New Dictionary Learning Methods for Two-Dimensional Signals

Firooz Shahriari-Mehr*, Javad Parsa*, Massoud Babaie-Zadeh*
Christian Jutten**

*Electrical Engineering Dep., Sharif University of Technology, Tehran, Iran.
**GIPSA-Lab, University of Grenoble Alpes, Grenoble, France.

January 2021

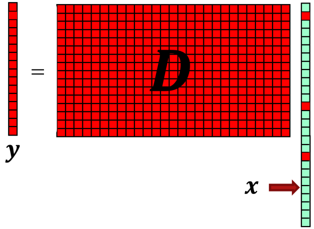
Outline

1. Introduction
   - Sparse Representation and Dictionary Learning for One-Dimensional Signals
   - Sparse Representation and Dictionary Learning for Two-Dimensional Signals
2. Proposed Methods
   - 2D-MOD
   - 2D-CMOD
3. Experimental Results
   - Recovery of Known Dictionary
   - Image Denoising

One-Dimensional Sparse Representation

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad s.t. \quad \mathbf{y} = \mathbf{D}\mathbf{x}$$

- $\mathbf{y} \in \mathbb{R}^n$ $\rightarrow$ One-Dimensional Signal
- $\mathbf{D} = [\mathbf{d}_i]$, $\mathbf{D} \in \mathbb{R}^{n \times m}$ $\rightarrow$ Dictionary, $\mathbf{d}_i \in \mathbb{R}^n$ $\rightarrow$ atom
- $\mathbf{x} \in \mathbb{R}^m$ $\rightarrow$ Sparse Signal Representation
- $m > n$ $\rightarrow$ Underdetermind Linear System of Equations



One-Dimensional Dictionary Learning

- $\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_L] \in \mathbb{R}^{n \times L}$ → Training Signals
- $\mathbf{X} \triangleq [\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_L] \in \mathbb{R}^{m \times L}$ → Sparse Representation matrix

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

$$\mathcal{D} \triangleq \{\mathbf{D} : \forall i, \|\mathbf{d}_i\|_2^2 = 1\}$$

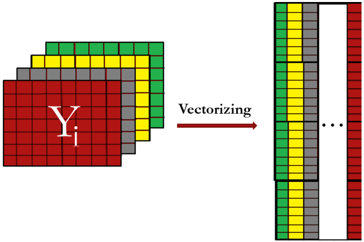
$$\mathcal{X} \triangleq \{\mathbf{X} : \forall i, \|\mathbf{x}_i\|_0 \leq \tau\}$$

- $\mathbf{X} \in \mathcal{X}$ → Impose Sparsity
- $\mathbf{D} \in \mathcal{D}$ → Avoid scaling ambiguity
- General approach: Alternating Minimization → MOD (Engan et al., 1999) - KSVD (Aharon et al., 2006)

Vectorizing and its consequent problems

Two-Dimemsional Signals?

- vectorize each signal and use usual 1D methods



- $\mathbf{Y}_i \in \mathbb{R}^{20\times20} \rightarrow \mathbf{y}_i \in \mathbb{R}^{400}$
- $\mathbf{D} \in \mathbb{R}^{400\times1600}$

Problems:

- Memory Consumption
- Computational Cost

Two-Dimensional Signal Representation

$$\mathbf{Y} = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} x_{ij} \mathbf{\Phi}_{ij}$$

- $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$
- $\mathbf{X} \in \mathbb{R}^{m_1 \times m_2}$
- $\mathbf{\Phi}_{ij} \in \mathbb{R}^{n_1 \times n_2}$

Separable Structure of 2D atoms in DIP[1]

$$\mathbf{\Phi}_{ij} = \mathbf{a}_i \mathbf{b}_j^T \quad \rightarrow \quad \mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{B}^T \quad \Longleftrightarrow \quad \mathbf{y} = \mathbf{D}\mathbf{x}$$

- $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{m_1}] \in \mathbb{R}^{n_1 \times m_1}$
- $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{m_2}] \in \mathbb{R}^{n_2 \times m_2}$
- $\mathbf{D} = \mathbf{B} \otimes \mathbf{A} \in \mathbb{R}^{n_1 n_2 \times m_1 m_2}$, $\mathbf{y} \in \mathbb{R}^{n_1 n_2}$, $\mathbf{x} \in \mathbb{R}^{m_1 m_2}$

[1] Ghaffari, Babaie-Zadeh and Jutten, "Sparse decomposition of two dimensional signals", ICASSP, 2009

Two-Dimensional Sparse Representation

Find the sparse representation of signal $\mathbf{Y}$ in separable dictionaries $\mathbf{A}$ and $\mathbf{B}$[2]

$$\boxed{\min_{\mathbf{X}} \|\mathbf{X}\|_0 \quad s.t. \quad \mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{B}^T}$$

Methods:

1. 2D-SL0[2]
2. 2D-OMP[3]

[2] Gha ari, Babaie-Zadeh and Jutten, \Sparse decomposition of two dimensional signals", ICASSP, 2009

[3] Fang, Wu and Huang, \2D sparse signal recovery via 2D orthogonal matching pursuit", SCIS, 2012

Two-Dimensional Dictionary Learning

- $\mathcal{Y} = (\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_L)$
- $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_L)$

$$\boxed{(\mathbf{A}^*, \mathbf{X}^*, \mathbf{B}^*) = \underset{\mathbf{X}_i \in \mathcal{X}_i, \mathbf{A} \in \mathcal{A}, \mathbf{B} \in \mathcal{B}}{\operatorname{argmin}} \sum_{i=1}^{L} \|\mathbf{Y}_i - \mathbf{A}\mathbf{X}_i\mathbf{B}^T\|_F^2} \quad (1)$$

- $\mathcal{A} \triangleq \{\mathbf{A} : \forall i, \|\mathbf{a}_i\|_2^2 = 1\}$
- $\mathcal{B} \triangleq \{\mathbf{B} : \forall i, \|\mathbf{b}_i\|_2^2 = 1\}$
- $\mathcal{X}_i \triangleq \{\mathbf{X}_i : \|\mathbf{X}_i\|_0 \leq \tau\}$

- The first two constraints avoid scaling ambiguity
- The last constraint impose the sparsity of representations

* SeDiL Algorith[4]

[4] Hawe, Seibert and Kleinsteuber, "Separable Dictionary Learning", CVPR, 2013

2D-MOD

Using Alternating Minimization:

1. **Update $\mathbf{X}_i$'s:** Use usual 2D sparse Rep. methods

$$\mathbf{X}_i^{(k+1)} = \underset{\mathbf{X}_i \in \mathcal{X}_i}{\operatorname{argmin}} \sum_{i=1}^{L} \left\| \mathbf{Y}_i \quad \mathbf{A}\mathbf{X}_i\mathbf{B}^T \right\|_F^2$$

2. **Update $\mathbf{A}$:** Use Gradient Projection

$$\text{normalize}\left\{ \left(\sum_{i=1}^{L} \mathbf{Y}_i \mathbf{B} \mathbf{X}_i^T \right) \left(\sum_{i=1}^{L} \mathbf{X}_i \mathbf{B}^T \mathbf{B} \mathbf{X}_i^T \right)^{-1} \right\} \tag{2}$$

3. **Update $\mathbf{B}$:** Use Gradient Projection

$$\text{normalize}\left\{ \left(\sum_{i=1}^{L} \mathbf{Y}_i^T \mathbf{A} \mathbf{X}_i \right) \left(\sum_{i=1}^{L} \mathbf{X}_i^T \mathbf{A}^T \mathbf{A} \mathbf{X}_i \right)^{-1} \right\} \tag{3}$$

2D-CMOD Idea

Convexification Idea[5]

$$\begin{cases} \mathbf{A} = \mathbf{A}_a + \mathbf{A} \quad \mathbf{A}_a \\ \mathbf{B} = \mathbf{B}_a + \mathbf{B} \quad \mathbf{B}_a \\ \mathbf{X} = \mathbf{X}_a + \mathbf{X} \quad \mathbf{X}_a \end{cases}$$

$$\mathbf{AXB}^T = \mathbf{A}_a\mathbf{X}_a\mathbf{B}^T + \mathbf{AX}_a\mathbf{B}_a^T + \mathbf{A}_a\mathbf{XB}_a^T \quad 2\mathbf{A}_a\mathbf{X}_a\mathbf{B}_a^T +$$
$$\mathbf{A}_a(\mathbf{X} \quad \mathbf{X}_a)(\mathbf{B} \quad \mathbf{B}_a)^T + (\mathbf{A} \quad \mathbf{A}_a)\mathbf{X}_a(\mathbf{B} \quad \mathbf{B}_a)^T +$$
$$(\mathbf{A} \quad \mathbf{A}_a)(\mathbf{X} \quad \mathbf{X}_a)\mathbf{B}_a^T + (\mathbf{A} \quad \mathbf{A}_a)(\mathbf{X} \quad \mathbf{X}_a)(\mathbf{B} \quad \mathbf{B}_a)^T$$

$$\boxed{\mathbf{AXB}^T \quad \mathbf{A}_a\mathbf{X}_a\mathbf{B}^T + \mathbf{AX}_a\mathbf{B}_a^T + \mathbf{A}_a\mathbf{XB}_a^T \quad 2\mathbf{A}_a\mathbf{X}_a\mathbf{B}_a^T}$$

[5] Sadeghi, Babaie-Zadeh and Jutten, \Dictionary learning for sparse representation: A novel approach", SPL, 2013

2D-CMOD Problem

New Cost function for 2D Dictionary Learning

$$(\mathbf{A}^*, \mathbf{X}^*, \mathbf{B}^*) = \underset{\mathbf{X}_i \in \mathcal{X}_i, \mathbf{A} \in \mathcal{A}, \mathbf{B} \in \mathcal{B}}{\operatorname{argmin}} \sum_{i=1}^{L} k\mathbf{Y}_i + 2\mathbf{A}_a\mathbf{X}_{a,i}\mathbf{B}_a^T$$

$$\mathbf{A}_a\mathbf{X}_{a,i}\mathbf{B}^T \quad \mathbf{A}\mathbf{X}_{a,i}\mathbf{B}_a^T \quad \mathbf{A}_a\mathbf{X}_i\mathbf{B}_a^T k_F^2$$

- **Jointly Convex** over $\mathbf{A}, \mathbf{B}$ and $\mathbf{X}_i$'s
- $\mathbf{A}_a, \mathbf{B}_a$ and $\mathbf{X}_{a,i}$ are parameters (previous values of variables)
- Different approaches exist to choose these parameters[6]

[6] Parsa, Sadeghi, Babaie-Zadeh and Jutten, \A new algorithm for dictionary learning based on convex approximation", EUSIPCO, 2019

2D-CMOD Algorithm (cntd.)

Using Alternating Minimization:

1. **Update $\mathbf{X}_i$'s**: Use usual 2D sparse representation methods for each $\mathbf{Z}_i$

$$\begin{cases} \mathbf{A}_a = \mathbf{A}^{(k-1)}, \mathbf{A} = \mathbf{A}^{(k)} \\ \mathbf{B}_a = \mathbf{B}^{(k-1)}, \mathbf{B} = \mathbf{B}^{(k)} \\ \mathbf{X}_a = \mathbf{X}^{(k)} \\ \mathbf{Z}_i = \mathbf{Y}_i \quad (\mathbf{A}^{(k)} \quad \mathbf{A}^{(k-1)})\mathbf{X}_i^{(k)}(\mathbf{B}^{(k-1)})^T \\ \qquad \mathbf{A}^{(k-1)}\mathbf{X}_i^{(k)}(\mathbf{B}^{(k)} \quad \mathbf{B}^{(k-1)})^T \end{cases}$$

$$\mathbf{X}_i^{(k+1)} = \underset{\mathbf{X}_i \in \mathcal{X}}{\operatorname{argmin}} \sum_{i=1}^{L} k\mathbf{Z}_i \quad \mathbf{A}^{(k-1)}\mathbf{X}_i(\mathbf{B}^{(k-1)})^T k_F^2$$

2D-CMOD Algorithm

Using Alternating Minimization:

2. **Update $\mathbf{A}$**

$$\begin{cases} \mathbf{X}_a = \mathbf{X} = \mathbf{X}^{(k+1)} \\ \mathbf{B}_a = \mathbf{B} = \mathbf{B}^{(k)} \end{cases}$$

- The same problem as 2D-MOD for updating $\mathbf{A}$, use (2)

3. **Update $\mathbf{B}$**

$$\begin{cases} \mathbf{X}_a = \mathbf{X} = \mathbf{X}^{(k+1)} \\ \mathbf{A}_a = \mathbf{A} = \mathbf{A}^{(k+1)} \end{cases}$$

- The same problem as 2D-MOD for updating $\mathbf{B}$, use (3)

2D-CMOD Pseudo-Code

Algorithm 1: 2D-CMOD

Input: Signal set: $\mathcal{Y}$, Sparsity level: s, Number of training signals: num_train, Algorithm iterations: $iter$.

Output: Sparse representations: $\mathbf{X}_i$'s, Dictionaries: $\mathbf{A}$ and $\mathbf{B}$.

Initialize dictionaries $\mathbf{A}$ and $\mathbf{B}$.

Set: $\mathbf{A}^{(0)} = \mathbf{A}^{(-1)} = \mathbf{A}, \mathbf{B}^{(0)} = \mathbf{B}^{(-1)} = \mathbf{B}$.

for $k = 0$ **to** $iter - 1$ **do**

  for $i = 1$ **to** num_train **do**

    $\mathbf{Z}_i = \mathbf{Y}_i - (\mathbf{A}^{(k)} - \mathbf{A}^{(k-1)})\mathbf{X}_i(\mathbf{B}^{(k-1)})^T - \mathbf{A}^{(k-1)}\mathbf{X}_i(\mathbf{B}^{(k)} - \mathbf{B}^{(k-1)})^T$

    $\mathbf{X}_i =$ Sparse Coding$(\mathbf{Z}_i, \mathbf{A}^{(k)}, \mathbf{B}^{(k)}, s)$

  end for

  $\mathbf{A}^{(k+1)}$ = Update dictionary $\mathbf{A}$ as in (2).

  $\mathbf{B}^{(k+1)}$ = Update dictionary $\mathbf{B}$ as in (3).

end for

Recovery of Known Dictionary

Generating synthetic data

* Assume $\mathbf{Y} \in \mathbb{R}^{n\times n}$

1. $\mathbf{A} \in \mathbb{R}^{n\times 2n}, \mathbf{B} \in \mathbb{R}^{n\times 2n} \rightarrow \mathcal{N}(0,1)$
2. $\mathbf{X}_i$'s are generated randomly with s non-zero elements
3. $\mathbf{Y}_i = \mathbf{A}\mathbf{X}_i\mathbf{B}^T + \mathbf{N}_i$

Metrics

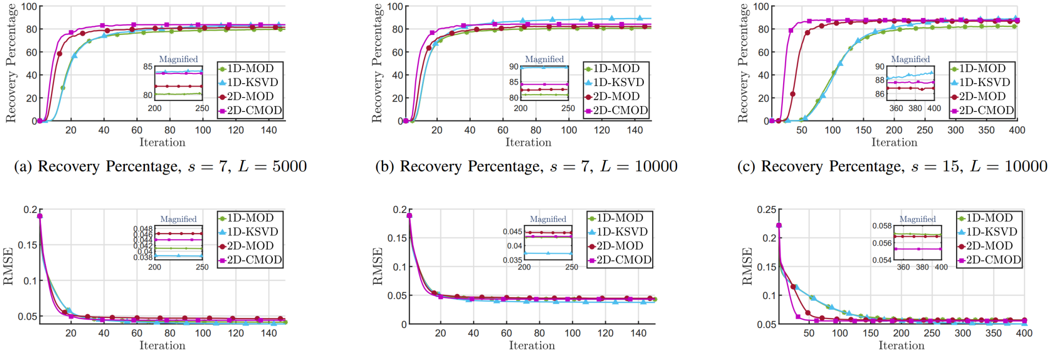
1. Successful Recovery Percentage of the Kronecker Dictionary.
$$\max(\mathbf{d}_i^T\mathbf{D}_t(:,j) > 0.99$$
2. Root Mean Square Error defined as:
$$\text{RMSE} = \sqrt{\sum_{i=1}^{L} \|\mathbf{Y}_i - \mathbf{A}\mathbf{X}_i\mathbf{B}^T\|_F^2 \Big/ n^2 L}$$

Some Details

- For all algorithms, Orthogonal Matching Pursuit (OMP)[7] has been used as the sparse coding algorithm
- All the simulations were performed in MATLAB 2018b environment on a system with 4.0 GHz CPU, and 16 GB RAM, under Microsoft Windows 10 64-bit operating system

[7] Troop and Gilbert, \Signal recovery from random measurements via orthogonal matching pursuit", TIT, 2007

Recovery of Known Dictionary



(a) Recovery Percentage, $s = 7$, $L = 5000$ (b) Recovery Percentage, $s = 7$, $L = 10000$ (c) Recovery Percentage, $s = 15$, $L = 10000$

(d) RMSE, $s = 7$, $L = 5000$ (e) RMSE, $s = 7$, $L = 10000$ (f) RMSE, $s = 15$, $L = 10000$

Figure 1: Successful Recovery Percentage and RMSE.

Recovery of Known Dictionary

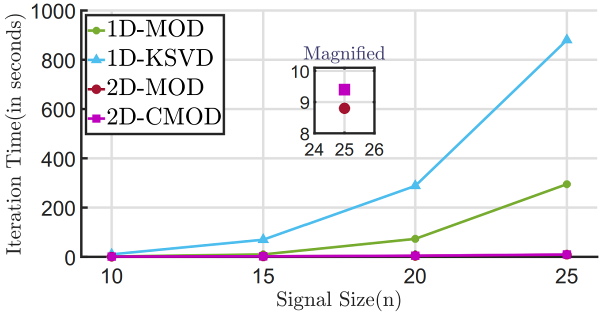


Figure 2: Average time each algorithm's iteration. $s = n$, $L = 1000n$.

Recovery of Known Dictionary

Table 1: Average Number of iterations and required times to achieve 80 percent recovery(times in seconds, reported between braces). sparsity level $s = n$, and $L = 1000n$.

Signals size	$n = 10$	$n = 15$	$n = 20$	$n = 25$
1D-MOD	62(90)	59(584)	70(5110)	—
1D-KSVD	52(527)	48(3339)	65(18720)	—
2D-MOD	59(47)	36(72)	34(146)	40(352)
2D-CMOD	**24(20)**	**23(49)**	**28(129)**	**25(235)**

Image Denoising[8]

- 40000 patches, size 12×12
- $\mathbf{A} \in \mathbb{R}^{12\times 24}$, $\mathbf{B} \in \mathbb{R}^{12\times 24}$, $\mathbf{D} \in \mathbb{R}^{144\times 576}$

Images	boat				house				Total Time
σ_{noise} (PSNR(dB))	10(28.12)	20(22.12)	30(18.61)	50(14.13)	10(28.18)	20(22.12)	30(18.60)	50(14.15)	
ODCT (Not Trained)	33.24	29.47	27.33	24.92	35.19	31.86	29.43	27.13	13
2D-MOD	33.33	29.70	27.60	25.19	35.22	32.16	29.76	27.47	524
2D-CMOD	33.26	29.59	27.57	25.17	35.03	31.98	29.69	27.44	636
SeDiL	31.14	27.20	25.20	23.47	32.91	29.00	26.39	24.32	573
KSVD	33.47	30.04	27.93	25.47	35.98	33.36	31.33	28.60	3130

[8] Elad and Aharon, \Image denoising via sparse and redundant representations over learned dictionaries", TIP, 2006

Conclusion

1. A new jointly convex objective function was introduced for 2D DL problem.
2. Two new algorithms were proposed to solve the 2D DL problem.
3. Experimental results show that the proposed methods have much less computational complexity than 1D methods. Moreover, they need fewer training signals and fewer iterations to converge.

Thank you for Your Attention!

Any Questions?