3.

2. THE PROPOSED ALGORITHM

To develop our proposed algorithm, consider the pair $(\mathbf{D}_a, \mathbf{X}_a)$, which is assumed to be known. Then, following [4], we can write

$$\begin{cases} \mathbf{D} = \mathbf{D}_a + \mathbf{D} - \mathbf{D}_a \\ \mathbf{X} = \mathbf{X}_a + \mathbf{X} - \mathbf{X}_a \end{cases} \tag{10}$$

$$\mathbf{D}\mathbf{X} = (\mathbf{D}_a + \mathbf{D} - \mathbf{D}_a)(\mathbf{X}_a + \mathbf{X} - \mathbf{X}_a) = \mathbf{D}_a\mathbf{X} + \mathbf{D}\mathbf{X}_a - \mathbf{D}_a\mathbf{X}_a + (\mathbf{D} - \mathbf{D}_a)(\mathbf{X} - \mathbf{X}_a) \tag{11}$$

$$\mathbf{D}^T\mathbf{D} = (\mathbf{D}_a + \mathbf{D} - \mathbf{D}_a)^T(\mathbf{D}_a + \mathbf{D} - \mathbf{D}_a) = \mathbf{D}_a^T\mathbf{D} + \mathbf{D}^T\mathbf{D}_a - \mathbf{D}_a^T\mathbf{D}_a + (\mathbf{D} - \mathbf{D}_a)^T(\mathbf{D} - \mathbf{D}_a). \tag{12}$$

Assuming that $\|(\mathbf{D} - \mathbf{D}_a)(\mathbf{X} - \mathbf{X}_a)\|_F$ and $\|(\mathbf{D} - \mathbf{D}_a)^T(\mathbf{D} - \mathbf{D}_a)\|_F$ are small, we can write

$$\mathbf{D}\mathbf{X} \approx \mathbf{D}_a\mathbf{X} + \mathbf{D}\mathbf{X}_a - \mathbf{D}_a\mathbf{X}_a \tag{13}$$

$$\mathbf{D}^T\mathbf{D} \approx \mathbf{D}_a^T\mathbf{D} + \mathbf{D}^T\mathbf{D}_a - \mathbf{D}_a^T\mathbf{D}_a, \tag{14}$$

from which we can propose the following approximate problem which is convex over $\mathbf{D}$:

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{X} \in \mathcal{X}} \ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \frac{\lambda}{2}\|\mathbf{D}^T\mathbf{D} - \mathbf{H}\|_F^2 \approx \tag{15}$$

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{X} \in \mathcal{X}} \ \|\mathbf{Y} + \mathbf{D}_a\mathbf{X}_a - \mathbf{D}_a\mathbf{X} - \mathbf{D}\mathbf{X}_a\|_F^2 + \frac{\lambda}{2}\|\mathbf{D}_a^T\mathbf{D} + \mathbf{D}^T\mathbf{D}_a - \mathbf{D}_a^T\mathbf{D}_a - \mathbf{H}\|_F^2, \tag{16}$$

To solve the above new problem, we use an alternating minimization approach, by optimizing the cost over one variable while fixing the other one. This procedure is summarized as follows.

2.1. Updating sparse coefficients (first term of (16)):

In this stage, suppose $\mathbf{D}_a = \mathbf{D}_{k-1}, \mathbf{D} = \mathbf{D}_k, \mathbf{X}_a = \mathbf{X}_k, \mathbf{Z}_k = \mathbf{Y} - (\mathbf{D}_k - \mathbf{D}_{k-1})\mathbf{X}_k$, then we can write:

$$\mathbf{X}_{k+1} = \min_{\mathbf{X} \in \mathcal{X}} \ \|\mathbf{Z}_k - \mathbf{D}_{k-1}\mathbf{X}\|_F^2. \tag{17}$$

To solve (17), we note that it is a sparse coding problem, with a lot of solvers existing in the literature [19].

2.2. Updating the dictionary (the two terms of (16)):

In this stage, we assume $\mathbf{X} = \mathbf{X}_a = \mathbf{X}_{k+1}$ and $\mathbf{D}_a = \mathbf{D}_k$. Then, the cost function for updating the dictionary of iteration $k + 1$ would be as follows:

$$G(\mathbf{D}) = \|\mathbf{Y} - \mathbf{D}\mathbf{X}_{k+1}\|_F^2 + \frac{\lambda}{2}\|\mathbf{D}_k^T\mathbf{D} + \mathbf{D}^T\mathbf{D}_k - \mathbf{D}_k^T\mathbf{D}_k - \mathbf{H}_k\|_F^2.$$

$$\nabla_{\mathbf{D}} G(\mathbf{D}) = (\mathbf{D}\mathbf{X}_{k+1} - \mathbf{Y})\mathbf{X}_{k+1}^T + \lambda\mathbf{D}_k(\mathbf{D}_k^T\mathbf{D} + \mathbf{D}^T\mathbf{D}_k - \mathbf{D}_k^T\mathbf{D}_k - \mathbf{H}_k) = \mathbf{D}\mathbf{X}_{k+1}\mathbf{X}_{k+1}^T + \lambda\mathbf{D}_k\mathbf{D}_k^T\mathbf{D} + \lambda\mathbf{D}_k\mathbf{D}^T\mathbf{D}_k - \mathbf{Y}\mathbf{X}_{k+1}^T - \lambda\mathbf{D}_k(\mathbf{D}_k^T\mathbf{D}_k + \mathbf{H}_k)$$

To minimize $G(\mathbf{D})$, $\nabla_{\mathbf{D}} G(\mathbf{D})$ is set to zero. By defining the auxiliary variables

$$\begin{cases} \mathbf{W}_k = (\mathbf{X}_{k+1})(\mathbf{X}_{k+1})^T \\ \mathbf{A}_k = \mathbf{H}_k + \mathbf{D}_k^T\mathbf{D}_k \\ \mathbf{C}_k = \mathbf{Y}(\mathbf{X}_{k+1})^T + \lambda\mathbf{D}_k\mathbf{A}_k \end{cases}, \tag{18}$$

the following equation is obtained to be solved in $\mathbf{D}$:

$$\mathbf{D}\mathbf{W}_k + \lambda\mathbf{D}_k\mathbf{D}^T\mathbf{D}_k + \lambda\mathbf{D}_k\mathbf{D}_k^T\mathbf{D} = \mathbf{C}_k. \tag{19}$$

By using the substitutions

$$\begin{cases} \mathbf{M}_1 = \mathbf{D}\mathbf{W}_k + \lambda\mathbf{D}_k\mathbf{D}_k^T\mathbf{D} \\ \mathbf{M}_2 = \lambda\mathbf{D}_k\mathbf{D}^T\mathbf{D}_k \\ \mathbf{M}_1 + \mathbf{M}_2 = \mathbf{C}_k \to \operatorname{vec}(\mathbf{M}_1) + \operatorname{vec}(\mathbf{M}_2) = \operatorname{vec}(\mathbf{C}_k) \end{cases}, \tag{20}$$

and using [20]:

$$\sum_n \mathbf{A}_n\mathbf{X}\mathbf{B}_n = \mathbf{R} \rightarrow (\sum_n \mathbf{B}_n^T \otimes \mathbf{A}_n)\text{vec}(\mathbf{X}) = \text{vec}(\mathbf{R}),$$

where $\otimes$ denotes Kronecker product, one obtains

$$\begin{cases} \text{vec}(\mathbf{M}_1) = (\mathbf{W}_k^T \otimes \mathbf{I}_m + \mathbf{I}_n \otimes (\lambda\mathbf{D}_k^T\mathbf{D}_k))\text{vec}(\mathbf{D}) \\ \text{vec}(\mathbf{M}_2) = \lambda(\mathbf{D}_k^T \otimes \mathbf{D}_k)\text{vec}(\mathbf{D}^T) \end{cases}$$

$$(\mathbf{W}_k^T \otimes \mathbf{I}_m + \mathbf{I}_n \otimes (\lambda\mathbf{D}_k^T\mathbf{D}_k))\text{vec}(\mathbf{D}) \\ + \lambda(\mathbf{D}_k^T \otimes \mathbf{D}_k)\text{vec}(\mathbf{D}^T) = \text{vec}(\mathbf{C}_k). \tag{21}$$

To solve the above equation with respect to $\mathbf{D}$, we first determine a matrix $\mathbf{B}_k$ such that $(\mathbf{D}_k^T \otimes \mathbf{D}_k)\text{vec}(\mathbf{D}^T)$ is equal to $\mathbf{B}_k\text{vec}(\mathbf{D})$. It is not difficult to see that such a $\mathbf{B}_k$ is obtained as:

$$\begin{cases} \mathbf{Q}_k \triangleq (\mathbf{D}_k^T \otimes \mathbf{D}_k), \mathbf{D}_k \in \mathbb{R}^{m\times n}, 1 \leq i \leq n, 1 \leq j \leq m \\ \mathbf{B}_k(:, ((i-1)m+j)) \triangleq \mathbf{Q}_k(:, (i+(j-1)n)) \end{cases}$$

So

$$(\mathbf{W}_k^T \otimes \mathbf{I}_m + \mathbf{I}_n \otimes (\lambda\mathbf{D}_k^T\mathbf{D}_k) + \lambda\mathbf{B}_k)\text{vec}(\mathbf{D}_{k+1}) = \text{vec}(\mathbf{C}_k) \\ \Rightarrow \text{vec}(\mathbf{D}_{k+1}) = (\mathbf{W}_k^T \otimes \mathbf{I}_m + \mathbf{I}_n \otimes (\lambda\mathbf{D}_k^T\mathbf{D}_k) + \lambda\mathbf{B}_k)^{-1} \\ \text{vec}(\mathbf{C}_k), \tag{22}$$

which determines $\mathbf{D}_{k+1}$ in closed-form.

2.3. Updating H:

It is updated by (9).

Note that our approach can be used on many dictionary learning algorithms to convexify the cost function. As an example, we apply it here on GSD [17] and RAMC [18]. The two new obtained methods are called Convex-GSD and Convex-RAMC.

The final algorithm (Convex-RAMC) is summarized in Algorithms 1 and Convex-GSD algorithm is achieved when $\beta_1 = 0$ in Algorithm 1.

Algorithm 1 The proposed algorithm (Convex-RAMC)

```
Input:$\mathbf{Y}, \mathbf{D}^0$, $s$ (sparsity parameter)
Initialization: Set initial dictionary $\mathbf{D}^1 = \mathbf{D}^0$.
for $k = 1$ to MaxIteration do
    Sparse approximation:$\mathbf{X}_{k+1}$ =OMP($\mathbf{Z}_k$, $\mathbf{D}_{k-1}$, $s$).
    Dictionary update:Dictionary is updated by equations (22).
    Normalize the columns of $\mathbf{D}_{k+1}$.
    Update $\mathbf{H}_{k+1}$ using (9) and replace $\mathbf{H}_{k+1}$ with
    $\beta_1\mathbf{I} + \beta_2\mathbf{H}_{k+1}$ in (18), in which $0 \leq \beta_1 \leq 1, 0 \leq \beta_2 \leq 1$ and $\beta_1 + \beta_2 = 1$.
end for
```

3. SIMULATION RESULTS

In this section, we experimentally evaluate Convex-GSD and Convex-RAMC, and compare them with GSD [17], RAMC [18] and MOD [6] for recovering a known dictionary. Our simulations were performed in MATLAB R2017b environment on a system with 4.00 GHz I7 CPU and 16 GB RAM, under Microsoft Windows 10 operating system. As a rough measure of complexity, we will mention the run times of the algorithms. The performance measures are root mean square error (RMSE) defined as $\varepsilon_k = \frac{\|\mathbf{Y}-\mathbf{D}^k\mathbf{X}^k\|_F}{\sqrt{ml}}$ [18], percentage of atom recovery, mutual coherence and average coherence (2). Assuming that $\mathbf{D}_t$ is the true dictionary and $\mathbf{D}$ is the recovered dictionary, we say that the ith atom of the dictionary $\mathbf{D}$ is successfully recovered if

$$\min_{i\neq j}(1- \mid \mathbf{D}(:,i)^T\mathbf{D}_t(:,j) \mid) < 0.01. \tag{23}$$

For OMP, we used the available MATLAB code at http://www.cs.technion.ac.il/~ronrubin/software.html. We generated a Gaussian random matrix $\mathbf{D}_t \in \mathbb{R}^{20\times 50}$ with zero mean and unit variance. Then 2500 training data $\{\mathbf{y}_i\}_{i=1}^{2500}$ were generated by random linear combinations of dictionary atoms. According to the size of the dictionary, the Welch bound (4) is computed as $\mu_{welch} = 0.1749$ and we chose $\mu_0 = \mu_{welch}$. In our simulation, we assume $\beta_1 = 0.2, \beta_2 = 0.8$, SNR $= 30$dB and $s = 7$ (sparsity parameter). In all simulations, the sparsity parameter (s) is constant while the hyper-parameter λ (balancing the two terms of the cost $F(\mathbf{D})$) has two values 5 and 10. We performed 2000 iterations between the sparse coding and dictionary updating. The dictionary was initialized by randomly choosing different signals from the training set followed by a normalization. We repeated all simulations 400 times and the averaged results are reported here.

Run times of algorithms are also compared as a rough measure of computational complexity. The average running times and iterations number of the algorithms for achieving a percentage of recovery equal to 80 are shown in Table I.

Figures 1 to 8 are the results of the simulation of our algorithms and its comparisons with other mentioned algorithms. According to all the figures and table, our methods have higher convergence rate and lower RMSE than the other algorithms while mutual and average coherence of our methods are similar to those achieved by GSD and RAMC. The overall running time to converge of our methods are lower than the other methods. According to Fig 3 and 4, for $\lambda = 5$ mutual and average coherences are similar for all the methods, while our methods have higher convergence rate than the other methods. In Figs.7 and 8, mutual and average coherence of GSD are a little bit lower than Convex-GSD but the final percentage of recovery of Convex-GSD is 15 percent higher than GSD (see Fig. 5).

4. CONCLUSION

In this paper, we proposed a new approach to convexify the cost function of dictionary learning problem with low mutual and average coherence. According to our simulations on synthetic dictionary recovery, our approach increases the convergence rate and decreases RMSE, while mutual and average coherence of our algorithms are reduced well.

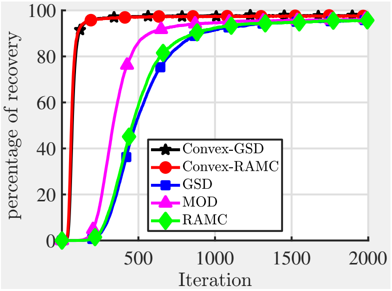

Fig. 1: Evaluation of percentage of recovery with assumptions: $\lambda = 5$.

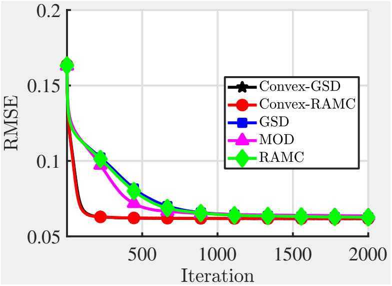

Fig. 2: Evaluation of RMSE with assumptions: $\lambda = 5$. The graphs of Convex-GSD and Convex-RAMC are almost superimposed.

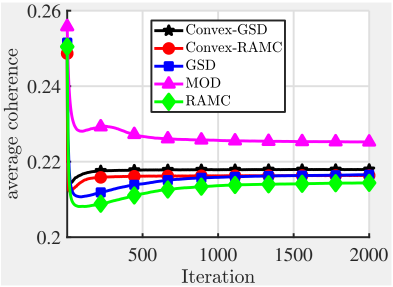

Fig. 3: Evaluation of average coherence with assumptions: $\lambda = 5$.

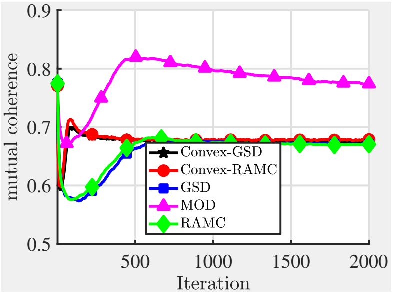

Fig. 4: Evaluation of mutual coherence with assumptions: $\lambda = 5$.

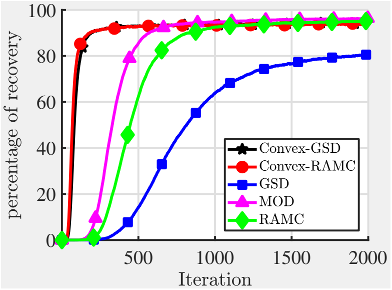

Fig. 5: Evaluation of percentage of recovery with assumptions: $\lambda = 10$.

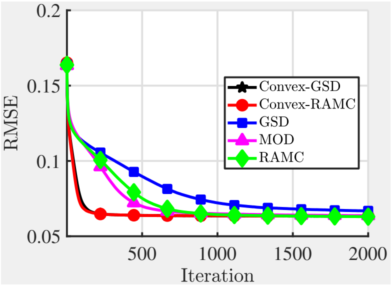

Fig. 6: Evaluation of RMSE with assumptions: $\lambda = 10$. The graphs of Convex-GSD and Convex-RAMC are almost superimposed.

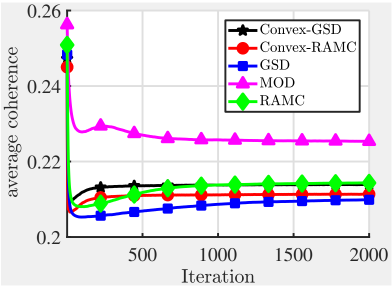

Fig. 7: Evaluation of average coherence with assumptions: $\lambda = 10$.

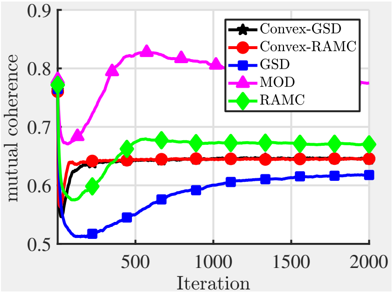

Fig. 8: Evaluation of mutual coherence with assumptions: $\lambda = 10$.

Table 1: Number of iterations and average running time (in seconds) for achieving percentage of recovery= 80. Average running times are reported in parentheses. In this table and all figures, $s = 7$ and SNR $= 30$dB are supposed.

Algorithm	$\lambda = 5$	$\lambda = 10$
Convex-GSD	91 (8.3s)	123 (12.1s)
Convex-RAMC	83 (7.5s)	102 (10.1s)
GSD	697 (18.6s)	1913 (59.3s)
RAMC	643 (17.2s)	626 (19.4s)
MOD	452 (9.4s)	449 (10.5s)

5. REFERENCES

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

[2] 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.

[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] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Dictionary learning for sparse representation: A novel approach," *IEEE Signal Proc. Letters*, vol. 20, no. 12, pp. 1195–1198, 2013.

[5] 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.

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

[7] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Learning over-complete dictionaries based on atom-by-atom updating," *IEEE Trans. on Signal Proc.*, vol. 62, no. 4, pp. 883–891, 2014.

[8] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Learning dictionaries with bounded self-coherence," *IEEE Signal Proc. Letters*, vol. 19, no. 12, pp. 861–864, 2012.

[9] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "A new algorithm for learning overcomplete dictionaries," in *Proceedings of 21th European Signal Processing Conference (EUSIPCO 2013)*, 2013.

[10] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[11] J. A. Tropp, "On the conditioning of random subdictionaries," *Appl. Computat. Harmon. Anal.*, vol. 25, no. 1, pp. 1–24, 2008.

[12] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *Proc. Nat. Aca. Sci*, vol. 100, no. 5, pp. 2197–2202, 2003.

[13] T. Strohmer and Heath R. W., "Grassmannian frames with applications to coding and communication," *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[14] W. Chen, M. R. D. Rodrigues, and J. I. Wassell, "Projection design for statistical compressive sensing: A tight frame based approach," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2016–2029, 2013.

[15] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Regularized low-coherence overcomplete dictionary learning for sparse signal decomposition," in *Proceedings of 24th European Signal Processing Conference (EUSIPCO 2016)*, 2016.

[16] D. Barchiesi and M. D. Plumbley, "Learning incoherent dictionaries for sparse approximation using iterative projections and rotations," *IEEE Trans. on Signal Proc.*, vol. 61, no. 8, pp. 2055–2065, 2013.

[17] G. li, Z. Zhu, H. Bai, and A. Yu, "A new framework for designing incoherent sparsifying dictionaries," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[18] J. Parsa, M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Joint low mutual and average coherence dictionary learning," in *Proceeding of 26th European Signal Processing Conference(EUSIPCO 2018)*, 2018.

[19] 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.

[20] K. B. Petersen and M.S. Pedersen, "The matrix cookbook," 2008.