CSE 392: Matrix and Tensor Algorithms for Data Spring 2024

Lecture 19 — 03-27

Instructor: Shashanka Ubaru *Scribe: Gabriel H. Brown*

1 The Tucker Decomposition

If a tensor $\mathcal{T} \in \mathbb{R}^{m \times n \times p}$ is multilinear rank (k_1, k_2, k_3) then it has a so called Tucker decomposition

$$T = \mathcal{G} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} \quad \text{where} \quad \mathcal{G} \in \mathbb{R}^{k_1 \times k_2 \times k_3},\ \boldsymbol{A} \in \mathbb{R}^{m \times k_1},\ \boldsymbol{B} \in \mathbb{R}^{n \times k_2},\ \boldsymbol{C} \in \mathbb{R}^{p \times k_3}. \tag{1}$$

Like the CP decomposition, the following shorthand is sometimes used to denote a tensor decomposed in the Tucker format: $\llbracket \mathcal{G}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \rrbracket$.

Note that the Tucker decomposition is nonunique; *proof*: let $\tilde{\mathcal{G}} \coloneqq \mathcal{G} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{W}$ for invertible matrices $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$, now note that $\mathcal{T} = \llbracket \mathcal{G}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \rrbracket = \llbracket \tilde{\mathcal{G}}; \boldsymbol{A}\boldsymbol{U}^{-1}, \boldsymbol{B}\boldsymbol{V}^{-1}, \boldsymbol{C}\boldsymbol{W}^{-1} \rrbracket$. In theory, this allows freedom to choose factor matrices such that the core tensor has many nonzeros. However, in practice the most common simplification is to assume that the factor matrices are orthogonal; the proof follows the one for nonuniqueness, but uses the ability to QR factorize any matrix.

The best multilinear rank (k_1, k_2, k_3) Tucker decomposition problem is defined by

$$\min_{\mathcal{G}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}} \|\mathcal{T} - \llbracket \mathcal{G}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \rrbracket\|_F^2. \tag{2}$$

In general, this problem is also NP-hard, however many of the algorithms for computing a Tucker approximation have guarantees on the quality of their solution. In another deviation from CP approximations, a best Tucker/multilinear rank approximation always exists.

Theorem 1. *For all choices of multilinear ranks (k_1, k_2, k_3) the tensor $\mathcal{T} \in \mathbb{R}^{m \times n \times p}$ has a best approximant in $\{\mathcal{X} : \mu\text{-rank}(\mathcal{X}) \le (k_1, k_2, k_3)\}$ where μ-rank denotes multilinear rank and the inequality between tuples holds elementwise.*

Proof. It suffices to prove that the set $\{\mathcal{X} : \mu\text{-rank}(\mathcal{X}) \le (k_1, k_2, k_3)\}$ is closed. Furthermore, this set may be written as $\{\mathcal{X} : \mu\text{-rank}(\mathcal{X})_1 \le k_1\} \cap \{\mathcal{X} : \mu\text{-rank}(\mathcal{X})_2 \le k_2\} \cap \{\mathcal{X} : \mu\text{-rank}(\mathcal{X})_3 \le k_3\}$ Since the intersection of closed sets is closed, it further suffices to prove that $\{\mathcal{X} : \mu\text{-rank}(\mathcal{X})_1 \le k_1\}$ is closed.
Now this set is equivalent to the set of $\{\boldsymbol{A} \in \mathbb{R}^{m \times np} : \text{rank}(\boldsymbol{A}) \le k_1\}$ which is closed as a classical result in algebraic geometry. ■

2 Tucker-ALS

An algorithm similar to CP-ALS may also be used for computing an approximate Tucker decomposition. First note that the three unfoldings of $\llbracket \mathcal{G}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \rrbracket$ are

$$\boldsymbol{A}\boldsymbol{G}_{(1)}(\boldsymbol{C} \otimes \boldsymbol{B})^\top,\ \boldsymbol{B}\boldsymbol{G}_{(2)}(\boldsymbol{C} \otimes \boldsymbol{A})^\top,\ \boldsymbol{C}\boldsymbol{G}_{(3)}(\boldsymbol{B} \otimes \boldsymbol{A})^\top. \tag{3}$$

Tucker-ALS consists of repeatedly executing the following steps until convergence

- $\boldsymbol{A} \leftarrow argmin_{\boldsymbol{A}} \left\| (\boldsymbol{C} \otimes \boldsymbol{B})\, \boldsymbol{G}_{(1)}^{\top} \boldsymbol{A}^{\top} - \boldsymbol{T}_{(1)}^{\top} \right\|_F^2$
- $\boldsymbol{B} \leftarrow argmin_{\boldsymbol{B}} \left\| (\boldsymbol{C} \otimes \boldsymbol{A})\, \boldsymbol{G}_{(2)}^{\top} \boldsymbol{B}^{\top} - \boldsymbol{T}_{(2)}^{\top} \right\|_F^2$
- $\boldsymbol{C} \leftarrow argmin_{\boldsymbol{C}} \left\| (\boldsymbol{B} \otimes \boldsymbol{A})\, \boldsymbol{G}_{(3)}^{\top} \boldsymbol{C}^{\top} - \boldsymbol{T}_{(3)}^{\top} \right\|_F^2$
- $\mathcal{G} \leftarrow \arg\min_{\mathcal{G}} \|(\boldsymbol{C} \otimes \boldsymbol{B} \otimes \boldsymbol{A})\mathrm{vec}(\mathcal{G}) - \mathrm{vec}(\mathcal{T})\|_2^2$.

As the name suggests, each of these are (linear) least squares problems. In practice this algorithm is not commonly used due to its high cost (the last step alone has complexity $\mathcal{O}((k_1k_2k_3)^2mnp)$). Furthermore, like CP-ALS, many local minima may exist and the quality of the solution has no guarantees. As we shall see shortly, there are faster methods with better guarantees.

3 HOSVD-type algorithms

The HOSVD algorithm computes the exact Tucker format representation of any tensor. The key idea is to use the left singular vectors of the SVDs of the matricizations. The algorithm consists of four steps to compute an exact representation of $\mathcal{T}$:

- $\boldsymbol{U}, *, * \leftarrow \texttt{svd}(\boldsymbol{T}_{(1)})$
- $\boldsymbol{V}, *, * \leftarrow \texttt{svd}(\boldsymbol{T}_{(2)})$
- $\boldsymbol{W}, *, * \leftarrow \texttt{svd}(\boldsymbol{T}_{(3)})$
- $\mathcal{G} \leftarrow \mathcal{T} \times_1 \boldsymbol{U}^{\top} \times_2 \boldsymbol{V}^{\top} \times_3 \boldsymbol{W}^{\top}$

Two questions are immediate:

- Does this provide compression for a general tensor $\mathcal{T}$?
- what if we replace the calls to `svd` with respective calls to the truncated matrix SVD function `tsvd` and ranks (k_1, k_2, k_3) corresponding to our desired approximation rank?

The answer to the first question is clearly no, generic tensors have full multilinear rank and so the core tensor would be the same size as $\mathcal{T}$. The "answer" to the second question is the so called truncated HOSVD (T-HOSVD) algorithm, which is extremely similar to the HOSVD algorithm but instead solves the best multilinear rank (k_1, k_2, k_3) problem.

- $\boldsymbol{U}, *, * \leftarrow \texttt{tsvd}(\boldsymbol{T}_{(1)}, k_1)$
- $\boldsymbol{V}, *, * \leftarrow \texttt{tsvd}(\boldsymbol{T}_{(2)}, k_2)$
- $\boldsymbol{W}, *, * \leftarrow \texttt{tsvd}(\boldsymbol{T}_{(3)}, k_3)$
- $\mathcal{G} \leftarrow \times_1 \boldsymbol{U}^{\top} \times_2 \boldsymbol{V}^{\top} \times_3 \boldsymbol{W}^{\top}$

We stress that while T-HOSVD is designed to solve the multilinear rank (k_1, k_2, k_3) approximation problem, it by no means provides an optimal solution (this is a polynomial time algorithm for an NP-hard problem). However, it intuitively provides a reasonable solution to the problem and we shall see shortly that its solution is what is called "quasi-optimal".

An algorithm related to T-HOSVD is the so called sequentially truncated HOSVD algorithm. Instead of computing 3 matrices of left singular vectors independently, we instead compress the representation of the tensor directly after computing a left singular vector matrix. The ST-HOSVD algorithm is:

- $\boldsymbol{U}, *, * \leftarrow \texttt{tsvd}(\boldsymbol{T}_{(1)}, k_1)$
- $\mathcal{G} \leftarrow \mathcal{T} \times_1 \boldsymbol{U}^\top$
- $\boldsymbol{V}, *, * \leftarrow \texttt{tsvd}(\boldsymbol{G}_1, k_2)$
- $\mathcal{G} \leftarrow \mathcal{G} \times_2 \boldsymbol{V}^\top$
- $\boldsymbol{W}, *, * \leftarrow \texttt{tsvd}(\boldsymbol{T}_{(3)}, k_3)$
- $\mathcal{G} \leftarrow \mathcal{G} \times_3 \boldsymbol{W}^\top$

Note that $\mathcal{G}$ is only the correct/expected shape after the final step. A few important facts about the ST-HOSVD are as follows:

- the worst case error bound is the same as T-HOSVD
- the modes may be visited in any order; the algorithm above chooses 1-2-3, but depending on the shape of the tensor alternative orderings will reduce the total complexity
- the ST-HOSVD has a lower complexity than T-HOSVD, but is also more serial in nature since the result of the SVDs are now dependent on previous SVDs.

An extremely attractive feature of the HOSVD, T-HOSVD, and ST-HOSVD algorithms is that they are quasi-optimal.

Theorem 2. *(Vannieuwenhoven et al., Hackbusch) Suppose tensor $\mathcal{T} \in \mathbb{R}^{m\times n\times p}$ has optimal Tucker approximant is $\mathcal{T}^*$ and the (T-/ST-)HOSVD algorithm returns tensor $\tilde{\mathcal{T}}$, then*

$$\|\mathcal{T} - \mathcal{T}^*\|_F \leq \|\mathcal{T} - \tilde{\mathcal{T}}\|_F \leq \sqrt{3}\|\mathcal{T} - \mathcal{T}^*\|_F$$

4 HOOI algorithm

We conclude by discussing an algorithm of a different nature than *HOSVD. The higher order orthogonal iteration algorithm iterates the following three steps until convergence:

- $\boldsymbol{U} \leftarrow \texttt{tsvd}(\mathcal{T} \times_2 \boldsymbol{V}^\top \times_3 \boldsymbol{W}^\top, k_1)$
- $\boldsymbol{V} \leftarrow \texttt{tsvd}(\mathcal{T} \times_1 \boldsymbol{U}^\top \times_3 \boldsymbol{W}^\top, k_2)$

- $\boldsymbol{W} \leftarrow \texttt{tsvd}(\mathcal{T} \times_1 \boldsymbol{U}^\top \times_2 \boldsymbol{V}^\top, k_3)$

Once convergence is reached, the core tensor is computed as $\mathcal{G} = \mathcal{T} \times_1 \boldsymbol{U}^\top \times_2 \boldsymbol{V}^\top \times_3 \boldsymbol{U}^\top$. As an iterative algorithm, HOOI requires an initialization for $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$. In practice HOOI is expensive, but converges in relatively few iterations.