Universität des Saarlandes

Fachrichtung 6.1 – Mathematik

Preprint

The Three-Way Decomposition

Ilghiz Ibraghimov

Preprint No. 66
Saarbrücken 2002

Universität des Saarlandes

Fachrichtung 6.1 – Mathematik

The Three-Way Decomposition

Ilghiz Ibraghimov

Saarland University
Department of Mathematics
Postfach 15 11 50
D–66041 Saarbrücken
Germany
E-Mail: ilgis@num.uni-sb.de

submitted: July 23, 2002

Preprint No. 66

Saarbrücken 2002

Edited by
FR 6.1 – Mathematik
Im Stadtwald
D–66041 Saarbrücken
Germany

Fax: + 49 681 302 4443
e-mail: preprint@math.uni-sb.de
WWW: http://www.math.uni-sb.de/

Abstract

In this article we discuss the decomposition of $A_k \in \mathbb{R}^{n_1 \times n_2}, k = 1, \ldots, n_3$, as $A_k \simeq BE\hat{D}_k C^*$ in the Frobenius norm, where $B \in \mathbb{R}^{n_1 \times r}$ and $C \in \mathbb{R}^{n_2 \times r}$ have normalized columns, E and $\hat{D}_k \in \mathbb{R}^{r \times r}$ are diagonal and $\sum_{k=1}^{n_3} \hat{D}_k^2$ is the identity matrix. This decomposition is widely used in the data processing and is the generalization of the singular value decomposition for the 3 dimensional case. We propose a new algorithm for finding B, C, $\hat{D}_k$ and E if A_k and r are given and B, C have full column rank. If A_k have exact decomposition then this algorithm has a linear convergence.

An implementation of the numerical algorithm was developed, several examples were tested and good results obtained.

1 Introduction

This paper gives a method for identifying the parameters of a multilinear model proposed by Harshman [1] and by Carroll and Chang [2]. In this problem we have an n_1-by-n_2-by-n_3 array $A_k \in \mathbb{R}^{n_1 \times n_2}, k = 1, \ldots, n_3$. The idea is to find *factors* $B \in \mathbb{R}^{n_1 \times r}$, $C \in \mathbb{R}^{n_2 \times r}$, $D \in \mathbb{R}^{n_3 \times r}$ and diagonal $E \in \mathbb{R}^{r \times r}$, where r is as small as possible and B, C, D have unit l_2 norm columns, so that

$$A_k = BE\hat{D}_k C^*,$$

where $\hat{D}_k \in \mathbb{R}^{r \times r}$ are diagonal and contain k-th row of D on the diagonal. Further, for the simplicity of explanation, we will use both $\hat{D}_k$ and D for the same data.

In the real applications $A_k = BE\hat{D}_k C^* + R_k$, where R_k represents some noise of the experiment. We say that the data have $z\%$ of noise if

$$\sqrt{\frac{\sum_{k=1}^{n_3} ||R_k||_F^2}{\sum_{k=1}^{n_3} \left(||B\hat{D}_k C^*||_F^2 + ||R_k||_F^2 \right)}} = \frac{z}{100}.$$

We discuss a general case where we have no any a priory information about R_k, then we will minimize the sum of their Frobenius norms as following:

$$\min_{B,C,E,\hat{D}_1,\ldots,\hat{D}_{n_3}} \sum_{k=1}^{n_3} ||A_k - B\hat{D}_k C^*||_F^2. \tag{1}$$

This decomposition is a generalization of Singular Value Decomposition (SVD) on three dimensional case. In SVD there are unique orthonormal B and C (we skip the case with repeated singular values), so that $A = BEC^*$. In the Three-Way Decomposition (TWD) the condition of orthonormality for B, C and D is dropped. It was proved by Kruskal [5], who gives such conditions on the sets of parameters that the parameters are identifiable and unique, but his paper does not include a construction.
Several successful applications have been demonstrated in quite different areas such as psychometrics [1, 2], chromatography [3], optical spectroscopy [4], number theory [5] and NMR [6]. Noise is zero in some applications like [5] and 30% in others [6], the size of problem changes from $r = 2$ till $r = 5$ in chromatography to really huge problems with $r > 100$ for NMR.
Harshman and Lundy [7] suggested a monotonically converging algorithm called `PARAFAC` for minimizing the distance between A_k and $BE\hat{D}_kC^*$ in Frobenius norm. Freeze any two of B, C and D, and the functional is linear in the third. For example, if we temporarily fix the matrices C and D, then we seek $\tilde{B}$ so

$$\sum_{k=1}^{n_3} ||A_k - \tilde{B}\hat{D}_kC^*||_F^2$$

is minimized. The new B and E are then defined by $\tilde{B} = BE$, where B's columns have unit l_2 norms. It is easy to show that $\tilde{B}$ solves the ordinary least squares problem

$$\min_{\tilde{B}} ||[A_1, \ldots, A_{n_3}] - \tilde{B}[\hat{D}_1C^*, \ldots, \hat{D}_{n_3}C^*]||_F^2.$$

This approach has poor convergence properties, sometimes it runs in local minima [8] or/and needs an incredibly large amount of iterations. In the `PARAFAC` algorithm r should be given.
If r is much smaller than n_1, n_2 or n_3, then, for given r_1, r_2 and r_3:

$$\begin{array}{llll} B = B'U_B, & B' \in \mathbb{R}^{n_1 \times r_1}, & U_B \in \mathbb{R}^{r_1 \times r}, & r \le r_1 \le n_1, \\ C = C'U_C, & C' \in \mathbb{R}^{n_2 \times r_2}, & U_C \in \mathbb{R}^{r_2 \times r}, & r \le r_2 \le n_2, \\ D = D'U_D, & D' \in \mathbb{R}^{n_3 \times r_3}, & U_D \in \mathbb{R}^{r_3 \times r}, & r \le r_3 \le n_3. \end{array}$$

We can compute B', C', D' and U_B, U_C, U_D independently. This idea was originally introduced by Tucker [9] and improved by Kronenberg and de Leeuw [10]. Consider

$$G_{k'} = U_B E \hat{U}_{D_{k'}} U_C^*, \qquad k' = 1, \ldots, r_3,$$

here $\hat{U}_{D_{k'}}$ created from U_D the same way as $\hat{D}_k$ from D, and assume that B', C', D' have orthonormal columns, and the Tucker algorithm is searching for

$$\min_{B',C',D',G_{k'}} \sum_{k=1}^{n_3} \left|\left| A_k - B' \left(\sum_{k'=1}^{r_3} G_{k'} d_{kk'} \right) C'^* \right|\right|_F^2, \tag{2}$$

where $d_{kk'}$ is the (k,k') entry of D'. $G_{k'}$ is called core array and U_B, U_C, U_D can be computed again from TWD of the core. In general, if (2) is bigger than zero — we call this case as the initial array A_k has some noise — then `Tucker` reduction does not preserve Frobenius norm, but in most applications it works good enough [11].

The algorithm that computes the core is similar to `PARAFAC`: freeze any two of B', C' and D', and the functional is linear in the third. For example, if we temporarily fix B' and C' matrices, then we seek $\tilde{D}$ and, as B' and C' are orthonormal, it leads to

$$\min_{D',G_{k'}} \sum_{k=1}^{n_3} \left|\left| B'^* A_k C' - \left(\sum_{k'=1}^{r_3} G_{k'} d_{kk'} \right) \right|\right|_F^2.$$

If we transform all elements of matrix $B'^* A_k C' \in \mathbb{R}^{r_1 \times r_2}$ to the vector $p_k \in \mathbb{R}^{r_1 r_2}$ and do the same for $G_{k'} \in \mathbb{R}^{r_1 \times r_2}$ (transform it to $q_{k'} \in \mathbb{R}^{r_1 r_2}$), then this problem becomes

$$\min_{D',q_{k'}} \sum_{k=1}^{n_3} \left|\left| [p_1, \ldots, p_{n_3}] - [q_1, \ldots, q_{r_3}] D'^* \right|\right|_F^2,$$

which is an ordinary least squares problem and D' contains the right singular vectors of the matrix $[p_1, \ldots, p_{n_3}]$. In the `Tucker` algorithm r_1, r_2 and r_3 should be given.

The main advantage of this algorithm is a better convergence: during computations of the core it is usually better than in `PARAFAC` because B', C' and D' can be chosen orthonormal. For example, if the initial data have no noise at all it converges within 3 iterations [8], but this algorithm still has just a monotonical convergence for the general case. One iteration of the `PARAFAC` algorithm linearly depends on the size of initial problem. If $r_1 r_2 r_3$ is reasonably smaller than $n_1 n_2 n_3$, then TWD with the `Tucker` approach works much faster [11].

A very important problem in nonlinear minimization is to compute an initial approximation. This field was discussed by Leurgans, Ross and Abel [12]. The most popular initial approaches are based either on one iteration of the `Tucker` algorithm or on the generalized eigenvalue decomposition of A_1 and A_2.

Thus, up to now, TWD has been solved by the PARAFAC + Tucker algorithm. The arithmetical complexity of three sequential iterations for Tucker is $n_1n_2n_3(n_1+n_2+n_3)$ and for PARAFAC $3r_1r_2r_3r$ operations. Usually the Tucker algorithm has good convergence [11] and for most cases it converges within $10-100$ iterations, while convergence of PARAFAC is really weak, thus it needs more than 10000 iterations for the problem with $r=30$ and usually does not converge at all for the problems with $r>100$ [6].
In this work we present a new algorithm for TWD with linear convergence if B and C have a full column rank. It dramatically reduces the computational time and produces no worse accuracy. The main idea is to decompose our problem into several small nonlinear eigenvector problems and solve them by standard minimization methods with good convergence.

2 The Algorithm

We introduce the following notations, Let $b_l \in \mathbb{R}^{n_1}$, $c_l \in \mathbb{R}^{n_2}$, $d_l \in \mathbb{R}^{n_3}$ are l-th columns of B, C and D; $E = \mathtt{diag}(e_1,\ldots,e_r)$, $e_l > 0$, $l = 1,\ldots,r$. We call b_l, c_l, d_l and e_l as the triad.

Theorem 1. *If exact solution of three-way decomposition of n_1-by-n_2-by-n_3 array $A_k, k=1,\ldots,n_3$ contains B and C with full column rank r, then it is possible to transform this problem to another three-way decomposition r-by-r-by-n_3 problem G_k with no more than $n_1n_2n_3(n_1+n_2)+n_1^3+n_2^3$ arithmetical operations.*

Proof. Let B' and C' are the left singular vectors of $Z_1 = [A_1,\ldots,A_{n_3}]$ and $Z_2 = [A_1^*,\ldots,A_{n_3}^*]$ correspondently. Because the original A_k array has exact decomposition with only r rank, then the SVD of Z_1 and Z_2 have only r nonzero singular values, then B' and C' contains the singular vectors belongs to the nonzero singular values. If we set the D' as the identity matrix, then (2) is zero. The SVD of Z_1 and Z_2 require no more than $n_1n_2n_3(n_1+n_2)+n_1^3+n_2^3$ arithmetical operations. ■

Theorem 2. *Suppose that the n_1-by-n_2-by-n_3 array $A_k, k=1,\ldots,n_3$, has an exact three-way decomposition with two quadratic nonsingular factors B and C. Then there is an algorithm to compute independently at least one triad.*

Proof. Because B and C are quadratic nonsingular matrices, there exist $S = B^{-*}$ and $T = C^{-*}$. Our problem is to find B, C, $\hat{D}_k$, E when $A_k = B\hat{D}_kEC^*, k=1,\ldots,n_3$, or $A_kTE^{-1} = B\hat{D}_k$, or $A_k^*SE^{-1} = C\hat{D}_k$. The last two problems can be splitted to r small nonlinear problems:

$$At_l e_l^{-1} = b_l \otimes d_l, \tag{3}$$
$$A' s_l e_l^{-1} = c_l \otimes d_l, \tag{4}$$

where $A = [A_1^*, \ldots, A_n^*]^*$, $A' = [A_1, \ldots, A_n]^*$, $\otimes$ denotes the Kronecker product of two vectors, b_l, c_l, d_l, s_l and t_l are l-th columns of B, C, D, S and T; e_l is the (l, l) entry of E.

By definition all t_l depend on all c_l, and the same holds for s_l and b_l. As A_k has an exact three-way decomposition then At_l and $A's_l$ can be transferred to matrices $r \times n$ with rank 1. Then we can solve (3) and (4) simultaneously. Let us perform a singular vector decomposition for A and A'. Then $A = V\Lambda_1 X^*$, and $A' = W\Lambda_2 Y^*$ and

$$\Lambda_1 X^* t_l e_l^{-1} = V^*(b_l \otimes d_l), \qquad \Lambda_2 Y^* s_l e_l^{-1} = W^*(c_l \otimes d_l). \tag{5}$$

As $||b_l||_2 = ||c_l||_2 = ||d_l||_2 = 1$, we can solve (3) and (4) by minimizing $F_1 = 1 - ||V^*(b_l \otimes d_l)||_2^2$ and $F_2 = 1 - ||W^*(c_l \otimes d_l)||_2^2$. These functions are nonnegative because they are transferred from the least squares problems, thus we obtain the solution when

$$\min_{b,c,d} ||V^*(b_l \otimes d_l)||_2^2 + ||W^*(c_l \otimes d_l)||_2^2 \tag{6}$$

is equal to 2. By definition of S and T: $b^* s = c^* t = 1$. According to (5):

$$e_l = \frac{1}{c_l^* X \Lambda_1^{-1} V^*(b_l \otimes d_l)} = \frac{1}{b_l^* Y \Lambda_2^{-1} W^*(c_l \otimes d_l)}. \tag{7}$$

The latter occurs only if (6) is equal to 2.

Thus we have proved that one set of b_l, c_l, d_l, and e_l can be computed independently from other triads. ■

To compute one triad we should compute SVD several times (about $n_1 n_2 n_3 r$ arithmetical operations), and find all local minima of the problem (6). To solve each particular minimum of (6) we can use either Newton's method or the method of sequential minimization. It is easy to prove that it converges at least linearly for the local minima. However, computing a full optimization problem for (6) is really difficult, but there is a simple remedy. If we compute at least one local minimum, then we find a corresponding triad. Let us subtract it from the initial data, and repeat the decomposition. If this problem after the `Tucker` algorithm has $r \times r \times n$ size, then the residual after `Tucker` algorithm will be $(r-1) \times (r-1) \times n$ size, and still contain two nonsingular factors. Then this approach terminates within r iterations.

Every new restart with residual data needs additional overheads of about $r^3(n_3+2)$ and each solution of (6) takes about cr^3 arithmetical operations, where c is the total number of iteration for the solution of (6). We can keep it in mind trying to find several independent triads by computation several local minima at the same time. When it is really impossible and takes a lot of additional work we can make a restart with residual matrix. Then the total estimation of arithmetical operations for our algorithm is bounded by $r^4(n_3+2+c)$ and $r^3(n_3+2+cr)$ arithmetical operations. This value is bigger than the cost of one iteration of `PARAFAC` algorithm (r^4 arithmetical operations), but if we consider the poor convergence and thousands of iterations then the total arithmetical complexity of our approach looks much more attractive.
All previous theorems can be generalized for N-way case ($N > 3$) but we did not present it because we do not know any application for it.
This approach minimize modified (1) problem. If B and C are square and nonsingular we can write the following minimization problem:

$$\min_{B,C,E,\hat{D}_1,\ldots,\hat{D}_{n_3}} \sum_{k=1}^{n_3} ||A_k C^{-*} - B\hat{D}_k||_F^2 + ||A_k B^{-*} - C\hat{D}_k||_F^2.$$

Since, this problem is not original (1) problem, we found out that our algorithm is really resistant to the different kind of noise from the numerical experiments.

3 Numerical Experiments

Now we demonstrate the efficiency of our new approach. We developed a software package containing `Tucker` and `PARAFAC`, and our new algorithm which we call Parallel Decomposition `PARDEC`. In addition we developed a software to provide initial data A_k. It is based on the normalized Gaussian-like functions with different variations of position in 3-D, different width distributed between $5-10\%$ from the size of each dimension, different scaling factor — amplitude distributed in $[0.5, 1]$ and threshold of orthogonality of initial function. All mentioned distributions are normal.
We make 3 series of data. The first series of data contains a $60 \times 60 \times 60$ array with 25 components. We variate the noise from 0% till 40%. The second series of data has the same size, number of components, and noise, but positions of peaks was randomly generated. The third series of data has bigger size — $150 \times 150 \times 150$ with 100 components distributed randomly in the domain with noise form 0% till 40%.
We reduce the size of all problems by `Tucker` algorithm to the cube with the edge equal to the number of components. It takes several seconds for

Table. The dependence of computational time, residual of approximation and the number of correctly found triads (N_t) on the amount of noise in the simulated data for the `PARAFAC` and `PARDEC` algorithms.

$60 \times 60 \times 60$ **problem**

Noise (%)	PARAFAC			PARDEC		
	Time (sec)	N_t	Residual	Time (sec)	N_t	Residual
0	210.8*	3	0.245	2.2	0	10^{-10}
5	210.8*	3	0.255	2.4	0	0.039
10	210.8*	3	0.255	2.5	0	0.081
15	254.7	0	0.128	2.5	0	0.128
20	78.2	0	0.180	2.7	0	0.180
25	137.4	0	0.237	2.7	0	0.237
30	161.3	0	0.299	2.9	0	0.299
35	8.9	0	0.366	4.6	1	0.367
40	5.4	0	0.439	3.5	0	0.439

$150 \times 150 \times 150$ **problem**

Noise (%)	PARAFAC			PARDEC		
	Time (sec)	N_t	Residual	Time (sec)	N_t	Residual
0	18385*	17	0.096	1357	0	10^{-10}
5	18423*	16	0.137	1935	0	0.041
10	18431*	13	0.170	2116	0	0.086
15	18437*	13	0.228	2353	2	0.143
20	18441*	11	0.281	2485	3	0.201
25	18444*	12	0.332	2471	2	0.261
30	18446*	10	0.367	2590	4	0.312
35	18448*	13	0.423	2644	5	0.380
40	18449*	15	0.459	2804	7	0.427

$*$ — computations were stopped after 10000 iterations.

the first and second series of data and several minutes for the third series, then we treat this reduced data by standard `PARAFAC` algorithm and our new `PARDEC` algorithm. We measure the computation time for an AMD Athlon 500 computer and the number of converged components. We consider that the component has converged if it is similar to the original triad in l_2 norm at least in 90% in each dimension and in the amplitude.

The first series of data is too easy to solve — both algorithms solve it within $1.5 - 2.5$ seconds and all components have converged for both algorithms. The results for the second and the third series of data are presented in the Table.

The computation time for the PARAFAC algorithm for the third series of data are constant for any amount of noise because the computations are stopped after 10000 iterations.
We tested some data of intermediate sizes and the results show the following properties: *if the data contain components which are close to orthogonal, then both algorithms solve it fast enough. If the noise grows, sometimes* PARAFAC *produces slightly better results than* PARDEC*, but works too slowly and not so stable. If the number of components grows, then* PARDEC *becomes faster and more stable.*

References

[1] Harshman, R.: Foundations of PARAFAC procedure: Models and conditions for an "exploratory" multi-mode analysis. UCLA Working Papers in Phonetics, 16, 1–84 (1970).

[2] Carroll, J. D., Chang, J.: Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decompositions. Psychometrika 35, 283–319 (1970).

[3] Ligny, C. L., Spanjer, N., Houwelingen, J. C., Weesie H.M.: Three-mode factor analysis of data on retention in normal-phase high performance liquid chromatography. J. Chromatogr. 301, 311–324 (1984).

[4] Harshman, R., Ladefoged, P., Goldstein, L.: Factor analysis of tongue shapes. J. Acoust. Soc. Am. 62, 693 1977.

[5] Kruskal, J. B.: Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. Lin. Alg. Appl. 18, 95–138 (1977).

[6] Orekhov, V., Ibraghimov, I., Billeter, M: MUNIN: A new Approach to Multi-dimensional NMR Spectra Interpretation, J. Biomol. NMR 20, 49–60, (2001).

[7] Harshman, R., Lundy, M.: The PARAFAC model for three-way factor analysis and Multidimensional scaling. Research Methods for Multimode Data Analysis. New York: Praeger 122–215, 1984.

[8] Henrion, R.: On global, local and stationary solutions in three-way data analysis, J. Chemom. 14(3), 261–274, (2000).

[9] Tucker, L. R.: Some mathematical notes on three-mode factor analysis, Psychometrika 31, 279–311, (1966).

[10] Kronenberg P. M., Leeuw, J.: Principal component analysis of three-mode data by means of alternating least squares algorithms, Psychometrika 45, 69–97, (1980).

[11] Bro, R.: PARAFAC. Tutorial and applications, Chemom. Int. Lab. Sys. 38, 149–171, (1997).

[12] Leurgans S. E., Ross, R. T., Abel, R. B.: A decomposition for three-way arrays, SIAM J. Matrix Anal. Appl. 14(4), 1064–1083, (1993).