Sparse Discriminative Tensor Dictionary Learning for Object Classification

Seyyid Emre Sofuoglu, Selin Aviyente

Department of Electrical and Computer Engineering,
Michigan State University

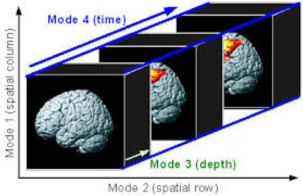
November 27, 2018

IEEE GlobalSIP

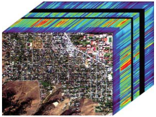
Overview

Introduction

- Higher-order datasets are encountered in
  - Computer vision: Grey level images and image sequences.
  - Neuroimaging: Electroencephalogram (EEG) recordings or functional magnetic resonance imaging (fMRI) data.



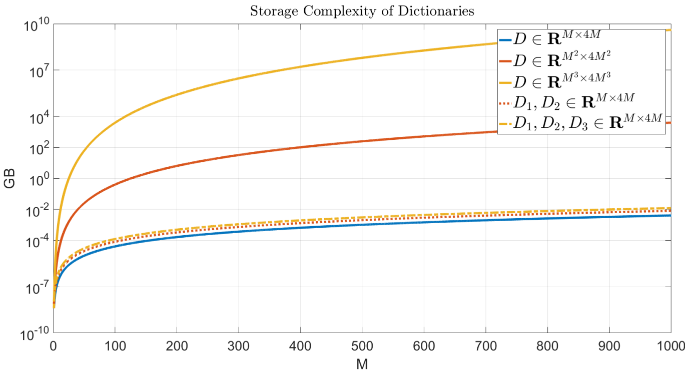
  - Remote sensing: Hyperspectral images or videos.



- To be able to classify these data, supervised tensor learning approaches are necessary.
- Some existing works are MPCA-LDA, DATER, DGTDA, CMDA.

Dictionary Learning (DL)

- DL has proven to be effective for many applications: Data recovery, denoising and compression
- Complexity of the model increases exponentially with increasing dimensionality.
- DL has also been employed in supervised learning:
    - Discriminative K-SVD [Zhang & Li, 2010], Label Consistent K-SVD [Jiang et.al., 2013], Sparse Representation Based Classification (SRC) [Wright et.al., 2009], etc.



Memory requirements in order to store overcomplete dictionaries for 1D, 2D, and 3D signals (tensors)(N=1, 2, 3). Here M is defined as the size of each mode of input signal.

- Tensor dictionary learning models are proposed for various problems but discriminative approaches were not deeply pursued.

Tensor Dictionary Learning

- Tensor extensions to well-known DL methods: T-MOD and K-HOSVD [Roemer et. al., 2014].
  $\Rightarrow$ Not efficient
- Discriminative TDL: [Zubair et. al., 2014] and [Wu et. al., 2017]
  $\Rightarrow$ Do not enforce sparsity of representations
  $\Rightarrow$ Learn overcomplete dictionaries
- Orthogonal TDL: [Quan et. al., 2015]
  $\Rightarrow$ Decrease computational complexity
  $\Rightarrow$ Not a supervised approach
- Propose an efficient orthogonal and discriminative DL method with sparsity constraints.

Objective

- Given a set of observed tensors $\mathcal{Y}_c^k \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ where $c \in \{1, \ldots, C\}$ denotes the class label and $k \in \{1, ..., K_c\}$ denotes the sample index for each class c, define $\mathcal{Y}_c \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times K_c}$ as the collection of all $\mathcal{Y}_c^k$.
- We aim to learn dictionaries for each class and mode, $D_c^{(n)} \in \mathbb{R}^{I_n \times I_n}$ such that ${D_c^{(n)}}^\top D_c^{(n)} = I$ and the dictionaries provide:
  1. Discriminability: Minimize within-class scatter:

$$S_w^{(n)} = \sum_{c=1}^{C} \sum_{k=1}^{K_c} \left[(\mathcal{Y}_c^k - \mathcal{M}_c) \prod_{\substack{m \in \{1,2,3\} \\ m \neq n}} \times_m {D_c^{(m)}}^\top \right]_{(n)} \left[(\mathcal{Y}_c^k - \mathcal{M}_c) \prod_{\substack{m \in \{1,2,3\} \\ m \neq n}} \times_m {D_c^{(m)}}^\top \right]_{(n)}^\top .$$

Objective

- Given a set of observed tensors $\mathcal{Y}_c^k \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ where $c \in \{1, \ldots, C\}$ denotes the class label and $k \in \{1, ..., K_c\}$ denotes the sample index for each class c, define $\mathcal{Y}_c \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times K_c}$ as the collection of all $\mathcal{Y}_c^k$.
- We aim to learn dictionaries for each class and mode, $D_c^{(n)} \in \mathbb{R}^{I_n \times I_n}$ such that $D_c^{(n)\top} D_c^{(n)} = I$ and the dictionaries provide:
  1. Discriminability: Minimize within-class scatter:

$$S_w^{(n)} = \sum_{c=1}^{C} \sum_{k=1}^{K_c} \left[(\mathcal{Y}_c^k - \mathcal{M}_c) \prod_{\substack{m \in \{1,2,3\} \\ m \neq n}} \times_m D_c^{(m)\top} \right]_{(n)} \left[(\mathcal{Y}_c^k - \mathcal{M}_c) \prod_{\substack{m \in \{1,2,3\} \\ m \neq n}} \times_m D_c^{(m)\top} \right]_{(n)}^{\top}.$$

  2. Sparsity: Minimize reconstruction error with a constraint on the sparsity of the projection:

$$\|\mathcal{Y}_c^k - \mathcal{X}_c^k \prod_{n=1}^{3} \times_n D_c^{(n)}\|_F^2 \quad \text{is equivalent to} \quad \|\mathcal{Y}_c^k \prod_{n=1}^{3} \times_n D_c^{(n)\top} - \mathcal{X}_c^k\|_F^2,$$

where $\mathcal{X}_c^k \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ are the sparse representations.

Sparse Disriminant Tensor Dictionary Learning (SDTDL)

- Combining reconstruction and discrimination terms, the optimization function with constraints is:

$$\underset{\mathcal{X}_c, D_c^{(1)}, D_c^{(2)}, D_c^{(3)}}{\operatorname{argmin}} \sum_{k=1}^{K_c} \|\mathcal{Y}_c^k - \mathcal{X}_c^k \prod_{n=1}^{3} \times_n D_c^{(n)}\|_F^2 + \lambda \sum_{n=1}^{3} tr(D_c^{(n)\top} S_w^{(n)} D_c^{(n)}) \quad , \|\mathcal{X}_c^k\|_0 \leq \tau \quad , \forall c, k, \quad (1)$$

  where $\mathcal{X}_c \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times K_c}$ is the collection of all $\mathcal{X}_c^k$.
- Non-convex problem: However, it can be solved for each class and each variable, separately.

Sparse Disriminant Tensor Dictionary Learning (SDTDL)

- Combining reconstruction and discrimination terms, the optimization function with constraints is:

$$\underset{\mathcal{X}_c, D_c^{(1)}, D_c^{(2)}, D_c^{(3)}}{\operatorname{argmin}} \sum_{k=1}^{K_c} \|\mathcal{Y}_c^k - \mathcal{X}_c^k \prod_{n=1}^{3} \times_n D_c^{(n)}\|_F^2 + \lambda \sum_{n=1}^{3} tr({D_c^{(n)}}^\top S_w^{(n)} D_c^{(n)}) \quad , \|\mathcal{X}_c^k\|_0 \leq \tau \quad , \forall c, k, \quad (1)$$

where $\mathcal{X}_c \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times K_c}$ is the collection of all $\mathcal{X}_c^k$.
- Non-convex problem: However, it can be solved for each class and each variable, separately.

1. Initialize dictionaries $D_c^{(i)}$.
2. Update sparse representations, $\mathcal{X}_c^k$. Keep τ highest values of $\mathcal{Y}_c^k \prod_{n=1}^3 \times_n {D_c^{(n)}}^\top$ and set the rest to zero.
3. Update dictionary $D_c^{(n)}$ while fixing all other dictionaries $D_c^{(m)}$ and $\mathcal{X}_c$.

Optimization

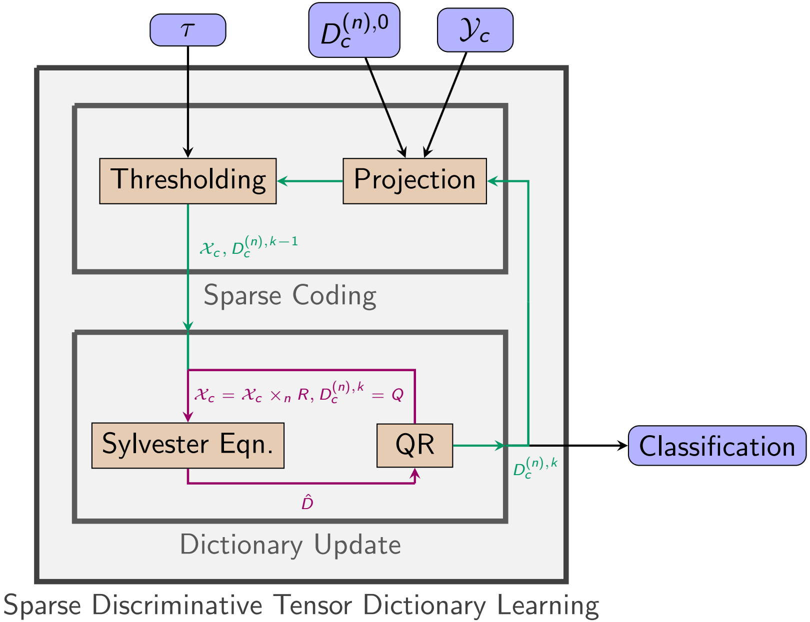
- Updating $D_c^{(n)}$: Since all other variables are fixed, taking the derivative of reconstruction and scatter terms is enough.

$$D_c^{(n)}\mathcal{C}_{(n)}\mathcal{C}_{(n)}^\top + \lambda S_w^{(n)} D_c^{(n)} = Y_{c_{(n)}}\mathcal{C}_{(n)}^\top, \tag{2}$$

where

$$\mathcal{C}_{(n)} = (\mathcal{X}_c \prod_{\substack{m\in\{1,2,3\}\\ m\neq n}} \times_m D_c^{(m)\top})_{(n)}.$$

- (2) is a Sylvester equation that is easily solved, but does not guarantee orthogonality.
- Thus, we apply QR decomposition to the solution of (2) $\hat{D} = QR$ and set the dictionary as $D_c^{(n)} = Q$.
- Then, the sparse representation is updated for the same mode as $\mathcal{X}_c \leftarrow \mathcal{X}_c \times_n R$.


$\mathcal{T}$
$D_c^{(n),0}$
$\mathcal{Y}_c$
Thresholding
Projection
$\mathcal{X}_c, D_c^{(n),k-1}$
Sparse Coding
$\mathcal{X}_c = \mathcal{X}_c \times_n R, D_c^{(n),k} = Q$
Sylvester Eqn.
QR
$\hat{D}$
$D_c^{(n),k}$
Classification
Dictionary Update
Sparse Discriminative Tensor Dictionary Learning

Classification

- To classify a test sample $\mathcal{Y}^t$, we find sparse representations $\mathcal{X}_c^t$ with respect to each class using learned dictionaries D_c.
- C different sparse representations for each test sample.
- Find the class which has the closest sparse representation:

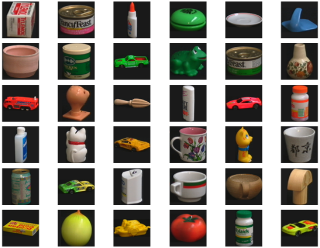
$$l = \underset{c}{\operatorname{argmin}}\{\min_k(\|\mathcal{X}_c^k - \mathcal{X}_c^t\|_F^2)\}, \tag{3}$$

Computational Complexity

- Assume $I_1 = I_2 = \cdots = I_N = I$, $K_c = K, \forall c$, $c \in \{1, \ldots, C\}$. Given inputs $\mathcal{Y}_c^k \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the training cost per iteration:
  1. Computational cost of extracting $\mathcal{X}_c$: $\mathcal{O}(NKI^{N+1})$ per class.
  2. Computational cost of learning $D_c^{(n)}$: $\mathcal{O}((CN + N + 1)NKI^{N+1})$, for all classes and modes.
  3. Total computational cost is $\mathcal{O}\left((CN + C + N + 1)NKI^{N+1}\right)$.
- The computational cost of MPCA is $\mathcal{O}\left((N + 1)CKNI^{N+1}\right)$.
- The cost is governed by Tensor-to-Matrix multiplications.
- The storage cost of SDTDL is $\mathcal{O}(NCI^2 + 2CK\tau)$, while the storage cost of MPCA (assuming all ranks are $\tau^{1/N}$) is $\mathcal{O}(NI\tau^{1/N} + CK\tau)$.

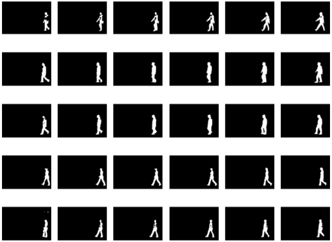
Datasets

- **COIL-100**: 100 objects with size 128×128, 72 images corresponding to pose angles. 16 non-overlapping patches of size 16×16: $\Rightarrow \mathcal{Y}_c^k \in \mathbb{R}^{16\times16\times16}$



Sample images from COIL-100 Dataset.

- **CASIA Gait Dataset A**: 4 sequences, 3 directions, 20 subjects. Image size 240×352. From sequences of size $60 \times 88 \times 50$, 200 patches of size $15 \times 20 \times 16$: $\Rightarrow \mathcal{Y}_c^k \in \mathbb{R}^{15\times20\times16\times200}$



Sample sequences from CASIA-GAIT Dataset.

Experiments

- Hold-out ratio is $r = N_{trn}/N_{ttl}\%$ where N_{trn} is the number of training samples and N_{ttl} is the total number of samples.
- The sparsity level, σ, is the ratio of the number of nonzero terms to total number of elements in each sample.

Comparison of Accuracy of Algorithms on COIL-100 Dataset.

Algorithms	Hold-out ratios (%)			
10 Class ($\sigma = 0.031$)	1.4	2.8	25	50
SDTDL	**70.21** ± 6.51	**76.11** ± 4.17	**97.00** ± 1.82	**99.19** ± 0.50
MPCA	67.86 ± 5.76	74.70 ± 3.39	96.78 ± 2.37	98.92 ± 0.79
DGTDA		74.3 ± 3.06	95.6 ± 1.5	99 ± 0.604
CMDA		70.9 ± 3.72	94.2 ± 1.72	98.4 ± 0.541
20 Class ($\sigma = 0.047$)	1.4	2.8	25	50
SDTDL	**60.51** ± 3.45	**72.61** ± 3.35	**95.91** ± 2.02	**99.22** ± 0.75
MPCA	59.24 ± 4.59	69.54 ± 3.53	93.75 ± 3.53	98.17 ± 1.20
DGTDA		71.8 ± 2.07	95.8 ± 0.642	98.9 ± 0.219
CMDA		68.9 ± 2.08	95.1 ± 0.728	98.6 ± 0.31

Comparison of Accuracy of Algorithms on CASIA Gait Dataset A. ($\sigma = 0.01$)

%	Hold-out ratios (%)		
Algorithms	12.5	25	50
SDTDL	**92.48** ± 5.36	**96.78** ± 2.54	98.67 ± 1.53
MPCA	90.76 ± 5.22	95.89 ± 1.58	98.17 ± 1.66
DGTDA	76.3 ± 3.5	96.4 ± 0.945	**99.4** ± 0.435
CMDA	69.5 ± 4.45	94.8 ± 1.13	99.2 ± 0.48

Discussion

- **Effect of Discriminability** ($r = 1.4\%, \sigma = 0.031$, COIL 10 classes):
  67.55 ± 4.89 vs $\mathbf{69.35 \pm 5.85}$.
- **Effect of Sparsity**: Increasing the number of non-zero elements lowers the accuracy.
- SDTDL outperforms other methods especially when the number of training samples is small.

Conclusions

- Proposed a sparse discriminative tensor dictionary learning algorithm for tensor-type object classification.
- Combination of reconstruction error and discrimination power to learn orthogonal and separable dictionaries for each class.
- Orthogonality and separability make the training efficient compared to learning overcomplete dictionaries.
- Higher classification accuracy compared to MPCA and MDA.

Thanks for listening!