zbMATH Open
THE FIRST RESOURCE FOR MATHEMATICS

Ballard, Grey; **Kolda, Tamara G.**
Tensor decompositions for data science. (English) Zbl 08031854
Cambridge: Cambridge University Press (ISBN 978-1-00-947167-1/hbk; 978-1-00-947166-4/ebook). xviii, 404 p. (2025).

This is a first course on tensors for data science students who are supposed to be familiar with basics from linear algebra, optimization and statistics. Summaries of the necessary prerequisites are added in appendices.

The concept of a tensor as a d-dimensional table of numbers is not difficult and generalizes the case $d = 1$ (vector) and $d = 2$ (matrix). Hence some issues are familiar but become more complicated as soon as $d \geq 3$. Many of the generalizations are first introduced in the simplest case $d = 3$ and then generalized to a more general d.

The position of an element is characterized by d indices $(i_1, \ldots, i_d)$. While a matrix has rows and columns, a tensor has fibers (a mode-k fiber is a one dimensional subsets corresponding to varying only the index i_k) or slices (k-dimensional subsets by varying only k indices with $2 \leq k \leq d-1$).

Elements of the tensor can be enumerated in several possible ways. A natural order is by enumerating all possible indices such that the numbers $n = i_d i_{d-1} \cdots i_1$ are in increasing order. This gives a long vectorized list of its elements consisting of all the mode-1 fibers. In a matricized version, these mode-1 fibers are arranged as columns of a matrix, called a a mode-1 unfolding. However one may permute the order of the indices and for example place index i_k at the end, keeping all the others in natural order, giving a mode-k unfolding. Or, more generally, one may select a subset of indices that are placed at the end after their complementary set. This corresponds to reshaping the tensor and rearranging it into a sequence of different subtensors. Enumerating the elements in that order corresponds to reshaping the tensor from a d-dimensional object into a one dimensional long vector and it requires index functions that allows to switch between the tensor multiindex $(i_1, i_2, \ldots, i_d)$ and for example the position of that element in a vectorized enumeration.

Next are the operations that can be performed with tensors. The inner product of two tensors is equivalent with the inner product of any of their vectorized forms. Like for vectors there are outer and Kronecker products of tensors and tensor times matrix (TTM) products that need to be defined. The latter means multiply each mode-k fiber (i.e., the mode-k unfolding) with the matrix. A particular useful product is the matricized tensor times Khatri-Rao product (MTTKRP). The rows of a KR product $A \odot B$ of matrices A and B consist of Hadamard products of rows of A and rows of B. How all this can be implemented is summarized in practical algorithms, often requiring to rearrange the elements of the tensor (reshaping the tensor).

The larger part of the book is devoted to to the main topic: decompositions of tensors. This is a way to find efficient representations that avoid the curse of dimensionality. Most of the algorithms to compute these decompositions depend in one way or another on SVD computations from linear algebra and they try to generalize the SVD idea of writing a matrix as a sum of the outer product of 2 vectors. The first is the Tucker decomposition in which a tensor $\mathcal{X}$ is (approximated by) a (small) kernel tensor $\mathcal{G}$ that is lifted by TTMs along the different modes to a full tensor approximating $\mathcal{X}$. To find a good (or best) Tucker decomposition an optimization (least squares) problem needs to be solved. There are several possible versions in which one has to balance between efficiency (compression ratio) amd accuracy. One way is to fix the size of the kernel as a generalization for fixing the rank k in SVD approximation of matrices Several optimization procedures and the associated error estimates are given.

In a relatively short chapter, a general tensor train decomposition is described compressing a tensor by writing it as a product (at most d) lower order tensors.

More important is a canonical polyadic decomposition (CP) also known as CANDECOMP or PARAFAC decomposition. It is discussed most extensively. It is another way to generalize the SVD composition which is closest to the original matrix SVD. The tensor is written as the sum of the outer product of d fibers (i.e. vectors). If the fibers are normalized, this is compensated by a weight, which is the generalization of the associated singular value. For $d = 2$ the number of terms is bounded by the rank of the matrix

© 2025 FIZ Karlsruhe GmbH

but for $d \geq 3$, the concept of rank is less trivial (and in fact NP-hard to compute). Also uniqueness is not guaranteed and depends on the independency of the fibers involved. Again there are several ways to compute a CP decomposition and it usually involves again a (weighted) least squares optimization problem. When stored in this CP decomposed form, this is called a Kruskal tensor and all the basic operations for tensors can be performed efficiently when it has the Kruskal form. Also the optimization algorithms to generate the CP decomposition are extensively discussed (gradient, Gauss-Newton, and others) with special attention for the case of sparse data and alternative loss functions to be optimized. The introduction also describes 4 practical open source data sets and the effect of the CP decomposition approximations are illustrated for some of these.

A short chapter comments on some loose ends: orthogonality, symmetry, rank computation, connection between Tucker and CP decomposition, etc.

The authors give proposals on how to compose a one semester course on the material of this book. The emphasis is definitely on the computational aspects of the manipulation and the decompositions and not on the applications. There are no systematic lists of exercises at the end of chapters, but there are some isolated exercises throughout the text. The layout is the classical one with definitions an theorems standing out in coloured frames. Also the decompositions and multiplications are amply illustrated in coloured 3D graphics which are very useful to help thinking in 3D and by extension in d dimensions.

Reviewer: Adhemar Bultheel (Leuven)

MSC:

68-01	Introductory exposition (textbooks, tutorial papers, etc.) pertaining to computer science	Cited in **1** Document
15-01	Introductory exposition (textbooks, tutorial papers, etc.) pertaining to linear algebra	
68T09	Computational aspects of data analysis and big data	

Keywords:

tensor decomposition; data science; SVD; Tucker decomposition; Kruskal tensor; canonical polyadic decomposition; CANDECOMP; PARAFAC

Software:

TensorToolbox; pyttb

Full Text: DOI

© 2025 FIZ Karlsruhe GmbH