A New Algorithm for Dictionary Learning Based on Convex Approximation

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

1) Electrical Engineering Department, Sharif University of Technology, Tehran, IRAN.
2) GIPSA-Lab, University of Grenoble Alpes, Grenoble, France.

Dictionary learning

- Dictionary learning problem [1]:

$$\begin{cases} (\boldsymbol{D}^*, \boldsymbol{X}^*) = \underset{\boldsymbol{D}\in\xi, \boldsymbol{X}\in\chi}{\operatorname{argmin}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2, \\ \chi = \{\mathbf{X}: \forall i, \|\boldsymbol{x}_i\|_1 \leq \Omega\}, \\ \xi = \{\boldsymbol{D}: \forall j, \|\boldsymbol{d}_j\|_2 \leq 1\}. \end{cases}$$

Dictionary learning problem is a non-convex problem over $\boldsymbol{D}$ and $\boldsymbol{X}$ jointly.

Dictionary learning

- Dictionary learning by alternative minimization using:

1) Sparse representation:

$$\boldsymbol{X}^{(k+1)} = \underset{\boldsymbol{X}\in\chi}{\operatorname{argmin}} \left\| \boldsymbol{Y} - \boldsymbol{D}^{(k)}\boldsymbol{X} \right\|_F^2 \quad (2)$$

2) Dictionary update:

$$\boldsymbol{D}^{(k+1)} = \underset{\boldsymbol{D}\in\xi}{\operatorname{argmin}} \left\| \boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}^{(k+1)} \right\|_F^2 \quad (3)$$

Stage 1 is an ordinary sparse coding problem [2]–[5], which can be done, for example, by Orthogonal Matching Pursuit (OMP) [6].

Convex dictionary learning [7]

- Convex approximation[7]:

$$\boldsymbol{D} = \boldsymbol{D}_0 + (\boldsymbol{D} - \boldsymbol{D}_0) \quad , \quad \boldsymbol{X} = \boldsymbol{X}_0 + (\boldsymbol{X} - \boldsymbol{X}_0)$$

$$\boldsymbol{DX} = [\boldsymbol{D}_0 + (\boldsymbol{D} - \boldsymbol{D}_0)][\boldsymbol{X}_0 + (\boldsymbol{X} - \boldsymbol{X}_0)]$$

$$\Longrightarrow \quad \boxed{\boldsymbol{DX} \approx \boldsymbol{D_0X} + \boldsymbol{DX_0} - \boldsymbol{D_0X_0} \qquad (4)}$$

in which, it is assumed that $\|(\boldsymbol{D} - \boldsymbol{D}_0)(\boldsymbol{X} - \boldsymbol{X}_0)\|_F$ is small.

- Convex dictionary learning problem:

$$\boxed{(\boldsymbol{D}^*, \boldsymbol{X}^*) = \underset{\boldsymbol{D}\in\xi, \boldsymbol{X}\in\chi}{\operatorname{argmin}} \|\boldsymbol{Y} + \boldsymbol{D_0X_0} - \boldsymbol{D_0X} - \boldsymbol{DX_0}\|_F^2 \qquad (5)}$$

Convex dictionary learning [7]

- Dictionary learning by alternative minimization using:
  1) Sparse representation:

$$\boldsymbol{D} = \boldsymbol{D}_0 = \boldsymbol{D}^{(k)},$$
$$\boldsymbol{X}^{(k+1)} = \underset{\boldsymbol{X}\in\chi}{\operatorname{argmin}} \left\|\boldsymbol{Y} - \boldsymbol{D}^{(k)}\boldsymbol{X}\right\|_F^2 \quad (6)$$

  2) Dictionary update:

$$\boldsymbol{X} = \boldsymbol{X}^{(k+1)}, \boldsymbol{X}_0 = \boldsymbol{X}^{(k)}, \boldsymbol{D}_0 = \boldsymbol{D}^{(k)}$$
$$\boldsymbol{D}^{(k+1)} = \underset{\boldsymbol{D}\in\xi}{\operatorname{argmin}} \left\|\boldsymbol{Y} - \boldsymbol{D}^{(k)}\left(\boldsymbol{X}^{(k+1)} - \boldsymbol{X}^{(k)}\right) - \boldsymbol{D}\boldsymbol{X}^{(k)}\right\|_F^2 \quad (7)$$

Another view point to Equation (4)

- Convex approximation by first order term of Taylor series:

$$\boldsymbol{H} = \boldsymbol{DX}, \boldsymbol{H}(j,i) = \boldsymbol{h}_{ji} = \boldsymbol{d}_{[j]}^{\mathrm{T}} \boldsymbol{x}_i, \boldsymbol{z} = \begin{bmatrix} \boldsymbol{d}_{[j]} \\ \boldsymbol{x}_i \end{bmatrix}, \boldsymbol{Q} \triangleq \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$$

$\boldsymbol{d}_{[j]}^{T}$ denotes the j-th row of $\boldsymbol{D}$ and $\boldsymbol{x}_i$ denotes the i-th column of $\boldsymbol{X}$.

$$\mathrm{f}_{ij}(\boldsymbol{z}) = \frac{\boldsymbol{1}}{\boldsymbol{2}} \boldsymbol{z}^{T} \boldsymbol{Q} \boldsymbol{z} = \boldsymbol{d}_{[j]}^{T} \boldsymbol{x}_i \approx \mathrm{f}_{ij}(\boldsymbol{z}_{\boldsymbol{0}}) + \nabla \mathrm{f}_{ij}(\boldsymbol{z}_{\boldsymbol{0}})^{\mathrm{T}} (\boldsymbol{z} - \boldsymbol{z}_{\boldsymbol{0}})$$

$$\mathrm{f}_{ij}(\boldsymbol{z}) = \boldsymbol{d}_{[j]}^{\mathrm{T}} \boldsymbol{x}_i \approx \boldsymbol{d}_{[j0]}^{T} \boldsymbol{x}_i + \boldsymbol{d}_{[j]}^{\mathrm{T}} \boldsymbol{x}_{i0} - \boldsymbol{d}_{[j0]}^{\mathrm{T}} \boldsymbol{x}_{i0} \quad (8)$$

Extending (8) to all elements of $\boldsymbol{H}$ results in (4).

Our idea

- Using (4) is possible in the 4 following ways:

1) **X-update:** $\mathbf{D} = \mathbf{D}_0 = \mathbf{D}^{(k)}$. $\mathbf{X}_0$ has no effect.
   D-update: $\mathbf{X} = \mathbf{X}_0 = \mathbf{X}^{(k+1)}$. $\mathbf{D}_0$ has no effect.
2) **X-update:** $\mathbf{D} = \mathbf{D}^{(k)}$, $\mathbf{D}_0 = \mathbf{D}^{(k-1)}$. $\mathbf{X}_0 = \mathbf{X}^{(k)}$.
   D-update: $\mathbf{X} = \mathbf{X}^{(k+1)}$, $\mathbf{X}_0 = \mathbf{X}^{(k)}$. $\mathbf{D}_0 = \mathbf{D}^{(k)}$.
3) **X-update:** $\mathbf{D} = \mathbf{D}_0 = \mathbf{D}^{(k)}$. $\mathbf{X}_0$ has no effect.
   D-update: $\mathbf{X} = \mathbf{X}^{(k+1)}$, $\mathbf{X}_0 = \mathbf{X}^{(k)}$. $\mathbf{D}_0 = \mathbf{D}^{(k)}$.
4) **X-update:** $\mathbf{D} = \mathbf{D}^{(k)}$, $\mathbf{D}_0 = \mathbf{D}^{(k-1)}$. $\mathbf{X}_0 = \mathbf{X}^{(k)}$.
   D-update: $\mathbf{X} = \mathbf{X}_0 = \mathbf{X}^{(k+1)}$. $\mathbf{D}_0$ has no effect.

- Which one is better? $\rightarrow$ The question addressed in this paper

Our idea (*continued*)

- The first case reduces to the traditional DL problem described by (2) and (3).
- The third case is the one used in [7], which is described by (6) and (7).
- The second and fourth cases are being considered in this paper.

- Note that all of these four cases can be applied on almost any DL algorithm and obtain new algorithms. To call the resulted algorithms, we add prefixes 'UD1' to 'UD4' to the name of the original algorithm, corresponding to the cases 1 to 4, respectively ('UD' stands for 'UpDated').

Summary of our proposed method

- Our proposed method uses the fourth case and "UD4" in the name of algorithms denote our proposed algorithms.

1) Sparse representation:

$$\boldsymbol{D} = \boldsymbol{D}^{(k)}, \boldsymbol{D}_0 = \boldsymbol{D}^{(k-1)}, \boldsymbol{X}_0 = \boldsymbol{X}^{(k)}$$

$$\boldsymbol{X}^{(k+1)} = \underset{\boldsymbol{X}\in\chi}{\operatorname{argmin}} \left\| \boldsymbol{Y} - \left(\boldsymbol{D}^{(k)} - \boldsymbol{D}^{(k-1)}\right)\boldsymbol{X}^{(k)} - \boldsymbol{D}^{(k-1)}\boldsymbol{X} \right\|_F^2$$

2) Dictionary update:

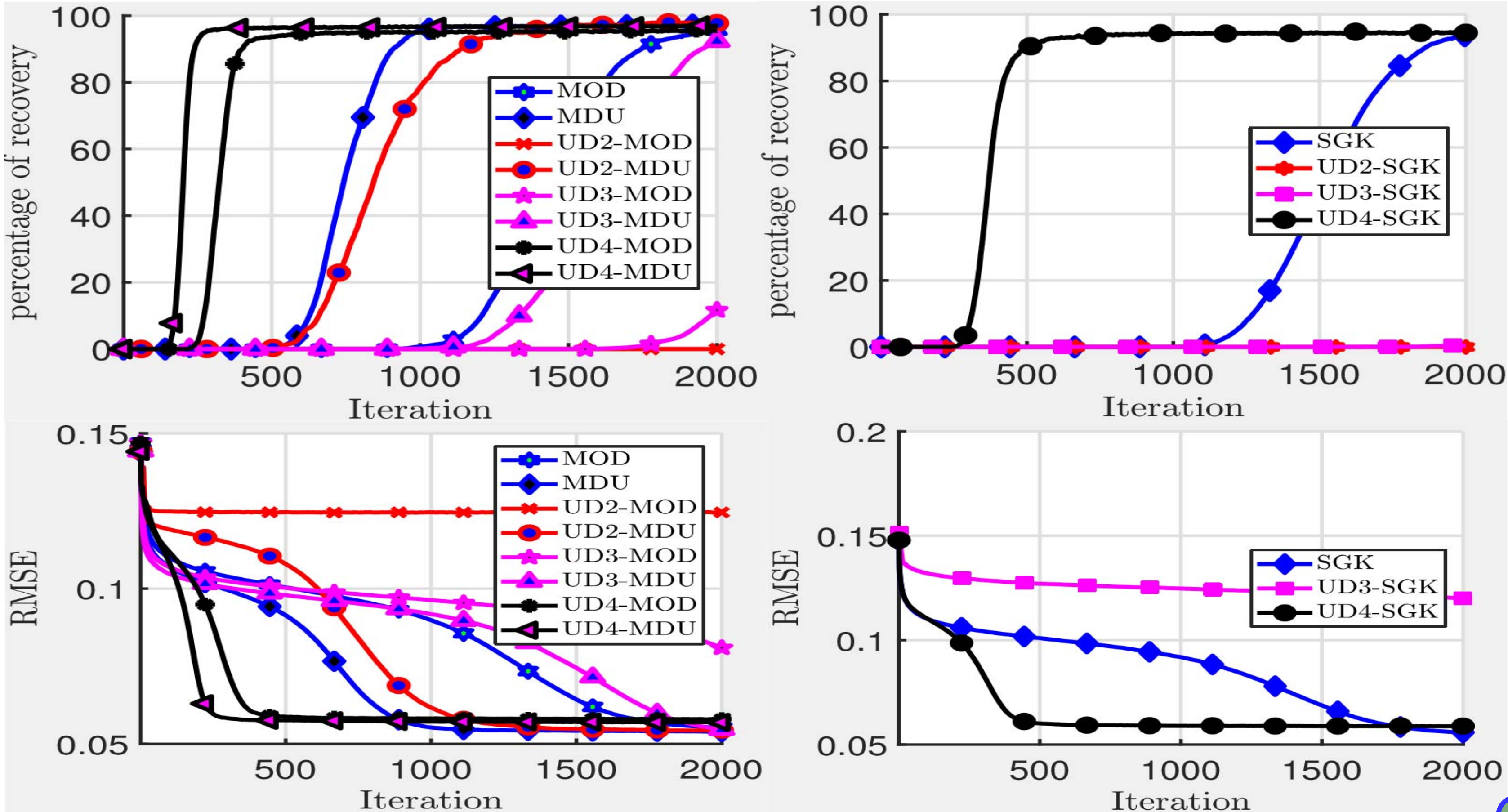
$$\boldsymbol{D}^{(k+1)} = \underset{\boldsymbol{D}\in\xi}{\operatorname{argmin}} \left\| \boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}^{(k+1)} \right\|_F^2$$

Simulation Results

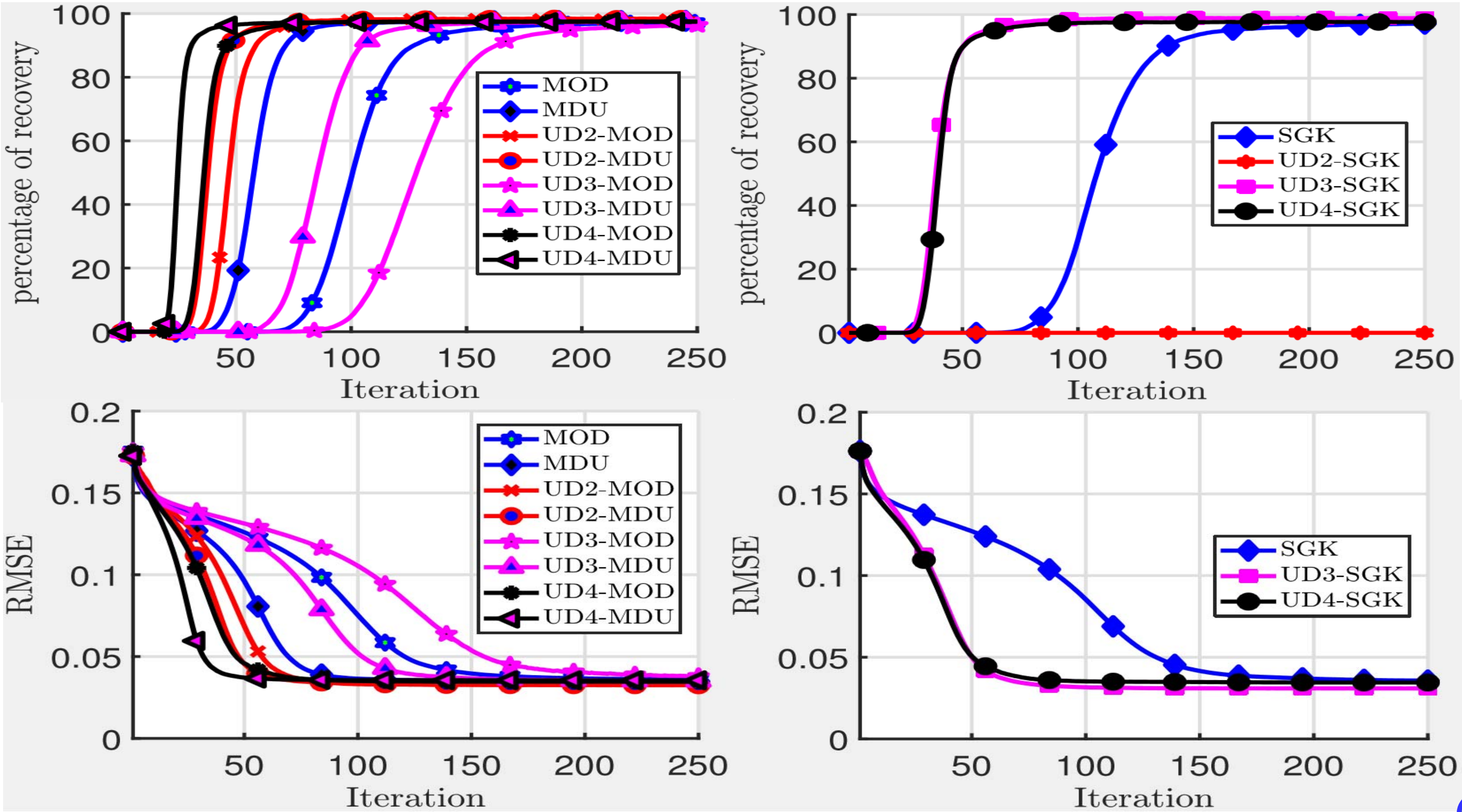
- Let's apply all cases on the MOD[8], SGK[9] and MDU[10], to evaluate their performance experimentally. The performance measures are root mean square error (RMSE) defined as $\varepsilon_K = \frac{\|\boldsymbol{Y}-\boldsymbol{D}^k\boldsymbol{X}^k\|_F}{\sqrt{mL}}$ and percentage of atom recovery. Assuming that $\boldsymbol{D_t}$ is the true dictionary and $\boldsymbol{D}$ is the recovered dictionary, we say that the $i-th$ atom of dictionary $\boldsymbol{D}$ is successfully recovered if:

$$min_j(1-|\boldsymbol{D}(:,i)^T\boldsymbol{D_t}(:,j)|) < 0.01.$$

- $\boldsymbol{D} \in \mathbb{R}^{40\times 100}$, $s = 15, 10$ (sparsity level), 3000 training data, SNR=30 dB



According to these simulations, our algorithms (denoted by UD4) have higher convergence rate and lower RMSE in comparison to the other algorithms. Sparsity level is 15.



According to these simulations, our algorithms (denoted by UD4) have higher convergence rate and lower RMSE in comparison to the other algorithms. Sparsity level is 10.

TABLE I: Average running times (in seconds) for achieving percentage of recovery=85. Those of our proposed algorithm (denoted by UD4) are reported in parentheses. The second case (UD2) diverges most of the time, so it is not in the table. A dash sign indicates divergence.

Algorithm	$s = 5$	$s = 10$	$s = 15$
MOD	**0.47** (0.36)	**5.7** (2.14)	**109.72** (25.78)
MDU	**2** (1.6)	**21.36** (9.12)	**315.06** (82.59)
SGK	**1** (0.68)	**10.05** (3.68)	**160.93** (39.90)
UD3-MOD	**0.65** (0.36)	**7.41** (2.14)	**172.72** (25.78)
UD3-MDU	**3.41** (1.6)	**21.36** (9.12)	**680.65** (82.59)
UD3-SGK	**0.86** (0.68)	**3.68** (3.68)	– (39.90)

According to Table I, if s increases, the difference in the convergence rate and running time between our approach and the other algorithms also increases.

Conclusions

- We showed that the main idea of [7] could actually been used in different ways, and the way it had been used in [7] was not the best one. We then experimentally proposed to use another choice that results in a highly better performance, in terms of both accuracy and speed.

- Note that his approach can be applied on almost any existing DL algorithm to obtain modified versions.

References

[1] 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.

[2] 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.

[3] H. Mohimani, M. Babaie-Zadeh, and Ch. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed L0 norm," IEEE Trans. on Signal Processing, vol. 57, pp. 289–301, 2009.

[4] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," Applied and Computational Harmonic Analysis, vol. 27, no. 3, pp. 265–274, 2009.

[5] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Gunturk, "Iteratively re-weighted least squares minimization for sparse recovery," Communications on Pure and Applied Mathematics, vol. 63, no. 1, pp. 1–38, Jan. 2010..

[6] J. A. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Trans. Info. Theory, vol. 53, no. 12, pp. 4655–4666, 2007.

[7] 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.

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

[9] S. K. Sahoo and A. Makur, "Dictionary training for sparse representation as generalization of K-Means clustering," IEEE Signal Proc. Letters, vol. 20, no. 6, pp. 587–590, 2013.

[10] L. N. Smith and M. Elad, "Improving dictionary learning: Multiple dictionary updates and coefficient reuse," IEEE Signal Proc. Letters, vol. 20, no. 1, pp. 79–82, 2013.