Locally Linear Low-rank Tensor Approximation

Alp Ozdemir[1], Mark A. Iwen[1,2] and Selin Aviyente[1]

[1] Department of Electrical and Computer Engineering, Michigan State University
[2] Deparment of the Mathematics, Michigan State University

December 15, 2015

Outline

Introduction

- High dimensional data sets in $\mathbb{R}^D$ lie in a lower d-dimensional manifold, or subspace, with $d \ll D$.

Introduction

- High dimensional data sets in $\mathbb{R}^D$ lie in a lower d-dimensional manifold, or subspace, with $d \ll D$.
- Linear subspace estimation methods: PCA, SVD.

Introduction

- High dimensional data sets in $\mathbb{R}^D$ lie in a lower d-dimensional manifold, or subspace, with $d \ll D$.
- Linear subspace estimation methods: PCA, SVD.
- Manifold Learning Methods:
  - Multidimensional Scaling (MDS): Embed the data into a graph to construct d-dimensional manifold (Tenenbaum et al. 2000).
  - Locally linear embedding (LLE)(Roweis and Saul 2000).
  - Geometric Multi-resolution Analysis (GMRA): Data dependent multi-scale dictionaries (Allard et al. 2012).

Introduction

- High dimensional data sets in $\mathbb{R}^D$ lie in a lower d-dimensional manifold, or subspace, with $d \ll D$.
- Linear subspace estimation methods: PCA, SVD.
- Manifold Learning Methods:
  - Multidimensional Scaling (MDS): Embed the data into a graph to construct d-dimensional manifold (Tenenbaum et al. 2000).
  - Locally linear embedding (LLE)(Roweis and Saul 2000).
  - Geometric Multi-resolution Analysis (GMRA): Data dependent multi-scale dictionaries (Allard et al. 2012).
- However, these approaches are not directly applicable to high order data, e.g. hyperspectral imaging, social and biological networks.

Introduction

- Linear Low-Rank Structure Learning:

Introduction

- Linear Low-Rank Structure Learning:
  - Higher order singular value decomposition (HOSVD)
  - Parallel Factor Analysis (PARAFAC)

Introduction

- Linear Low-Rank Structure Learning:
  - Higher order singular value decomposition (HOSVD)
  - Parallel Factor Analysis (PARAFAC)
- Manifold Learning Methods for Tensors:

Introduction

- Linear Low-Rank Structure Learning:
  - Higher order singular value decomposition (HOSVD)
  - Parallel Factor Analysis (PARAFAC)
- Manifold Learning Methods for Tensors:
  - He et al.(2005) extended locality preserving projections to second order tensors.

Introduction

- Linear Low-Rank Structure Learning:
  - Higher order singular value decomposition (HOSVD)
  - Parallel Factor Analysis (PARAFAC)
- Manifold Learning Methods for Tensors:
  - He et al.(2005) extended locality preserving projections to second order tensors.
  - Dai and Yeung (2006) extended following embedding methods to tensors:

Introduction

- Linear Low-Rank Structure Learning:
  - Higher order singular value decomposition (HOSVD)
  - Parallel Factor Analysis (PARAFAC)
- Manifold Learning Methods for Tensors:
  - He et al.(2005) extended locality preserving projections to second order tensors.
  - Dai and Yeung (2006) extended following embedding methods to tensors:
    - Local discriminant embedding
    - Neighborhood preserving embedding
    - Locality preserving projection

MICHIGAN STATE UNIVERSITY

Introduction

- Linear Low-Rank Structure Learning:
  - Higher order singular value decomposition (HOSVD)
  - Parallel Factor Analysis (PARAFAC)
- Manifold Learning Methods for Tensors:
  - He et al.(2005) extended locality preserving projections to second order tensors.
  - Dai and Yeung (2006) extended following embedding methods to tensors:
    - Local discriminant embedding
    - Neighborhood preserving embedding
    - Locality preserving projection
- These methods are mostly limited to learning the optimal linear transformation for supervised classification of high-order data.

Introduction

- Goal: Propose a unsupervised data structure learning method to exploit locally linear low-rank structure of high-order data.

Introduction

- Goal: Propose a unsupervised data structure learning method to exploit locally linear low-rank structure of high-order data.
- Two step approach:

Introduction

- Goal: Propose a unsupervised data structure learning method to exploit locally linear low-rank structure of high-order data.
- Two step approach:
  - Decompose the tensor into subtensors.

Introduction

- Goal: Propose a unsupervised data structure learning method to exploit locally linear low-rank structure of high-order data.
- Two step approach:
  - Decompose the tensor into subtensors.
  - Apply higher order singular value decomposition (HOSVD) to these subtensors.

Outline

Tensor Algebra

- A multidimensional array with N modes $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is called a tensor, where $x_{i_1, i_2, ..i_N}$ denotes the $(i_1, i_2, ..i_N)^{th}$ element of the tensor $\mathcal{X}$.

Tensor Algebra

- A multidimensional array with N modes $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is called a tensor, where $x_{i_1, i_2, ..i_N}$ denotes the $(i_1, i_2, ..i_N)^{th}$ element of the tensor $\mathcal{X}$.
- Vectors obtained by fixing all indices of the tensor except the one that corresponds to nth mode are called mode-n fibers.

Tensor Algebra

- A multidimensional array with N modes $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is called a tensor, where $x_{i_1, i_2, ..i_N}$ denotes the $(i_1, i_2, ..i_N)^{th}$ element of the tensor $\mathcal{X}$.
- Vectors obtained by fixing all indices of the tensor except the one that corresponds to nth mode are called mode-n fibers.
- Tensor matricization: The mode-n matricization of tensor $\mathcal{Y}$ is denoted as $\mathbf{Y}_{(n)}$ and is obtained by arranging mode-n fibers to be the columns of the resulting matrix.

Tensor Algebra

- A multidimensional array with N modes $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is called a tensor, where $x_{i_1, i_2, ..i_N}$ denotes the $(i_1, i_2, ..i_N)^{th}$ element of the tensor $\mathcal{X}$.
- Vectors obtained by fixing all indices of the tensor except the one that corresponds to nth mode are called mode-n fibers.
- Tensor matricization: The mode-n matricization of tensor $\mathcal{Y}$ is denoted as $\mathbf{Y}_{(n)}$ and is obtained by arranging mode-n fibers to be the columns of the resulting matrix.
- Mode-n product: $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U}$ yields $\mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}$.

Tensor Algebra

- A multidimensional array with N modes $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is called a tensor, where $x_{i_1, i_2, ..i_N}$ denotes the $(i_1, i_2, ..i_N)^{th}$ element of the tensor $\mathcal{X}$.
- Vectors obtained by fixing all indices of the tensor except the one that corresponds to nth mode are called mode-n fibers.
- Tensor matricization: The mode-n matricization of tensor $\mathcal{Y}$ is denoted as $\mathbf{Y}_{(n)}$ and is obtained by arranging mode-n fibers to be the columns of the resulting matrix.
- Mode-n product: $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U}$ yields $\mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}$.
- Tensor *n-rank* of $\mathcal{X}$ is the collection of ranks of mode matrices $\mathbf{X}_{(n)}$:
  $n\text{-rank}(\mathcal{X}) = \left(\text{rank}(\mathbf{X}_{(1)}),\ \text{rank}(\mathbf{X}_{(2)}), ...,\ \text{rank}(\mathbf{X}_{(N)})\right)$.

Higher-Order Singular Value Decomposition (HOSVD)

Any tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ can be decomposed as:

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_N \mathbf{U}^{(N)}, \tag{1}$$

Higher-Order Singular Value Decomposition (HOSVD)

Any tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ can be decomposed as:

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_N \mathbf{U}^{(N)}, \tag{1}$$

where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$s are the left singular vectors of $\mathbf{X}_{(n)}$ and $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is the core tensor computed as:

$$\mathcal{S} = \mathcal{X} \times_1 (\mathbf{U}^{(1)})^\top \times_2 (\mathbf{U}^{(2)})^\top \ldots \times_N (\mathbf{U}^{(N)})^\top. \tag{2}$$

Outline

Identifying Subtensors

Direct Division:

Identifying Subtensors

Direct Division:

- Unfold tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ across each mode to obtain $\mathbf{X}_n \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ whose columns are the mode-n fibers of $\mathcal{X}$.

Identifying Subtensors

Direct Division:

- Unfold tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ across each mode to obtain $\mathbf{X}_n \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ whose columns are the mode-n fibers of $\mathcal{X}$.
- For each mode, the mode-n fibers are partitioned into c_n non-overlapping clusters through METIS (Karypis and Kumar 1998).

Identifying Subtensors

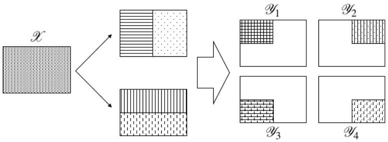
Direct Division:

- Unfold tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ across each mode to obtain $\mathbf{X}_n \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ whose columns are the mode-n fibers of $\mathcal{X}$.
- For each mode, the mode-n fibers are partitioned into c_n non-overlapping clusters through METIS (Karypis and Kumar 1998).
- Cartesian product of the fiber labels coming from different modes yields $K = \prod_{i=1}^{N} c_n$ subtensors $\mathcal{Y}_k$.

Identifying Subtensors

Direct Division:

- Unfold tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ across each mode to obtain $\mathbf{X}_n \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ whose columns are the mode-n fibers of $\mathcal{X}$.
- For each mode, the mode-n fibers are partitioned into c_n non-overlapping clusters through METIS (Karypis and Kumar 1998).
- Cartesian product of the fiber labels coming from different modes yields $K = \prod_{i=1}^{N} c_n$ subtensors $\mathcal{Y}_k$.



Identifying Subtensors

Sequential Division:

Identifying Subtensors

Sequential Division:

- Mode-1 fibers of $\mathcal{X}$ are grouped into c_1 clusters to obtain c_1 subtensors.

Identifying Subtensors

Sequential Division:

- Mode-1 fibers of $\mathcal{X}$ are grouped into c_1 clusters to obtain c_1 subtensors.
- Mode-2 fibers of each of the newly created subtensors are clustered into c_2 clusters separately which yields $c_1 \times c_2$ subtensors.

Identifying Subtensors

Sequential Division:

- Mode-1 fibers of $\mathcal{X}$ are grouped into c_1 clusters to obtain c_1 subtensors.
- Mode-2 fibers of each of the newly created subtensors are clustered into c_2 clusters separately which yields $c_1 \times c_2$ subtensors.
- This procedure is applied N times by clustering the fibers of different modes at each step and $K = \prod_{i=1}^{N} c_n$ subtensors are obtained.

Identifying Subtensors

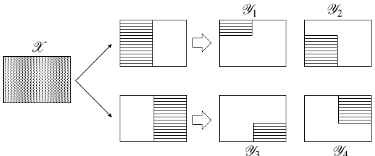
Sequential Division:

- Mode-1 fibers of $\mathcal{X}$ are grouped into c_1 clusters to obtain c_1 subtensors.
- Mode-2 fibers of each of the newly created subtensors are clustered into c_2 clusters separately which yields $c_1 \times c_2$ subtensors.
- This procedure is applied N times by clustering the fibers of different modes at each step and $K = \prod_{i=1}^{N} c_n$ subtensors are obtained.
- Choosing different ordering of the modes yields $N!$ different possible decompositions, $\mathcal{Y}_k$.

Identifying Subtensors

Sequential Division:

- Mode-1 fibers of $\mathcal{X}$ are grouped into c_1 clusters to obtain c_1 subtensors.
- Mode-2 fibers of each of the newly created subtensors are clustered into c_2 clusters separately which yields $c_1 \times c_2$ subtensors.
- This procedure is applied N times by clustering the fibers of different modes at each step and $K = \prod_{i=1}^{N} c_n$ subtensors are obtained.
- Choosing different ordering of the modes yields $N!$ different possible decompositions, $\mathcal{Y}_k$.



Locally Linear High Order Singular Value Decomposition

Goal: Low *n*-rank approximation to subtensors of an *N*th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ to better capture local nonlinearities.

Locally Linear High Order Singular Value Decomposition

Goal: Low *n*-rank approximation to subtensors of an *N*th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ to better capture local nonlinearities.

- Decompose tensor $\mathcal{X}$ into K subtensors $\mathcal{Y}_k \in \mathbb{R}^{I_{1,k} \times I_{2,k} \times \ldots \times I_{N,k}}$ with $k \in \{1,\ 2, \ldots\ K\}$ by direct division or sequential division approaches.

Locally Linear High Order Singular Value Decomposition

Goal: Low *n*-rank approximation to subtensors of an *N*th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ to better capture local nonlinearities.

- Decompose tensor $\mathcal{X}$ into K subtensors $\mathcal{Y}_k \in \mathbb{R}^{I_{1,k} \times I_{2,k} \times \ldots \times I_{N,k}}$ with $k \in \{1, 2, \ldots K\}$ by direct division or sequential division approaches.
- Mapping functions f_ks are defined on the index sets from $\mathcal{X}$ to $\mathcal{Y}_k$ as:

$$f_k : J_1 \times J_2 \times \ \ldots \ \times J_N \mapsto J_{1,k} \times J_{2,k} \times \ \ldots \ \times J_{N,k}, \qquad (3)$$

where $J_n = \{1, 2, \ldots, I_n\}$, $J_{n,k} \subset \{1, 2, \ldots, I_{n,k}\}$ with $n \in \{1, 2, \ldots N\}$.

Locally Linear High Order Singular Value Decomposition

Goal: Low n-rank approximation to subtensors of an Nth order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ to better capture local nonlinearities.

- Decompose tensor $\mathcal{X}$ into K subtensors $\mathcal{Y}_k \in \mathbb{R}^{I_{1,k} \times I_{2,k} \times \ldots \times I_{N,k}}$ with $k \in \{1, 2, \ldots K\}$ by direct division or sequential division approaches.
- Mapping functions f_ks are defined on the index sets from $\mathcal{X}$ to $\mathcal{Y}_k$ as:

$$f_k : J_1 \times J_2 \times \ \ldots \ \times J_N \mapsto J_{1,k} \times J_{2,k} \times \ \ldots \ \times J_{N,k}, \qquad (3)$$

  where $J_n = \{1, 2, \ldots, I_n\}$, $J_{n,k} \subset \{1, 2, \ldots, I_{n,k}\}$ with $n \in \{1, 2, \ldots N\}$.
- f_ks satisfy $\cup_{k=1}^{K} J_{n,k} = J_n$ and $J_{n,k} \cap J_{n,l} = \emptyset$ when $k \neq l$ for all $k, l \in \{1, 2, \ldots, K\}$.

MICHIGAN STATE UNIVERSITY

Locally Linear High Order Singular Value Decomposition

- HOSVD is used to obtain the low *n*-rank approximation for each $\mathcal{Y}_k$.

Locally Linear High Order Singular Value Decomposition

- HOSVD is used to obtain the low n-rank approximation for each $\mathcal{Y}_k$.
- Let $\hat{\mathcal{Y}}_k$ be a low n-rank approximation of $\mathcal{Y}_k$ computed as:

$$\hat{\mathcal{Y}}_k = \hat{\mathcal{S}}_k \times_1 \hat{\mathbf{U}}^{(1,k)} \times_2 \hat{\mathbf{U}}^{(2,k)} \ldots \times_N \hat{\mathbf{U}}^{(N,k)}, \tag{4}$$

where $\hat{\mathbf{U}}^{(n,k)}$s are the truncated projection matrices of $\mathcal{Y}_k$ obtained by keeping the first r_n columns of $\mathbf{U}^{(n,k)}$ for $n \in \{1, 2, \ldots N\}$ and $\hat{\mathcal{S}}_k$ is the core tensor

$$\hat{\mathcal{S}}_k = \hat{\mathcal{Y}}_k \times_1 (\hat{\mathbf{U}}^{(1,k)})^\top \times_2 (\hat{\mathbf{U}}^{(2,k)})^\top \ldots \times_N (\hat{\mathbf{U}}^{(N,k)})^\top. \tag{5}$$

Locally Linear High Order Singular Value Decomposition

- $\hat{\mathcal{Y}}_k$s corresponds to:

$$\hat{\mathcal{Y}}_k = \hat{\mathcal{X}}_{f_k(J_1 \times J_2 \times \ldots \times J_N)}, \tag{6}$$

Locally Linear High Order Singular Value Decomposition

- $\hat{\mathcal{Y}}_k$s corresponds to:

$$\hat{\mathcal{Y}}_k = \hat{\mathcal{X}}_{f_k(J_1 \times J_2 \times \ldots \times J_N)}, \tag{6}$$

- Combining all of the subtensors $\hat{\mathcal{Y}}_k$s by using the inverse mapping functions f_k^{-1} provides piecewise-linear approximation of $\mathcal{X}$:

$$\hat{\mathcal{X}} = \sum_{k=1}^{K} \hat{\mathcal{Y}}_{k,(f_k^{-1}(J_{1,k} \times J_{2,k} \times \ldots \times J_{N,k}))}. \tag{7}$$

Outline

Simulated Datasets

- Translating Subspaces

Simulated Datasets

- Translating Subspaces
  - Two point clouds with 100 Gaussian random variables in $\mathbb{R}^{20}$ were generated.
  - The two subspaces in which the point clouds live are orthogonal to each other in $\mathbb{R}^{100}$.
  - The first point cloud is static whereas the second one is translating in time $t \in \{1, 2, \ldots 60\}$.
  - A 3-mode tensor $\mathcal{X} \in \mathbb{R}^{100 \times 200 \times 60}$ is created.

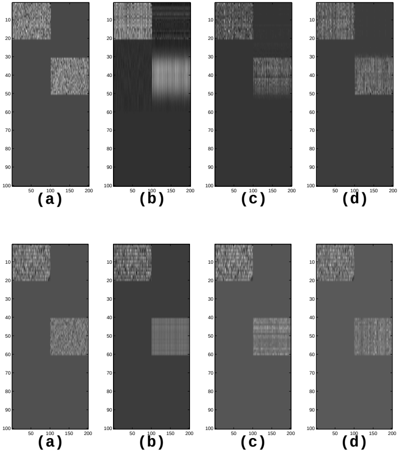
Simulated Datasets

- Translating Subspaces
  - Two point clouds with 100 Gaussian random variables in $\mathbb{R}^{20}$ were generated.
  - The two subspaces in which the point clouds live are orthogonal to each other in $\mathbb{R}^{100}$.
  - The first point cloud is static whereas the second one is translating in time $t \in \{1, 2, ...60\}$.
  - A 3-mode tensor $\mathcal{X} \in \mathbb{R}^{100\times200\times60}$ is created.
- Rotating Subspaces

Simulated Datasets

- Translating Subspaces
  - Two point clouds with 100 Gaussian random variables in $\mathbb{R}^{20}$ were generated.
  - The two subspaces in which the point clouds live are orthogonal to each other in $\mathbb{R}^{100}$.
  - The first point cloud is static whereas the second one is translating in time $t \in \{1, 2, ...60\}$.
  - A 3-mode tensor $\mathcal{X} \in \mathbb{R}^{100 \times 200 \times 60}$ is created.
- Rotating Subspaces
  - Two point clouds with 100 Gaussian random variables in $\mathbb{R}^{20}$ were generated.
  - The first point cloud is static whereas the second one is rotating in time $t \in \{1, 2, ...60\}$ with the rotation matrix
    $\mathbf{A} = \mathbf{I}_{10\times 10} \otimes \begin{bmatrix} cos(\theta t) & sin(\theta t) \\ -sin(\theta t) & cos(\theta t) \end{bmatrix}$ and $\theta = \begin{cases} \frac{\pi}{120}, & t \leq 30 \\ \frac{\pi}{60}, & t > 30 \end{cases}$.
  - A 3-mode tensor $\mathcal{X} \in \mathbb{R}^{100 \times 200 \times 60}$ is created.

Simulated Datasets



- Figure 1: Low n-rank approximations of $\mathcal{X}$ are computed by HOSVD, LL-HOSVD(DD) and LL-HOSVD(SD) with various n-rank and the cluster number along each mode $C = (4, 4, 4)$. Sample outputs for translating (left) and rotating (right) subspaces: (a) original slice, (b) HOSVD, (c)LL-HOSVD(DD), (d) LL-HOSVD(SD).

Simulated Datasets

TABLE I
AVERAGE MSE FOR THE RECONSTRUCTED 3-WAY TENSOR $\mathscr{X} \in \mathbb{R}^{100\times200\times60}$ FOR MOVING SUBSPACES BY HOSVD, LL-HOSVD(DD) AND LL-HOSVD(SD) APPROACHES AT VARYING n-RANK OVER 25 TRIALS.

	$R=(3,3,3)$	$R=(5,5,5)$	$R=(7,7,7)$	$R=(9,9,9)$
HOSVD	0.1131	0.1006	0.0943	0.0885
LL-HOSVD(DD)	0.1029	0.0926	0.0792	0.0662
LL-HOSVD(SD)	0.0838	0.0584	0.0407	0.0285

TABLE II
AVERAGE MSE FOR THE RECONSTRUCTED 3-WAY TENSOR $\mathscr{X} \in \mathbb{R}^{100\times200\times60}$ FOR ROTATING SUBSPACES BY HOSVD, LL-HOSVD(DD) AND LL-HOSVD(SD) APPROACHES AT VARYING n-RANK OVER 25 TRIALS.

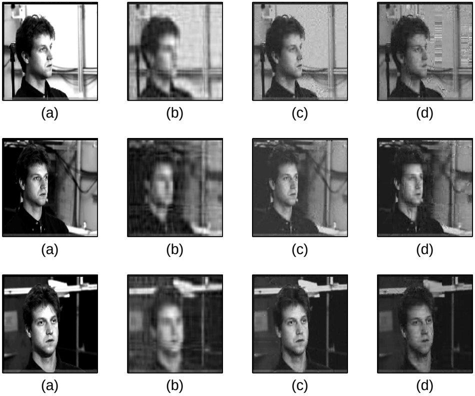
	$R=(3,3,3)$	$R=(5,5,5)$	$R=(7,7,7)$	$R=(9,9,9)$
HOSVD	0.1016	0.0896	0.0786	0.0685
LL-HOSVD(DD)	0.685	0.0540	0.0474	0.0432
LL-HOSVD(SD)	0.0493	0.0231	0.0137	0.0084

PIE Dataset

- A 3-mode tensor $\mathcal{X} \in \mathbb{R}^{122 \times 160 \times 138}$ is created from PIE dataset (Sim et al. 2003).
- The tensor contains 138 images from 6 different yaw angles and varying illumination conditions collected from a subject.
- Each image is converted to gray scale and downsampled to 122×160.
- n-rank$(\hat{\mathcal{Y}}_k) = (20, 25, 15)$ and the cluster number along each mode is chosen as $C = (4, 4, 4)$.

MICHIGAN STATE UNIVERSITY

PIE Dataset



- Figure 2: Frames corresponding to 3 different yaw angles obtained from approximated low n-rank tensor: (a) original image, (b) HOSVD, MSE $=$ 439.0140, (c) LL-HOSVD(DD), MSE $=$ 140.6469, (d) LL-HOSVD(SD), MSE $=$ 378.3899

Outline

Conclusions & Future Work

Conclusions:

Conclusions & Future Work

Conclusions:

- We introduced a new low-rank tensor approximation technique to learn the underlying nonlinear structure.

Conclusions & Future Work

Conclusions:

- We introduced a new low-rank tensor approximation technique to learn the underlying nonlinear structure.
- We proposed two approaches to decompose a tensor into its subtensors.

Conclusions & Future Work

Conclusions:

- We introduced a new low-rank tensor approximation technique to learn the underlying nonlinear structure.
- We proposed two approaches to decompose a tensor into its subtensors.
- Proposed approach provides better approximation than HOSVD by fitting piece-wise linear low-rank model to local non-linearities.

Conclusions & Future Work

Conclusions:

- We introduced a new low-rank tensor approximation technique to learn the underlying nonlinear structure.
- We proposed two approaches to decompose a tensor into its subtensors.
- Proposed approach provides better approximation than HOSVD by fitting piece-wise linear low-rank model to local non-linearities.

Conclusions & Future Work

Future work:

Conclusions & Future Work

Future work:

- Selection of parameters in the algorithm: the number of clusters and the appropriate rank.

Conclusions & Future Work

Future work:

- Selection of parameters in the algorithm: the number of clusters and the appropriate rank.
- Adapting the method to dynamic tensors for identifying structural changes to the tensor in time.

Conclusions & Future Work

Future work:

- Selection of parameters in the algorithm: the number of clusters and the appropriate rank.
- Adapting the method to dynamic tensors for identifying structural changes to the tensor in time.
- Combining the algorithm with the multiscale structure of GMRA to obtain a multi-resolution tree structure for high order datasets.

Conclusions & Future Work

Future work:

- Selection of parameters in the algorithm: the number of clusters and the appropriate rank.
- Adapting the method to dynamic tensors for identifying structural changes to the tensor in time.
- Combining the algorithm with the multiscale structure of GMRA to obtain a multi-resolution tree structure for high order datasets.
- Learning multiresolution tree structure provides better compression rate than HOSVD.

Outline

Acknowledgements

- This work was in part supported by NSF CCF-1422262, NSF CCF-1218377, NSF DMS-1416752 and NSA H98230-13-1-0275.