Low rank approximations for tensors

Suraj Kumar

Inria & ENS Lyon

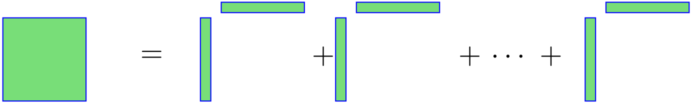
Email:*suraj.kumar@inria.fr*

CR12: September 2023

https://surakuma.github.io/courses/daamtc.html

Recap on Singular Value Decomposition (SVD)

- It decomposes a matrix $A \in \mathbb{R}^{m \times n}$ to the form $U \Sigma V^T$
  - U is an $m \times m$ orthogonal matrix
  - V is an $n \times n$ orthogonal matrix
  - Σ is an $m \times n$ rectangular diagonal matrix
- The diagonal entries $\sigma_i = \Sigma_{ii}$ of Σ are called singular values
  - $\sigma_i \geq 0$ and $\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_{\min(m,n)}$
- The largest r such that $\sigma_r \neq 0$ is called the rank of the matrix
- SVD represents a matrix as the sum of r rank one matrices



Properties of SVD

The SVD of $A \in \mathbb{R}^{m \times n}$ can be written as $A = U \Sigma V^T$. Here $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are othogonal matrices and $\Sigma \mathbb{R}^{m \times n}$ is a rectangular diagonal matrix.

- Columns of U are also eigen vectors of AA^T
- Similarly, columns of V are eigen vectors of $A^T A$
- If $\sigma_i > 0$ is a singular value of A then σ_i^2 is an eigen value of AA^T and $A^T A$

$\Sigma \Sigma^T$ and $\Sigma^T \Sigma$ are diagonal matrices. Their diagonal entries are the eigen values of AA^T and $A^T A$, respectively.

We can also express SVD as

$$A = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1 V_2 \end{pmatrix}^T = U_1 \Sigma_1 V_1^T + U_2 \Sigma_2 V_2^T.$$

This is equivalent to

$$A = U_1 U_1^T A + U_2 U_2^T A = A V_1 V_1^T + A V_2 V_2^T.$$

Properties of matrix Frobenius norm for real matrices

$$||A||_F^2 = \sum_{i,j} A^2(i,j) = Trace(AA^T) = Trace(A^TA)$$

$$||A+B||_F^2 = ||A||_F^2 + ||B||_F^2 + 2\langle A, B\rangle_F$$

Here $\langle A, B\rangle_F$ is known as Frobenius inner product and defined as $\langle A, B\rangle_F = Trace(A^TB) = Trace(B^TA)$.

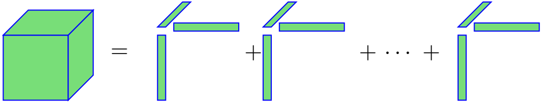
If Q is an orthonormal matrix then,

$$||A||_F^2 = ||QQ^TA||_F^2 + ||(I - QQ^T)A||_F^2.$$

Table of Contents

CP decomposition of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$

It factorizes a tensor into a sum of rank one tensors.



CP decomposition of a 3-dimensional tensor.

$$\mathcal{A} = \sum_{\alpha=1}^{r} U_1(:,\alpha) \circ U_2(:,\alpha) \circ \cdots \circ U_d(:,\alpha)$$

It can be concisely expressed as $\mathcal{A} = [[U_1, U_2, \cdots, U_d]]$. CP decomposition for a 3-dimensional tensor in matricized form can be written as:

$$A_{(1)} = U_1(U_3 \odot U_2)^T, \; A_{(2)} = U_2(U_3 \odot U_1)^T \; A_{(3)} = U_3(U_2 \odot U_1)^T.$$

It is useful to assume that $U_1, U_2 \cdots U_d$ are normalized to length one with the weights given in a vector $\lambda \in \mathbb{R}^r$.

$$\mathcal{A} = [[\lambda; U_1, U_2, \cdots, U_d]] = \sum_{\alpha=1}^{r} \lambda_\alpha U_1(:,\alpha) \circ U_2(:,\alpha) \circ \cdots \circ U_d(:,\alpha)$$

Tensor rank

$$\mathcal{A} = \sum_{\alpha=1}^{r} \lambda_\alpha U_1(:,\alpha) \circ U_2(:,\alpha) \circ \cdots \circ U_d(:,\alpha)$$

- The minimum r required to express $\mathcal{A}$ is called the rank of $\mathcal{A}$

The rank of a real-valued tensor may be different over $\mathbb{R}$ and $\mathbb{C}$. For example, consider the frontal slices of $\mathcal{A} \in \mathbb{R}^{2\times2\times2}$

$$\mathcal{A}(:,:,1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \mathcal{A}(:,:,2) = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

This has rank three over $\mathbb{R}$ and two over $\mathbb{C}$. The CP decomposition over $\mathbb{R}$ has the following factor matrices:

$$U_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix}, U_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \text{ and } U_3 = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix}.$$

The CP decomposition over $\mathbb{C}$ has the following factor matrices:

$$U_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -i & i \end{pmatrix}, U_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}, \text{ and } U_3 = \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}.$$

Rank and low-rank approximations

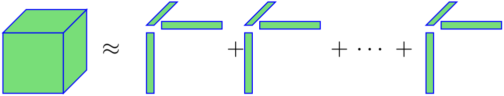
- Determining the rank of a tensor is an NP-complete problem

- If $\mathcal{A} = \sum_{\alpha=1}^{r} \lambda_\alpha U_1(:,\alpha) \circ U_2(:,\alpha) \circ \cdots \circ U_d(:,\alpha)$, summing $k < r$ terms may not yield a best rank-k approximation

- Possible that the best rank-k approximation of a tensor may not exist

Table of Contents

CP optimization problem for a 3-dimensional tensor



For fixed rank k, we want to solve

$$\min_{U_1, U_2 U_3} ||\mathcal{A} - \sum_{\alpha=1}^{k} \lambda_\alpha U_1(:,\alpha) \circ U_2(:,\alpha) \circ U_3(:,\alpha)||.$$

- It is a nonlinear, nonconvex optimization problem
- In the matrix case, the SVD provides us the optimal solution
- In the tensor case, convergence to optimum not guaranteed

Alternating Least Squares (ALS) method

Fixing all but one factor matrix, we have a linear least squares problem:

$$\min_{\hat{U}_1} ||\mathcal{A} - \sum_{\alpha=1}^{k} \hat{U}_1(:,\alpha) \circ U_2(:,\alpha) \circ U_3(:,\alpha)||$$

or equivalently

$$\min_{\hat{U}_1} ||A_{(1)} - \hat{U}_1(U_3 \odot U_2)^T||$$

ALS works by alternating over factor matrices, updating one at a time.

CP-ALS algorithm

Repeat until maximum iterations reached or no further improvement obtained

1. Solve $U_1(U_3 \odot U_2)^T = A_{(1)}$ for $U_1 \Rightarrow U_1 = A_{(1)}(U_3 \odot U_2)(U_3^T U_3 * U_2^T U_2)^\dagger$
2. Normalize columns of U_1
3. Solve $U_2(U_3 \odot U_1)^T = A_{(2)}$ for $U_2 \Rightarrow U_2 = A_{(2)}(U_3 \odot U_1)(U_3^T U_3 * U_1^T U_1)^\dagger$
4. Normalize columns of U_2
5. Solve $U_3(U_2 \odot U_1)^T = A_{(3)}$ for $U_3 \Rightarrow U_3 = A_{(3)}(U_2 \odot U_1)(U_2^T U_2 * U_1^T U_1)^\dagger$
6. Normalize columns of U_3

Here $A^\dagger$ denotes the Moore–Penrose pseudoinverse of A. We use the following identity to get expressions for U_1, U_2 and U_3:

$$(A \odot B)^T (A \odot B) = A^T A * B^T B$$

ALS for computing a CP decomposition

Algorithm 1 CP-ALS method to compute CP decomposition

Require: input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, desired rank k, initial factor matrices $U_j \in \mathbb{R}^{n_j \times k}$ for $1 \leq j \leq d$

Ensure: $[[\lambda; U_1, \cdots, U_d]]$: a rank-k CP decomposition of $\mathcal{A}$

repeat

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

    $V \leftarrow U_1^\mathsf{T} U_1 * \cdots * U_{i-1}^\mathsf{T} U_{i-1} U_{i+1}^\mathsf{T} U_{i+1} * \cdots * U_d^\mathsf{T} U_d$

    $U_i \leftarrow A_{(i)}(U_d \odot \cdots \odot U_{i+1} \odot U_{i-1} \odot U_1)$

    $U_i \leftarrow U_i V^\dagger$

    $\lambda \leftarrow$ normalize colums of U_i

  end for

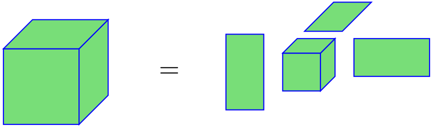
until converge or the maximum number of iterations

- U_j can be chosen randomly or by setting k left singular vectors of $A_{(j)}$ for $1 \leq j \leq d$

Table of Contents

Tucker decomposition of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$

It represents a tensor with d matrices (usually orthonormal) and a small core tensor.



Tucker decomposition of a 3-dimensional tensor.

$$\mathcal{A} = \mathcal{G} \times_1 U_1 \cdots \times_d U_d$$

$$\mathcal{A}(i_1, \cdots, i_d) = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_d=1}^{r_d} \mathcal{G}(\alpha_1, \cdots, \alpha_d) U_1(i_1, \alpha_1) \cdots U_d(i_d, \alpha_d)$$

It can be concisely expressed as $\mathcal{A} = [[\mathcal{G}; U_1, \cdots, U_d]]$.

Here r_j for $1 \leq j \leq d$ denote a set of ranks. Matrices $U_j \in \mathbb{R}^{n_j \times r_j}$ for $1 \leq j \leq d$ are usually orthonormal and known as factor matrices. The tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}$ is called the core tensor.

Table of Contents

High Order SVD (HOSVD) for computing a Tucker decomposition

Algorithm 2 HOSVD method to compute a Tucker decomposition

Require: input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, desired rank $(r_1, \cdots, r_d)$
Ensure: $\mathcal{A} = \mathcal{G} \times_1 U_1 \times_2 U_2 \cdots \times_d U_d$
  for $k = 1$ to d **do**
    $U_k \leftarrow r_k$ leading left singular vectors of $A_{(k)}$
  end for
  $\mathcal{G} = \mathcal{A} \times_1 U_1^T \times_2 U_2^T \cdots \times_d U_d^T$

- When $r_i < rank(A_{(i)})$ for one or more i, the decomposition is called the truncated-HOSVD (T-HOSVD)
- Output of T-HOSVD can be used as a starting point for an ALS algorithm

Sequentially T-HOSVD (ST-HOSVD) for Tucker decomposition

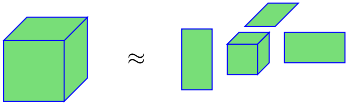
- This method is more work efficient than T-HOSVD
- In each step, it reduces the size of one dimension of the tensor

Algorithm 3 ST-HOSVD method to compute a Tucker decomposition

Require: input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, desired rank $(r_1, \cdots, r_d)$
Ensure: $[[\mathcal{G}; U_1, \cdots, U_d]]$: a $(r_1, \cdots, r_d)$-rank Tucker decomposition of $\mathcal{A}$

  $\mathcal{B} \leftarrow \mathcal{A}$
  for $k = 1$ to d **do**
    $S \leftarrow B_{(k)} B_{(k)}^T$
    $U_k \leftarrow r_k$ leading eigen vectors of S
    $\mathcal{B} \leftarrow \mathcal{B} \times_i U_i$
  end for
  $\mathcal{G} = \mathcal{A} \times_1 U_1^\mathsf{T} \times_2 U_2^\mathsf{T} \cdots \times_d U_d^\mathsf{T}$

Tucker decomposition optimization problem for a 3-dimensional tensor



For fixed ranks orthonormal matrices U_1, U_2, U_3, we want to solve

$$\min_{U_1, U_2, U_3} ||\mathcal{A} - \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3||, \text{ where } \mathcal{G} = \mathcal{A} \times_1 U_1^T \times_2 U_2^T \times_3 U_3^T.$$

This is equivalent to

$$\max_{U_1, U_2, U_3} ||\mathcal{A} \times_1 U_1^T \times_2 U_2^T \times_3 U_3^T||.$$

It is a nonlinear, nonconvex optimization problem.

Higher-order orthogonal iteration (HOOI) method

Fixing all but one factor matrix, we have a matrix problem:

$$\max_{\hat{U}_1} ||\mathcal{A} \times_1 \hat{U}_1{}^T \times_2 U_2^T \times_3 U_3^T||$$

HOOI works by alternating over factor matrices, updating one by computing left singular vectors

HOOI method for computing a Tucker decomposition

Algorithm 4 HOOI method to compute Tucker decomposition

Require: input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, desired ranks $(r_1, \cdots, r_d)$, initial factor matrices $U_j \in \mathbb{R}^{n_j \times r_j}$ for $1 \leq j \leq d$

Ensure: $[[\mathcal{G}; U_1, \cdots, U_d]]$: a $(r_1, \cdots, r_d)$-rank Tucker decomposition of $\mathcal{A}$

repeat

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

    $\mathcal{B} \leftarrow \mathcal{A} \times_1 U_1^T \cdots \times_{i-1} U_{i-1}^T \times_{i+1} U_{i+1}^T \cdots \times_d U_d^T$

    $U_i \leftarrow r_i$ left singular vectors of $B_{(i)}$

  end for

until converge or the maximum number of iterations

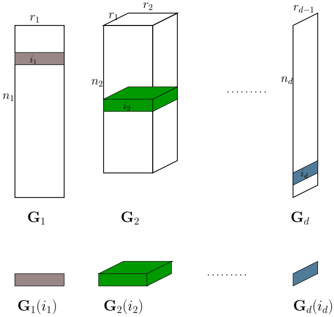
$\mathcal{G} \leftarrow \mathcal{A} \times_1 U_1^T \times_2 U_2^T \cdots \times_d U_d^T$

- Outputs of HOSVD (U_j for $1 \leq j \leq d$) can be used as a starting point for this method

Table of Contents

Tensor Train (TT) decomposition: Product of matrices view

- A d-dimensional tensor is represented with 2 matrices and d-2 3-dimensional tensors.



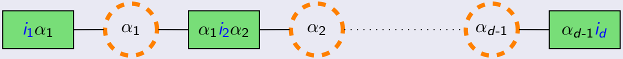
$$\mathbf{A}(i_1, i_2, \cdots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d)$$

An entry of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is computed by multiplying corresponding matrix (or row/column) of each matrix/tensor.

Tensor Train decomposition

$\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is represented with cores $\mathcal{G}_k \in \mathbb{R}^{r_{k-1}\times n_k\times r_k}$, $k=1,2,\cdots d$, $r_0=r_d=1$ and its elements satisfy the following expression:

$$\begin{aligned}\mathcal{A}(i_1,\cdots,i_d) &= \sum_{\alpha_0=1}^{r_0}\cdots\sum_{\alpha_d=1}^{r_d}\mathcal{G}_1(\alpha_0,i_1,\alpha_1)\cdots\mathcal{G}_d(\alpha_{d-1},i_d,\alpha_d)\\ &= \sum_{\alpha_1=1}^{r_1}\cdots\sum_{\alpha_{d-1}=1}^{r_{d-1}}\mathcal{G}_1(1,i_1,\alpha_1)\cdots\mathcal{G}_d(\alpha_{d-1},i_d,1)\end{aligned}$$



The ranks r_k are called TT-ranks.

- The number of entries in this decomposition = $\mathcal{O}(n_1r_1 + n_2r_1r_2 + n_3r_2r_3 + \cdots + n_{d-1}r_{d-2}r_{d-1} + n_dr_{d-1})$