Higher Order Orthogonal Iteration of Tensors

Multi-Linear Subspace Learning

Dan Mathews Robin

MSc. Machine Intelligence
IITMK

July 27, 2021

What is Higher Order Orthogonal Iteration?

- Mathematically, a low rank approximation technique for tensors

What is Higher Order Orthogonal Iteration?

- Mathematically, a low rank approximation technique for tensors
- Can be considered as a unified view of dimension reduction techniques

What is Higher Order Orthogonal Iteration?

- Mathematically, a low rank approximation technique for tensors
- Can be considered as a unified view of dimension reduction techniques
- PCA - a special instance of HOOI

Agenda

- Primer on Principal Component Analysis(PCA), Single Value Decomposition(SVD) and Tensors
- Higher Order Orthogonal Iteration
- Relation of PCA and HOOI
- Application to Dimensionality Reduction

Principal Component Analysis

- Why PCA?

Principal Component Analysis

- Why PCA?
  - method of dimensionality reduction

Principal Component Analysis

- Why PCA?
  - method of dimensionality reduction
  - transformation of data from a high-dimensional space into a low-dimensional space

Principal Component Analysis

- Why PCA?
  - method of dimensionality reduction
  - transformation of data from a high-dimensional space into a low-dimensional space
  - this lower dimensional data - retains meaningful properties of the original data

Principal Component Analysis

Reduced no. of variables = trade-off for accuracy?

- Yes but actually no

Principal Component Analysis

Reduced no. of variables = trade-off for accuracy?

- Yes but actually no
- Doesn't hurt a trade a little accuracy for simplicity

Principal Component Analysis

Reduced no. of variables = trade-off for accuracy?

- Yes but actually no
- Doesn't hurt a trade a little accuracy for simplicity
- Smaller data sets -

Principal Component Analysis

Reduced no. of variables = trade-off for accuracy?

- Yes but actually no
- Doesn't hurt a trade a little accuracy for simplicity
- Smaller data sets -
  - easier to explore and visualize

Principal Component Analysis

Reduced no. of variables = trade-off for accuracy?

- Yes but actually no
- Doesn't hurt a trade a little accuracy for simplicity
- Smaller data sets -
    - easier to explore and visualize
    - makes analyzing data much easier and faster for machine learning algorithms

Principal Component Analysis

Reduced no. of variables = trade-off for accuracy?

- Yes but actually no
- Doesn't hurt a trade a little accuracy for simplicity
- Smaller data sets -
  - easier to explore and visualize
  - makes analyzing data much easier and faster for machine learning algorithms
  - without extraneous variables to process.

PCA (two dimensional idea)

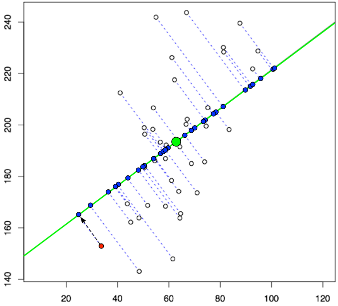


Figure 1: PCA in 2D

PCA (Mathematical Treatment I)

Let us take a model problem. We are given a set of matrices $M_k, k = 1, 2, \ldots K$, all of the same dimension $I \times J$.

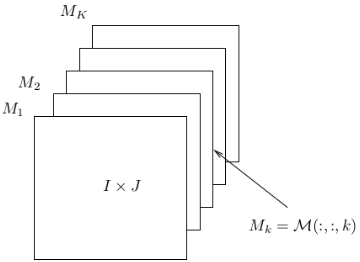


Figure 2: Set of K matrices $M_k R \in I \times J$ represented as an $I \times J \times K$ tensor $\mathcal{M}$

Tensor PCA and HOOI

PCA (Mathematical Treatment II)

- To apply PCA, lets first vectorize the matrix(image).

PCA (Mathematical Treatment II)

- To apply PCA, lets first vectorize the matrix(image).
- Each image $M_i \in R^{I \times J}$ is rearranged into a vector $x_i \in R^N$ where $N = IJ$ is the number of pixels in the image.

PCA (Mathematical Treatment II)

- To apply PCA, lets first vectorize the matrix(image).
- Each image $M_i \in R^{I \times J}$ is rearranged into a vector $x_i \in R^N$ where $N = IJ$ is the number of pixels in the image.
- First, let's define the mean and covariance of the image

PCA (Mathematical Treatment II)

- To apply PCA, lets first vectorize the matrix(image).
- Each image $M_i \in R^{I \times J}$ is rearranged into a vector $x_i \in R^N$ where $N = IJ$ is the number of pixels in the image.
- First, let's define the mean and covariance of the image
- $$\mu = \frac{1}{K} \sum_{i=1}^{K} x_i$$

PCA (Mathematical Treatment II)

- To apply PCA, lets first vectorize the matrix(image).
- Each image $M_i \in R^{I \times J}$ is rearranged into a vector $x_i \in R^N$ where $N = IJ$ is the number of pixels in the image.
- First, let's define the mean and covariance of the image
- $$\mu = \frac{1}{K}\sum_{i=1}^{K} x_i$$
- $$C = AA^T \in R^{N \times N}$$

PCA (Mathematical Treatment II)

- To apply PCA, lets first vectorize the matrix(image).
- Each image $M_i \in R^{I \times J}$ is rearranged into a vector $x_i \in R^N$ where $N = IJ$ is the number of pixels in the image.
- First, let's define the mean and covariance of the image
- $$\mu = \frac{1}{K}\sum_{i=1}^{K} x_i$$
- $$C = AA^T \in R^{N \times N}$$
- where $A \equiv [y_1, y_2, ...y_K]$, $y_i = x_i - \mu$, $i = 1, .., K$

PCA (Mathematical Treatment III)

- Low rank approximations to the data set A are obtained by computing a small number $r \ll K$ of the largest eigenvalues of C,

$$Cu_i = \lambda u_i, i = 1, ..., r$$

writing in the matrix form

PCA (Mathematical Treatment III)

- Low rank approximations to the data set A are obtained by computing a small number $r \ll K$ of the largest eigenvalues of C,

$$Cu_i = \lambda u_i, i = 1, ..., r$$

writing in the matrix form

- $$C = U_r \lambda_r U_r^T$$

PCA (Mathematical Treatment III)

- Low rank approximations to the data set A are obtained by computing a small number $r \ll K$ of the largest eigenvalues of C,

$$Cu_i = \lambda u_i, i = 1, ..., r$$

writing in the matrix form

- $$C = U_r \lambda_r U_r^T$$

- where $U_r = [u_1, ..., u_r]$ and $\Lambda_r = diag\{\lambda_1, ..., \lambda_r\}$

PCA (Mathematical Treatment III)

- Low rank approximations to the data set A are obtained by computing a small number $r \ll K$ of the largest eigenvalues of C,

$$Cu_i = \lambda u_i, i = 1, ..., r$$

writing in the matrix form

-
$$C = U_r \lambda_r U_r^T$$

- where $U_r = [u_1, ..., u_r]$ and $\Lambda_r = diag\{\lambda_1, ..., \lambda_r\}$
- note that U_r is the orthogonal vector and Λ_r is the diagonal matrix with the diagonal as the eigenvalues.

PCA (Mathematical Treatment IV)

Finally, we reorient the data from the original axes to the ones represented by the principal components. For that we project the original data points x_i to the space of U_r to obtain,

$$\widetilde{x}_i = U_r^T(x_i - \mu)$$

$$x_i = \mu + U_r \widetilde{x}_i$$

PCA and HOOI

SVD

Any real matrix $A_{m\times n}$ can be written (factorized) in the form of a product of three matrices $U_{m\times m}$, $\Sigma_{m\times n}$ and $V^T_{n\times m}$. This process is known as singular value decomposition. The diagonal entries $\sigma_i = \Sigma_{ii}$ of Σ are known as the singular values of M. Mathematically,

$$A = U\Sigma V^T$$

Relevance of SVD in our Discussion

As we discussed in the earlier slides for PCA

- $C = U_r \lambda_r U_r^T$, $\mu = \frac{1}{K}\sum_{i=1}^{K} x_i$, $C = AA^T \in R^{N \times N}$

This U_r can be obtained using SVD also.

Relevance of SVD in our Discussion

As we discussed in the earlier slides for PCA

- $C = U_r \lambda_r U_r^T$, $\mu = \frac{1}{K}\sum_{i=1}^{K} x_i$, $C = AA^T \in R^{N \times N}$
- where $U_r = [u_1, ..., u_r]$, U_r is the orthogonal vector and

This U_r can be obtained using SVD also.

Relevance of SVD in our Discussion

As we discussed in the earlier slides for PCA

- $C = U_r \lambda_r U_r^T$, $\mu = \frac{1}{K}\sum_{i=1}^{K} x_i$, $C = AA^T \in R^{N \times N}$
- where $U_r = [u_1, ..., u_r]$, U_r is the orthogonal vector and
- $A \equiv [y_1, y_2, ...y_K]$, $y_i = x_i - \mu$, $i = 1, .., K$

This U_r can be obtained using SVD also.

Relevance of SVD in our Discussion

As we discussed in the earlier slides for PCA

- $C = U_r \lambda_r U_r^T$, $\mu = \frac{1}{K}\sum_{i=1}^{K} x_i$, $C = AA^T \in R^{N\times N}$
- where $U_r = [u_1, ..., u_r]$, U_r is the orthogonal vector and
- $A \equiv [y_1, y_2, ...y_K]$, $y_i = x_i - \mu$, $i = 1, .., K$

This U_r can be obtained using SVD also.

- compute a truncated SVD of $A \in R^{N\times K}$ (till r values)

Relevance of SVD in our Discussion

As we discussed in the earlier slides for PCA

- $C = U_r \lambda_r U_r^T$, $\mu = \frac{1}{K}\sum_{i=1}^{K} x_i$, $C = AA^T \in R^{N\times N}$
- where $U_r = [u_1, ..., u_r]$, U_r is the orthogonal vector and
- $A \equiv [y_1, y_2, ...y_K]$, $y_i = x_i - \mu$, $i = 1, .., K$

This U_r can be obtained using SVD also.

- compute a truncated SVD of $A \in R^{N\times K}$ (till r values)
- $$A \approx U_r \Sigma_r V_r^T$$

Relevance of SVD in our Discussion

As we discussed in the earlier slides for PCA

- $C = U_r \lambda_r U_r^T$, $\mu = \frac{1}{K}\sum_{i=1}^{K} x_i$, $C = AA^T \in R^{N \times N}$
- where $U_r = [u_1, ..., u_r]$, U_r is the orthogonal vector and
- $A \equiv [y_1, y_2, ...y_K]$, $y_i = x_i - \mu$, $i = 1, .., K$

This U_r can be obtained using SVD also.

- compute a truncated SVD of $A \in R^{N \times K}$ (till r values)
- $$A \approx U_r \Sigma_r V_r^T$$
- where $\Sigma_r = diag\{\sigma_1, ..., \sigma_r\} \in R^{r \times r}$ is the diagonal matrix containing the r largest singular values of A in the descending order $\sigma_1, \sigma_2, ..., \sigma_r$

Tensors

Tensor Diagram

- Tensor - generalization of - vector and matrix

Tensors

Tensor Diagram

- Tensor - generalization of - vector and matrix
- Vector - 1D, Matrix - 2D, Tensor - N dimensional in general

Tensors

Tensor Diagram

- Tensor - generalization of - vector and matrix
- Vector - 1D, Matrix - 2D, Tensor - N dimensional in general
- 3D Tensor - vector of matrices

Tensors

Tensor Diagram

- Tensor - generalization of - vector and matrix
- Vector - 1D, Matrix - 2D, Tensor - N dimensional in general
- 3D Tensor - vector of matrices
- Originally introduced in physics, gained popularity in the 60's to analyze multi-way datasets.

Tensors

Tensor Diagram

- Tensor - generalization of - vector and matrix
- Vector - 1D, Matrix - 2D, Tensor - N dimensional in general
- 3D Tensor - vector of matrices
- Originally introduced in physics, gained popularity in the 60's to analyze multi-way datasets.
- PARFAC, HOSVD utilizes tensors

Tensors

Tensor Diagram

- Tensor - generalization of - vector and matrix
- Vector - 1D, Matrix - 2D, Tensor - N dimensional in general
- 3D Tensor - vector of matrices
- Originally introduced in physics, gained popularity in the 60's to analyze multi-way datasets.
- PARFAC, HOSVD utilizes tensors
- Next few slides - some mathematical aspects of Tensor calculations required for HOOI

Tensor Maths I

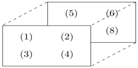


Figure 3: $\mathcal{A}$, a $2 \times 2 \times 2$ matrix

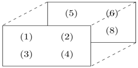


Figure 4: $\mathcal{B}$, another $2 \times 2 \times 2$ matrix

- The inner product of two such tensors $\mathcal{A}, \mathcal{B}$ can be calculated in the following form

Tensor Maths I

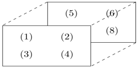


Figure 3: $\mathcal{A}$, a $2 \times 2 \times 2$ matrix

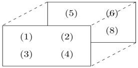


Figure 4: $\mathcal{B}$, another $2 \times 2 \times 2$ matrix

- The inner product of two such tensors $\mathcal{A}, \mathcal{B}$ can be calculated in the following form
- $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} \mathcal{A}_{ijk} \mathcal{B}_{ijk}$

Tensor Maths I

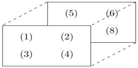


Figure 3: $\mathcal{A}$, a $2 \times 2 \times 2$ matrix

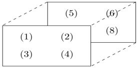


Figure 4: $\mathcal{B}$, another $2 \times 2 \times 2$ matrix

- The inner product of two such tensors $\mathcal{A}, \mathcal{B}$ can be calculated in the following form
- $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} \mathcal{A}_{ijk} \mathcal{B}_{ijk}$
- $= \sum_{i=1}^{2} \sum_{j=1}^{2} \mathcal{A}_{ij1} \mathcal{B}_{ij1} + \sum_{i=1}^{2} \sum_{j=1}^{2} \mathcal{A}_{ij2} \mathcal{B}_{ij2}$

Tensor Maths I

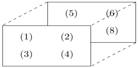


Figure 3: $\mathcal{A}$, a $2 \times 2 \times 2$ matrix

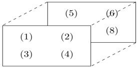


Figure 4: $\mathcal{B}$, another $2 \times 2 \times 2$ matrix

- The inner product of two such tensors $\mathcal{A}, \mathcal{B}$ can be calculated in the following form
- $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} \mathcal{A}_{ijk} \mathcal{B}_{ijk}$
- $= \sum_{i=1}^{2} \sum_{j=1}^{2} \mathcal{A}_{ij1} \mathcal{B}_{ij1} + \sum_{i=1}^{2} \sum_{j=1}^{2} \mathcal{A}_{ij2} \mathcal{B}_{ij2}$
- $= \sum_{i=1}^{2} \mathcal{A}_{i11} \mathcal{B}_{i11} + \sum_{i=1}^{2} \mathcal{A}_{i21} \mathcal{B}_{i21} + \sum_{i=1}^{2} \mathcal{A}_{i12} \mathcal{B}_{i12} + \sum_{i=1}^{2} \mathcal{A}_{i22} \mathcal{B}_{i22}$

Tensor Maths II

- extending this idea two T'th order tensors $\mathcal{A}, \mathcal{B} \in R^{I_1 \times I_2 \times \ldots \times I_T}$;

Tensor Maths II

- extending this idea two T'th order tensors $\mathcal{A}, \mathcal{B} \in R^{I_1 \times I_2 \times \ldots \times I_T}$;
-

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_T=1}^{I_T} \mathcal{A}_{i_1,,i_2,\ldots,i_T} \mathcal{B}_{i_1,,i_2,\ldots,i_T}$$

Tensor Maths II

- extending this idea two T'th order tensors $\mathcal{A}, \mathcal{B} \in R^{I_1 \times I_2 \times \ldots \times I_T}$;
- $$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_T=1}^{I_T} \mathcal{A}_{i_1,,i_2,\ldots,i_T} \mathcal{B}_{i_1,,i_2,\ldots,i_T}$$
- inducing the idea of Frobenius(Euclidean) norm

Tensor Maths II

- extending this idea two T'th order tensors $\mathcal{A}, \mathcal{B} \in R^{I_1 \times I_2 \times \ldots \times I_T}$;
- $$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_T=1}^{I_T} \mathcal{A}_{i_1,,i_2,\ldots,i_T} \mathcal{B}_{i_1,,i_2,\ldots,i_T}$$
- inducing the idea of Frobenius(Euclidean) norm
- $$\|\mathcal{A}\|_F = \langle \mathcal{A}, \mathcal{A} \rangle^{1/2}$$

Tensor Maths II

- extending this idea two T'th order tensors $\mathcal{A}, \mathcal{B} \in R^{I_1 \times I_2 \times \ldots \times I_T}$;
- $$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_T=1}^{I_T} \mathcal{A}_{i_1,,i_2,\ldots,i_T} \mathcal{B}_{i_1,,i_2,\ldots,i_T}$$
- inducing the idea of Frobenius(Euclidean) norm
- $$\|\mathcal{A}\|_F = \langle \mathcal{A}, \mathcal{A} \rangle^{1/2}$$
- The Frobenius norm can be used to measure the distance between tensors $\mathcal{A}$ and $\mathcal{B}$ as

Tensor Maths II

- extending this idea two T'th order tensors $\mathcal{A}, \mathcal{B} \in R^{I_1 \times I_2 \times \ldots \times I_T}$;
- $$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_T=1}^{I_T} \mathcal{A}_{i_1,,i_2,\ldots,i_T} \mathcal{B}_{i_1,,i_2,\ldots,i_T}$$
- inducing the idea of Frobenius(Euclidean) norm
- $$\|\mathcal{A}\|_F = \langle \mathcal{A}, \mathcal{A} \rangle^{1/2}$$
- The Frobenius norm can be used to measure the distance between tensors $\mathcal{A}$ and $\mathcal{B}$ as
- $$dist(\mathcal{A}, \mathcal{B}) = \|\mathcal{A} - \mathcal{B}\|_F$$

Tensor Maths III (Modes of Tensors)

- The number of dimensions (ways) of a tensor is its **order**,denoted by N. Each dimension (way) is called a **mode**.

Tensor Maths III (Modes of Tensors)

- The number of dimensions (ways) of a tensor is its **order**,denoted by N. Each dimension (way) is called a **mode**.

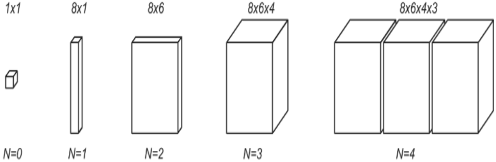


Figure 5: Illustration of tensors of order N= 0,1,2,3,4.

Tensor Maths III (Modes of Tensors Contd.)

- In our discussion, we use mode-n to indicate the nth mode for clarity.

Tensor Maths III (Modes of Tensors Contd.)

- In our discussion, we use mode-n to indicate the nth mode for clarity.
- When we have a set of N vectors or matrices, one for each mode, we denote the nth (i.e., mode-n) **vector** or **matrix** using a superscript in parenthesis,for example, as $\mathbf{u}^{(n)}$ or $\mathbf{U}^{(n)}$ and the whole set as $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(N)}\}$ or $\{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}\}$, or more compactly as $\{\mathbf{u}^{(n)}\}$ or $\{\mathbf{U}^{(N)}\}$.

Tensor Maths III (Modes of Tensors Contd.)

- In our discussion, we use mode-n to indicate the nth mode for clarity.
- When we have a set of N vectors or matrices, one for each mode, we denote the nth (i.e., mode-n) **vector** or **matrix** using a superscript in parenthesis,for example, as $\mathbf{u}^{(n)}$ or $\mathbf{U}^{(n)}$ and the whole set as $\{\ \mathbf{u}^{(1)}, \mathbf{u}^{(2)}, ..., \mathbf{u}^{(N)}\ \}$ or $\{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, ..., \mathbf{U}^{(N)}\ \}$, or more compactly as$\{\ \mathbf{u}^{(n)}\ \}or\{\ \mathbf{U}^{(N)}\ \}$.
- Next lets, look at mode-n vectors and mode-n slices of $\mathcal{A}$.

Tensor Maths III (Modes of Tensors Contd. II)

- The **mode-n vectors** of $\mathcal{A}$ are defined as the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying index i_n while keeping all the other indices fixed.

Tensor Maths III (Modes of Tensors Contd. II)

- The **mode-n vectors** of $\mathcal{A}$ are defined as the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying index i_n while keeping all the other indices fixed.

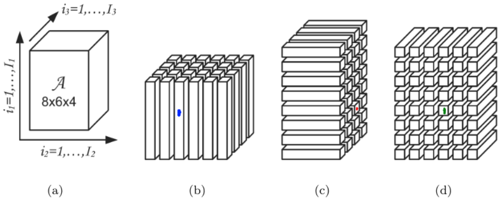


Figure 6: Illustration of the mode-n vectors: (a) a tensor $\mathcal{A} \in \mathbb{R}^{8\times6\times4}$,(b) the mode-1 vectors, (c) the mode-2 vectors, and (d) the mode-3 vectors.

Tensor Maths III (Modes of Tensors Contd. II)

- The **mode-n vectors** of $\mathcal{A}$ are defined as the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying index i_n while keeping all the other indices fixed.

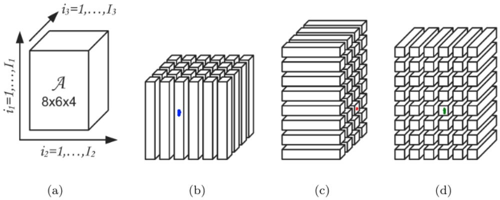
- 

Figure 6: Illustration of the mode-n vectors: (a) a tensor $\mathcal{A} \in \mathbb{R}^{8\times6\times4}$,(b) the mode-1 vectors, (c) the mode-2 vectors, and (d) the mode-3 vectors.

- notation, blue - $\mathcal{A}(:, 3, 1)$, red - $\mathcal{A}(6, :, 3)$, green - $\mathcal{A}(5, 4, :)$

Tensor Maths III (Modes of Tensors Contd. III)

- Similarly, the i_nth **mode-n slice** of $\mathcal{A}$ are defined as the $(N-1)$th-order tensor obtained by fixing the mode-n index of $\mathcal{A}$ to be i_n: $\mathcal{A}(:, \ldots :, i_n, :, \ldots, :)$

Tensor Maths III (Modes of Tensors Contd. III)

- Similarly, the i_nth **mode-n slice** of $\mathcal{A}$ are defined as the $(N-1)$th-order tensor obtained by fixing the mode-n index of $\mathcal{A}$ to be i_n: $\mathcal{A}(:, \ldots :, i_n, :, \ldots, :)$

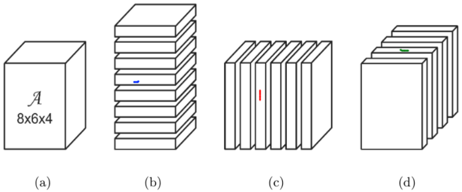


Figure 7: Illustration of the mode-n slices: (a) a tensor $\mathcal{A} \in \mathbb{R}^{8\times 6\times 4}$,(b) the mode-1 slices, (c) the mode-2 slices, and (d) the mode-3 slices.

Tensor Maths III (Modes of Tensors Contd. III)

- Similarly, the i_nth **mode-n slice** of $\mathcal{A}$ are defined as the $(N-1)$th-order tensor obtained by fixing the mode-n index of $\mathcal{A}$ to be i_n: $\mathcal{A}(:, \dots :, i_n, :, \dots, :)$

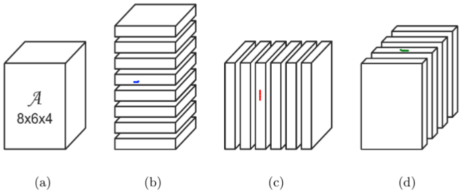
- 

Figure 7: Illustration of the mode-n slices: (a) a tensor $\mathcal{A} \in \mathbb{R}^{8\times6\times4}$,(b) the mode-1 slices, (c) the mode-2 slices, and (d) the mode-3 slices.

- notation, blue - $\mathcal{A}(4, :, :)$, red - $\mathcal{A}(:, 3, :)$, green - $\mathcal{A}(:, :, 2)$

Tensor Maths IV (Basic Operations - Unfolding)

- Also known as tensor to matrix transformation, flattening, matricization

properties

Tensor Maths IV (Basic Operations - Unfolding)

- Also known as tensor to matrix transformation, flattening, matricization
- A tensor can be unfolded into a matrix by rearranging its **mode-n** vectors.The **mode-n unfolding** of $\mathcal{A}$ is denoted by $A_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \ldots \times I_n - 1 \times I_n + 1 \times \ldots \times I_N)}$, where the column vectors of $A_{(n)}$ are the **mode-n** vectors of $\mathcal{A}$.

properties

Tensor Maths IV (Basic Operations - Unfolding)

- Also known as tensor to matrix transformation, flattening, matricization
- A tensor can be unfolded into a matrix by rearranging its **mode-n** vectors. The **mode-n unfolding** of $\mathcal{A}$ is denoted by $A_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \ldots \times I_n - 1 \times I_n + 1 \times \ldots \times I_N)}$, where the column vectors of $A_{(n)}$ are the **mode-n** vectors of $\mathcal{A}$.

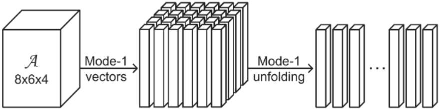


Figure 8: Visual illustration of the mode-1 unfolding from $\mathcal{A}_{8\times6\times4}$ to $A_{8\times24}$

properties

Tensor Maths IV (Basic Operations - Unfolding)

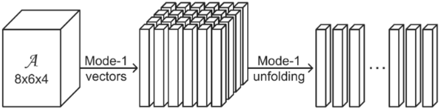
- Also known as tensor to matrix transformation, flattening, matricization
- A tensor can be unfolded into a matrix by rearranging its **mode-n** vectors. The **mode-n unfolding** of $\mathcal{A}$ is denoted by $A_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \ldots \times I_n - 1 \times I_n + 1 \times \ldots \times I_N)}$, where the column vectors of $A_{(n)}$ are the **mode-n** vectors of $\mathcal{A}$.
- 

Figure 8: Visual illustration of the mode-1 unfolding from $\mathcal{A}_{8\times 6\times 4}$ to $A_{8\times 24}$

- Similarly vectorization would provide a 192×1 matrix.

properties

Tensor Maths IV (Basic Operations - Mode-n product)

- The **mode-n product** of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N)}$ by a matrix $U \in \mathbb{R}^{J_n \times I_n}$ is a tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times \ldots \times I_n - 1 \times J_n \times I_n + 1 \times \ldots \times I_N)}$ denoted by

$$\mathcal{B} = \mathcal{A} \times_n U$$

.

Tensor Maths IV (Basic Operations - Mode-n product)

- The **mode-n product** of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N)}$ by a matrix $U \in \mathbb{R}^{J_n \times I_n}$ is a tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times \ldots \times I_n - 1 \times J_n \times I_n + 1 \times \ldots \times I_N)}$ denoted by

$$\mathcal{B} = \mathcal{A} \times_n U$$

.

- where each entry of $\mathcal{B}$ is defined as the sum of products of corresponding entries in $\mathcal{A}$ and U:

$$\mathcal{B}(i_1, ..., i_{n-1}, j_n, i_{n+1}, ..., i_N) = \sum_{i_n} \mathcal{A}(i_1, ..., i_N).U(j_n, i_n)$$

Tensor Maths IV (Basic Operations - Mode-n product)

- The **mode-n product** of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N)}$ by a matrix $U \in \mathbb{R}^{J_n \times I_n}$ is a tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times \ldots \times I_n - 1 \times J_n \times I_n + 1 \times \ldots \times I_N)}$ denoted by

$$\mathcal{B} = \mathcal{A} \times_n U$$

.

- where each entry of $\mathcal{B}$ is defined as the sum of products of corresponding entries in $\mathcal{A}$ and U:

$$\mathcal{B}(i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N) = \sum_{i_n} \mathcal{A}(i_1, \ldots, i_N).U(j_n, i_n)$$

- equivalent to premultiplying each mode-n vector of $\mathcal{A}$ by U. Thus,the mode-n product above can be written using the mode-n unfolding as

$$B_{(n)} = UA_{(n)}$$

Tensor Maths IV (Basic Operations - Mode-*1* product)

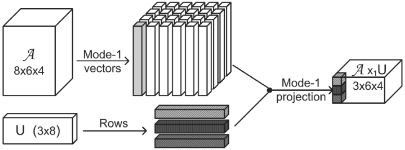


Figure 9: Visual illustration of the mode-n(mode-1) multiplication

Tensor Maths V (Some Properties of Mode Multiplication)

- multiplication of more than mode possible, order immaterial provided the modes are distinct $\mathcal{A} \times_{n_1} U_{n_1} \times_{n_2} U_{n_2} \ldots \times_{n_p} U_{n_p}$

Tensor Maths V (Some Properties of Mode Multiplication)

- multiplication of more than mode possible, order immaterial provided the modes are distinct $\mathcal{A} \times_{n_1} U_{n_1} \times_{n_2} U_{n_2} \ldots \times_{n_p} U_{n_p}$
- when the same mode is involved, $\mathcal{A} \times_p U \times_p V = \mathcal{A} \times_p (VU)$

Tensor Maths V (Some Properties of Mode Multiplication)

- multiplication of more than mode possible, order immaterial provided the modes are distinct $\mathcal{A} \times_{n_1} U_{n_1} \times_{n_2} U_{n_2} \ldots \times_{n_p} U_{n_p}$
- when the same mode is involved, $\mathcal{A} \times_p U \times_p V = \mathcal{A} \times_p (VU)$
- in particular if $U^T U = I$(orthogonal), then $\mathcal{A} \times_p U \times_p U_T = \mathcal{A}$

Tensor Maths V (Some Properties of Mode Multiplication)

- multiplication of more than mode possible, order immaterial provided the modes are distinct $\mathcal{A} \times_{n_1} U_{n_1} \times_{n_2} U_{n_2} \ldots \times_{n_p} U_{n_p}$
- when the same mode is involved, $\mathcal{A} \times_p U \times_p V = \mathcal{A} \times_p (VU)$
- in particular if $U^T U = I$(orthogonal), then $\mathcal{A} \times_p U \times_p U_T = \mathcal{A}$
- $\langle \mathcal{A} \times_p U, \mathcal{B} \rangle = \langle \mathcal{A}, \mathcal{B} \times_p U^T \rangle$

Tensor Maths V (Some Properties of Mode Multiplication)

- multiplication of more than mode possible, order immaterial provided the modes are distinct $\mathcal{A} \times_{n_1} U_{n_1} \times_{n_2} U_{n_2} \ldots \times_{n_p} U_{n_p}$
- when the same mode is involved, $\mathcal{A} \times_p U \times_p V = \mathcal{A} \times_p (VU)$
- in particular if $U^T U = I$(orthogonal), then $\mathcal{A} \times_p U \times_p U_T = \mathcal{A}$
- $\langle \mathcal{A} \times_p U, \mathcal{B} \rangle = \langle \mathcal{A}, \mathcal{B} \times_p U^T \rangle$
- if $U^T U = I$, $\|\mathcal{A} \times_p U\|_F^2 = \|\mathcal{A}\|_F^2$, $\mathcal{A} \times_n I = \mathcal{A}$ HOOI

Tensor Maths V (Some Properties of Mode Multiplication)

- multiplication of more than mode possible, order immaterial provided the modes are distinct $\mathcal{A} \times_{n_1} U_{n_1} \times_{n_2} U_{n_2} \ldots \times_{n_p} U_{n_p}$
- when the same mode is involved, $\mathcal{A} \times_p U \times_p V = \mathcal{A} \times_p (VU)$
- in particular if $U^T U = I$(orthogonal), then $\mathcal{A} \times_p U \times_p U_T = \mathcal{A}$
- $\langle \mathcal{A} \times_p U, \mathcal{B} \rangle = \langle \mathcal{A}, \mathcal{B} \times_p U^T \rangle$
- if $U^T U = I$, $\|\mathcal{A} \times_p U\|_F^2 = \|\mathcal{A}\|_F^2$, $\mathcal{A} \times_n I = \mathcal{A}$ HOOI
- mode-n unfolding of $\mathcal{A}$, denoted by $\mathcal{A}_{(n)}$ - a matrix unfolding

Higher Order Orthogonal Iteration

- iterative algorithm for low rank approximations of tensors

Higher Order Orthogonal Iteration

- iterative algorithm for low rank approximations of tensors
- Let $\mathcal{A}$ be an $I_1 \times I_2 \times \ldots \times I_T$ tensor and let $r_1, r_2, \ldots r_T$ be a set of integers satisfying $1 \leq r_n \leq I_n$, for $n = 1, \ldots, T$.

Higher Order Orthogonal Iteration

- iterative algorithm for low rank approximations of tensors
- Let $\mathcal{A}$ be an $I_1 \times I_2 \times \ldots \times I_T$ tensor and let $r_1, r_2, \ldots r_T$ be a set of integers satisfying $1 \leq r_n \leq I_n$, for $n = 1, \ldots, T$.
- The rank-$\{ r_1, r_2, \ldots r_T \}$ approximation problem is to find a set of $I_n \times r$ matrices $U^{(n)}$ with orthogonal columns, $n = 1, \ldots, T$ and a $r_1 \times \ldots \times r_T$ core tensor $\mathcal{B}$ such that the optimization problem

$$\min_{\substack{U^{(1)}, U^{(2)}, \ldots, U^{(T)} \\ \mathcal{B}}} \left\| \mathcal{A} - \mathcal{B} \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_T U^{(T)} \right\|_F^2$$

Higher Order Orthogonal Iteration

- iterative algorithm for low rank approximations of tensors
- Let $\mathcal{A}$ be an $I_1 \times I_2 \times \ldots \times I_T$ tensor and let $r_1, r_2, \ldots r_T$ be a set of integers satisfying $1 \leq r_n \leq I_n$, for $n = 1, \ldots, T$.
- The rank-$\{\ r_1, r_2, \ldots r_T\ \}$ approximation problem is to find a set of $I_n \times r$ matrices $U^{(n)}$ with orthogonal columns, $n = 1, \ldots, T$ and a $r_1 \times \ldots \times r_T$ core tensor $\mathcal{B}$ such that the optimization problem

$$\min_{\substack{U^{(1)}, U^{(2)}, \ldots, U^{(T)} \\ \mathcal{B}}} \left\| \mathcal{A} - \mathcal{B} \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_T U^{(T)} \right\|_F^2$$

- optimal $\mathcal{B} = \mathcal{A} \times_{(1)} {U^{(1)}}^T \times_{(2)} {U^{(2)}}^T \ldots \times_T {U^{(T)}}^T$ properties

Higher Order Orthogonal Iteration contd.

- an alternating least squares approach(ALS)

Higher Order Orthogonal Iteration contd.

- an alternating least squares approach(ALS)
  - factorizes a given matrix R into two factors U and V such that $R \approx U^T V$

Higher Order Orthogonal Iteration contd.

- an alternating least squares approach(ALS)
  - factorizes a given matrix R into two factors U and V such that $R \approx U^T V$
- successively solves the restricted optimization problems

$$\min_{U^{(p)}} \left\| \mathcal{A} - \mathcal{B} \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_T U^{(T)} \right\|_F^2$$

Higher Order Orthogonal Iteration contd.

- an alternating least squares approach(ALS)
  - factorizes a given matrix R into two factors U and V such that $R \approx U^T V$
- successively solves the restricted optimization problems

$$\min_{U^{(p)}} \left\| \mathcal{A} - \mathcal{B} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_T U^{(T)} \right\|_F^2$$

- optimization done for p^{th} matrix $U^{(p)}$ while the other matrices $U^{(i)}, i \neq p$ are kept constant for the particular instance and $U^{(p)}$ is treated as the only unknown.

Higher Order Orthogonal Iteration contd.

- an alternating least squares approach(ALS)
  - factorizes a given matrix R into two factors U and V such that $R \approx U^T V$
- successively solves the restricted optimization problems

$$\min_{U^{(p)}} \left\| \mathcal{A} - \mathcal{B} \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_T U^{(T)} \right\|_F^2$$

- optimization done for p^{th} matrix $U^{(p)}$ while the other matrices $U^{(i)}, i \neq p$ are kept constant for the particular instance and $U^{(p)}$ is treated as the only unknown.
- HOOI algo. for 3^{rd} order tensor is explained in the next slide. Extension to higher order tensors is straightforward.

HOOI - Algorithm

input : $\mathcal{A}_{I\times J\times K}$ and r_1, r_2, r_3
output: $L \in \mathbb{R}^{I\times r_1}$, $R \in \mathbb{R}^{I\times r_2}$, $V \in \mathbb{R}^{I\times r_3}, \mathcal{B}$

Choose initial R, V with orthonormal columns ;

while *until convergence* **do**

$\mathcal{C} = \mathcal{A} \times_2 R^T \times_3 V^T$;
$L = SVD(r_1, \mathcal{C}_{(1)})$;
$\mathcal{D} = \mathcal{A} \times_1 L^T \times_3 V^T$;
$R = SVD(r_2, \mathcal{D}_{(2)})$;
$\mathcal{E} = \mathcal{A} \times_1 L^T \times_2 R^T$;
$V = SVD(r_3, \mathcal{E}_{(3)})$;

end

$\mathcal{B} = \mathcal{E} \times_3 V^T$

Algorithm 1: HOOI

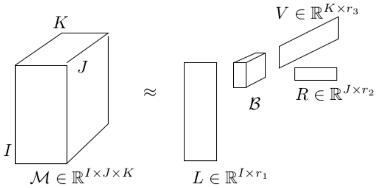


Figure 10: $\mathcal{M}$ approximated by $\mathcal{B} \times_1 L \times_2 R \times_3 V$. $\mathcal{B}$- core tensor, $L(U^{(1)}), R(U^{(2)}), V(U^{(3)})$ are projection matrices

Relation between PCA and HOOI

3^{rd} order tensor

- PCA can be regarded as a special case of HOOI

Relation between PCA and HOOI

3^{rd} order tensor

- PCA can be regarded as a special case of HOOI
- Let $\mathcal{M}$ be a 3^{rd} order tensor with dimensions $I \times J \times K$ and individual matrices being $M_k = \mathcal{M}(:,:,k)$. Let $I_I \in \mathbb{R}^{I \times I}$ and $I_J \in \mathbb{R}^{J \times J}$ be identity matrices in the first and second dimensions. Assuming that the matrices are centered, i.e., that $\mu = \frac{1}{k} \sum_{p=1}^{K} M_p = 0$. Then the following course of actions for dimensionality reduction are equal:

Relation between PCA and HOOI

3^{rd} order tensor

- PCA can be regarded as a special case of HOOI
- Let $\mathcal{M}$ be a 3^{rd} order tensor with dimensions $I \times J \times K$ and individual matrices being $M_k = \mathcal{M}(:,:,k)$. Let $I_I \in \mathbb{R}^{I \times I}$ and $I_J \in \mathbb{R}^{J \times J}$ be identity matrices in the first and second dimensions. Assuming that the matrices are centered, i.e., that $\mu = \frac{1}{k} \sum_{p=1}^{K} M_p = 0$. Then the following course of actions for dimensionality reduction are equal:
  - using HOOI to compute a rank(I, J, r) approximation $\mathcal{B} \times_1 I_I \times_2 I_J \times_3 V_r = \mathcal{B} \times_3 V_r$ to $\mathcal{M}$, where $V_r \in \mathbb{R}^{K \times r}$ is the projection matrix determined by HOOI, and

Relation between PCA and HOOI

3^{rd} order tensor

- PCA can be regarded as a special case of HOOI
- Let $\mathcal{M}$ be a 3^{rd} order tensor with dimensions $I \times J \times K$ and individual matrices being $M_k = \mathcal{M}(:,:,k)$. Let $I_I \in \mathbb{R}^{I \times I}$ and $I_J \in \mathbb{R}^{J \times J}$ be identity matrices in the first and second dimensions. Assuming that the matrices are centered, i.e., that $\mu = \frac{1}{k}\sum_{p=1}^{K} M_p = 0$. Then the following course of actions for dimensionality reduction are equal:
  - using HOOI to compute a rank(I, J, r) approximation $\mathcal{B} \times_1 I_I \times_2 I_J \times_3 V_r = \mathcal{B} \times_3 V_r$ to $\mathcal{M}$, where $V_r \in \mathbb{R}^{K \times r}$ is the projection matrix determined by HOOI, and
  - using PCA to compute projection matrix $U_r \in \mathbb{R}^{IJ \times r}$

PCA Review

1. Introduction
2. Primer
3. HOOI
4. PCA and HOOI
5. Application of HOOI
6. References

Application

- HOOI is useful for dimension reduction because the memory required to store projection matrices L, R, and V and the core tensor $\mathcal{B}$, i. e., $Ir1 + Jr2 + Kr3 + r1r2r3$ is often significantly less than the storage IJK required for the original $I \times J \times K$ tensor, $\mathcal{M}$.

Application

- HOOI is useful for dimension reduction because the memory required to store projection matrices L, R, and V and the core tensor $\mathcal{B}$, i. e., $Ir1 + Jr2 + Kr3 + r1r2r3$ is often significantly less than the storage IJK required for the original $I \times J \times K$ tensor, $\mathcal{M}$.
- Not only is the memory saved, but also when the data processed, the algorithms can use it quicker since the dimensions are reduced.

Application

- HOOI is useful for dimension reduction because the memory required to store projection matrices L, R, and V and the core tensor $\mathcal{B}$, i. e., $Ir1 + Jr2 + Kr3 + r1r2r3$ is often significantly less than the storage IJK required for the original $I \times J \times K$ tensor, $\mathcal{M}$.
- Not only is the memory saved, but also when the data processed, the algorithms can use it quicker since the dimensions are reduced.
- Can be used in classification as well as regression tasks dimensionality reduction

[1] Bernard N. Sheehan and Yousef Saad *Higher order orthogonal iteration of tensors (HOOI) and its relation to PCA and GLRAM.*. Proceedings of the 2007 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2007.

[2] Haiping Lu *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. CRC Press, 2012.

Thanks!