Tensor Decompositions and Applications

Xiaochen Zhang

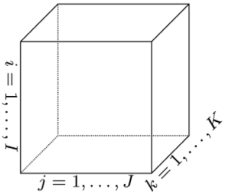
September 18, 2017

Outline

- Definitions and Operations
- CP Decomposition
- Tucker Decomposition
- Toolbox

Definitions and Operations: What Is Tensor

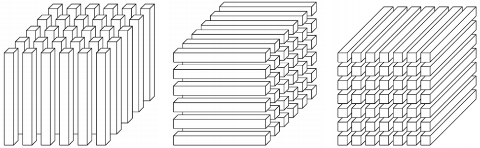
1. A tensor is a multidimensional array
2. An N-dimensional array is called N-way tensor or Nth-order tensor



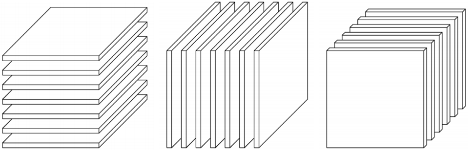
Definitions and Operations: What Is Tensor

	Matrix A	Tensor $\mathcal{X}$
Element	a_{ij}	x_{ijk}
Subarrays	$a_{:j}$(columns) $a_{i:}$(rows)	$x_{ij:}$(Fibers) $X_{i::}$(Slices)
Norm	$\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}$	$\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{s} x_{ijk}^2}$
Symmetry	$A = A^T$	$x_{ijk} = x_{ikj} = x_{jik} = x_{jki} = x_{kij} = x_{kji}$

Definitions and Operations: Subarrays



(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$



(a) Horizontal slices: $\mathbf{X}_{i::}$ (b) Lateral slices: $\mathbf{X}_{:j:}$ (c) Frontal slices: $\mathbf{X}_{::k}$ (or $\mathbf{X}_k$)

Definitions and Operations: Matricization

1. Matricization, also known as unfolding or flattening, is the process of reordering the elements of an N-way array into a matrix.
2. The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_m}$ is denoted by $X(n)$ and arranges the mode-n fibers to be the columns of the resulting matrix.

Definitions and Operations: Matricization

Let the frontal slices of $\mathcal{X} \in \mathbb{R}^{3\times4\times2}$ be

$$\mathbf{X}_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}.$$

Then the three mode-n unfoldings are

$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix},$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix},$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & \cdots & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & \cdots & 21 & 22 & 23 & 24 \end{bmatrix}.$$

Definitions and Operations: Multiplication

The n-mode (matrix) product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $U \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n U$ and is of size $I_1 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N$. Elementwise, we have

$$(\mathcal{X} \times_n U)_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n}$$

Some properties:

$$\mathcal{X} \times_m A \times_n B = \mathcal{X} \times_n B \times_m A \quad (m \neq n)$$

$$\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA)$$

Definitions and Operations: Multiplication

Kronecker product:
Given matrices $A \in \mathbb{R}^{I \times J}$ and $B \in \mathbb{R}^{K \times L}$, their Kronecker product is denoted by $A \otimes B$. The result is a matrix of size $(IK) \times (JL)$ defined by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_1 \otimes \mathbf{b}_2 & \mathbf{a}_1 \otimes \mathbf{b}_3 & \cdots & \mathbf{a}_J \otimes \mathbf{b}_{L-1} & \mathbf{a}_J \otimes \mathbf{b}_L \end{bmatrix}.$$

Khatri-Rao product:
It is the matching columnwise Kronecker product. Given matrices $A \in \mathbb{R}^{I \times K}$ and $B \in \mathbb{R}^{J \times K}$, their Khatri-Rao product is denoted by $A \odot B$. The result is a matrix of size $(IJ) \times K$ defined by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_K \otimes \mathbf{b}_K \end{bmatrix}.$$

Definitions and Operations: Multiplication

Hadamard product:
The Hadamard product is the elementwise matrix product. Given matrices A and B, both of size $I \times J$, their Hadamard product is denoted by $A * B$. The result is also of size $I \times J$ and defined by

$$\mathbf{A} * \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix}.$$

Definitions and Operations: Rank

An N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is rank one if it can be written as the outer product of N vectors, i.e.,

$$\mathcal{X} = a^{(1)} \circ a^{(2)} \circ \ldots \circ a^{(N)}$$

Each element is the product of the corresponding vector elements:

$$x_{i_1 i_2 \ldots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \ldots a^{(N)}_{i_N}$$

CP Decomposition

The CP decomposition factorizes a tensor into a sum of component rank-one tensors. For example, given a third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, we wish to write it as:

$$\mathcal{X} \approx \sum_{r=1}^{R} a_r \circ b_r \circ c_r$$

where R is a positive integer and $a_r \in \mathbb{R}^I$, $b_r \in \mathbb{R}^J$ and $c_r \in \mathbb{R}^K$ for $r = 1, \dots, R$. Elementwise:

$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$$

CP Decomposition

The factor matrix is the combination of the vectors from the rank-one components:

$$A = [a_1; a_2; \dots a_R]$$

Then the CP model can be expressed as:

$$\mathcal{X} \approx [A, B, C] \equiv \sum_{r=1}^{R} a_r \circ b_r \circ c_r$$

If we normalize matrices to length one:

$$\mathcal{X} \approx [\lambda; A, B, C] \equiv \sum_{r=1}^{R} \lambda_r a_r \circ b_r \circ c_r$$

CP Decomposition: Rank

1. The rank of a tensor $\mathcal{X}$, denoted rank($\mathcal{X}$), is defined as the smallest number of rank-one tensors that generate $\mathcal{X}$ as their sum. In other words, this is the smallest number of components in an exact CP decomposition.
2. An exact CP decomposition with $R = \text{rank}(\mathcal{X})$ components is called the rank decomposition.
3. The definition of tensor rank is an exact analogue to the definition of matrix rank.

CP Decomposition: Rank

- The rank of a real-valued tensor may actually be different over $\mathbb{R}$ and $\mathbb{C}$.

  Here are frontal slices of a tensor:

  $$\mathbf{X}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{X}_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

  The rank decomposition over $\mathbb{R}$:

  $$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix},$$

  The rank decomposition over $\mathbb{C}$:

  $$\mathbf{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix}, \quad \mathbf{B} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}, \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}.$$

CP Decomposition: Rank

- There is no straightforward algorithm to determine the rank of a specific given tensor. The problem is NP-hard.
- Tensors may have different maximum and typical ranks.
  1. The maximum rank is defined as the largest attainable rank.
  2. The typical rank is any rank that occurs with probability greater than zero (i.e., on a set with positive Lebesgue measure).

For the collection of $I \times J$ matrices, the maximum and typical ranks are identical and equal to $\min\{I, J\}$. For tensors, the two ranks may be different. Moreover, over $\mathbb{R}$, there may be more than one typical rank, whereas there is always only one typical rank over $\mathbb{C}$.

CP Decomposition: Rank Decomposition

- Higher-order tensors is that their rank decompositions are often unique, whereas matrix decompositions are not.
  Let $X \in \mathbb{R}^{I \times J}$ be a matrix of rank R. Then a rank decomposition of this matrix is

$$X = AB^T = \sum_{r=1}^{R} a_r \circ b_r$$

  If the SVD of X is $U\Sigma V^T$, then we can choose $A = U\Sigma W$ and $B = VW$, where W is some $R \times R$ orthogonal matrix.

CP Decomposition: Rank Decomposition

The k-rank of a matrix A, denoted k_A, is defined as the maximum value k such that any k columns are linearly independent.
A sufficient condition for uniqueness for CP decomposition of a three-way tensor is:

$$k_A + k_B + k_C \geq 2R + 2$$

For N-way tensor, the condition is:

$$\sum_{n=1}^{N} k_{A^{(n)}} \geq 2R + (N - 1)$$

For a given Three-way tensor, its CP decomposition is deterministically or generically (i.e., with probability one) unique if

$$R \leq K \quad and \quad R(R-1) \leq I(I-1)J(J-1)/2$$

CP Decomposition: Low Rank Approximation

Let R be the rank of a matrix A and assume its SVD is given by

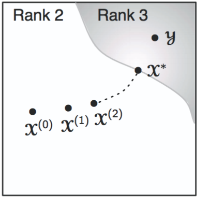
$$A = \sum_{r=1}^{R} \sigma_r u_r \circ v_r \quad with \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_R > 0$$

Then a rank-k approximation that minimizes $||A - B||$ is given by

$$B = \sum_{r=1}^{k} \sigma_r u_r \circ v_r$$

CP Decomposition: Low Rank Approximation

The best rank-k approximation may not exist.



Border rank: the minimum number of rank-one tensors that are sufficient to approximate the given tensor with arbitrarily small nonzero error.

$$\widetilde{\mathrm{rank}}(\mathcal{X}) = \min\{\, r \mid \text{for any } \epsilon > 0, \text{ there exists a tensor } \mathcal{E} \text{ such that } \|\mathcal{E}\| < \epsilon \text{ and } \mathrm{rank}(\mathcal{X} + \mathcal{E}) = r \,\}.$$

Algorithm:

procedure CP-ALS($\mathfrak{X}$,R)
  initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$
  repeat
    for $n = 1, \ldots, N$ **do**
      $\mathbf{V} \leftarrow \mathbf{A}^{(1)\mathsf{T}}\mathbf{A}^{(1)} * \cdots * \mathbf{A}^{(n-1)\mathsf{T}}\mathbf{A}^{(n-1)} * \mathbf{A}^{(n+1)\mathsf{T}}\mathbf{A}^{(n+1)} * \cdots * \mathbf{A}^{(N)\mathsf{T}}\mathbf{A}^{(N)}$
      $\mathbf{A}^{(n)} \leftarrow \mathbf{X}_{(n)}(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)})\mathbf{V}^{\dagger}$
      normalize columns of $\mathbf{A}^{(n)}$ (storing norms as $\boldsymbol{\lambda}$)
    end for
  until fit ceases to improve or maximum iterations exhausted
  return $\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

Use three-way tensor as an example:

$$\min_{\hat{\mathfrak{X}}} \|\mathfrak{X} - \hat{\mathfrak{X}}\| \quad \text{with} \quad \hat{\mathfrak{X}} = \sum_{r=1}^{R} \lambda_r \; \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [\![\boldsymbol{\lambda} \,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!].$$

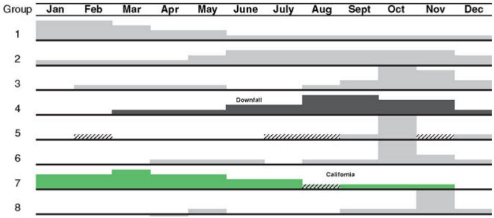
CP Decomposition: Computing Decomposition

1. Simple to understand and implement
2. The potential for numerical ill-conditioning
3. Taking many iterations to converge
4. Not guaranteed to converge to a global minimum or even a stationary point
5. Heavily dependent on the starting guess

CP Decomposition: Application

Discussion tracking in email [Bader et al., 2008]:

1. Term-author-time array
2. $\mathcal{X} \approx \sum_{l=1}^{r} A_l \circ B_l \circ C_l$
3. 25-component decomposition, each rank one tensor refers to a topic



Tucker Decomposition

Tucker decomposition is a form of higher-order PCA. A three-way tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is decomposed as:

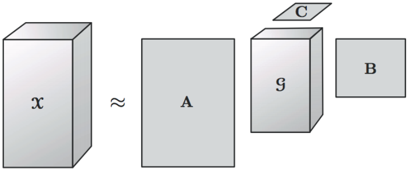
$$\mathbf{\mathcal{X}} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \, \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r = [\![\mathcal{G} \, ; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!].$$

where $A \in \mathbb{R}^{I \times P}$, $B \in \mathbb{R}^{J \times Q}$, $C \in \mathbb{R}^{K \times R}$ are factor matrices which are usually orthogonal and $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ is core tensor.

Tucker Decomposition

Elementwise, the Tucker decomposition is:

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_{ip} b_{jq} c_{kr}$$



Tucker Decomposition

Higher order decomposition:

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} = [\![\mathcal{G} \,; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!]$$

The matricized version:

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)} \mathbf{G}_{(n)} (\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})^{\mathsf{T}}.$$

Tucker Decomposition: Computing Decomposition

Algorithm1:

procedure HOSVD($\mathcal{X}$, R_1, R_2, $\ldots$, R_N)
  for $n = 1, \ldots, N$ **do**
    $\mathbf{A}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{X}_{(n)}$
  end for
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
  return $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

When $R_n < rank_n(\mathcal{X})$, for one or more n, where $rank_n(\mathcal{X})$ is the column rank of $X_{(n)}$, then the decomposition is called truncated HOSVD. Truncated HOSVD is not optimal in terms of giving the best fit as measured by the norm of the difference, but it is a good starting point for ALS.

Tucker Decomposition: Computing Decomposition

Formulation:

$$\min_{\mathcal{G},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}} \left\| \mathcal{X} - [\![\mathcal{G} \,; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \right\|$$
$$\text{subject to } \mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N},$$
$$\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n} \text{ and columnwise orthogonal for } n = 1, \ldots, N.$$

Algorithm2:

procedure HOOI($\mathcal{X}, R_1, R_2, \ldots, R_N$)
  initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$ using HOSVD
  repeat
    for $n = 1, \ldots, N$ **do**
      $\mathcal{Y} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \cdots \times_{n-1} \mathbf{A}^{(n-1)\mathsf{T}} \times_{n+1} \mathbf{A}^{(n+1)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
      $\mathbf{A}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{Y}_{(n)}$
    end for
  until fit ceases to improve or maximum iterations exhausted
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
  return $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

Tucker Decomposition: Application

Classifying hand-written digits [Savas and Eldén, 2007]:

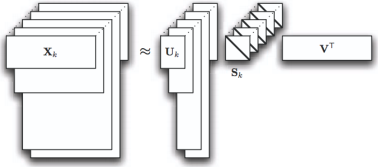
1. $\mathcal{A} = \mathcal{T} \times U \times V \times W$
2. Let $A_v = \mathcal{T}(:,:,v) \times U \times V$, then $\mathcal{A} = \sum_{v=1}^{K} A_v \times w_v$
3. Solve $\min_\alpha ||D - \sum_{v=1}^{k} \alpha_v^\mu A_v^\mu||$

Other Decompositions: PARAFAC2

It is a variant of CP that can be applied to a collection of matrices that each have the same number of columns but a different number of rows. We have a set of matrices X_k for $k = 1, \dots, K$ such that each X_k is of size $I_k \times J$.

$$X_k \approx U_k S_k V^T \quad k = 1, \dots K$$

where U_k is an $I_k \times R$ matrix, S_k is an $R \times R$ diagonal matrix and V is a $J \times R$ factor matrix that does not vary with k.



Toolbox

1. Matlab:
   - Dense multidimensional arrays and elementwise manipulation on tensors
   - External toolboxes: N-way Toolbox, CuBatch, PLSToolbox, Tensor Toolbox
2. Mathematica: sparse tensors
3. Multilinear Engine by Paatero: supports CP, PARAPAC2 and more
4. C++: HUJI Tensor Library, supports inner product, addition, elementwise multiplication

References

Bader, B. W., Berry, M. W., and Browne, M. (2008).
Discussion tracking in enron email using parafac.
Survey of Text Mining II, pages 147–163.

Savas, B. and Eldén, L. (2007).
Handwritten digit classification using higher order singular value decomposition.
Pattern recognition, 40(3):993–1003.

Thank you!