Chapter 7

Multilinear Algebra for Analyzing Data with Multiple Linkages

Daniel M. Dunlavy[*], *Tamara G. Kolda*[†], *and W. Philip Kegelmeyer*[†]

Abstract

Tensors are a useful tool for representing multi-link graphs, and tensor decompositions facilitate a type of link analysis that incorporates all link types simultaneously. An adjacency tensor is formed by stacking the adjacency matrix for each link type to form a three-way array. The CANDECOMP/PARAFAC (CP) tensor decomposition provides information about adjacency tensors of multi-link graphs analogous to that produced for adjacency matrices of single-link graphs using the singular value decomposition (SVD). The CP tensor decomposition generates feature vectors that incorporate all linkages simultaneously for each node in a multi-link graph. Feature vectors can be used to analyze bibliometric data in a variety of ways, for example, to analyze five years of publication data from journals published by the Society for Industrial and Applied Mathematics (SIAM). Experiments presented include analyzing a body of work, distinguishing between papers written by different authors with the same name, and predicting the journal in which a paper is published.

[*]Computer Science and Informatics Department, Sandia National Laboratories, Albuquerque, NM 87185–1318 (dmdunla@sandia.gov).

[†]Informatics and Decision Sciences Department, Sandia National Laboratories, Livermore, CA 94551-9159 (tgkolda@sandia.gov, wpk@sandia.gov).

Note: First appeared as Sandia National Laboratories Technical Report SAND2006-2079, Albuquerque, NM and Livermore, CA, April 2006.

7.1 Introduction

Multi-link graphs, i.e., graphs with multiple link types, are challenging to analyze, yet such data are ubiquitous. For example, Adamic and Adar [Adamic & Adar 2005] analyzed a social network where nodes are connected by organizational structure, i.e., each employee is connected to his or her boss, and also by direct email communication. Social networks clearly have many types of links—familial, communication (phone, email, etc.), organizational, geographical, etc.

Our overarching goals are to analyze data with multiple link types and to derive feature vectors for each individual node (or data object). As a motivating example, we use journal publication data—specifically considering several of the many ways that two papers may be linked. The analysis is applied to five years of journal publication data from eleven journals and a set of conference proceedings published by the Society for Industrial and Applied Mathematics (SIAM). The nodes represent published papers. Explicit, directed links exist whenever one paper cites another. Undirected similarity links are derived based on title, abstract, keyword, and authorship. Historically, bibliometric researchers have focused solely on citation analysis or text analysis, but not both simultaneously. Though this work focuses on the analysis of publication data, the techniques are applicable to a wide range of tasks, such as higher order web link graph analysis [Kolda & Bader 2006, Kolda et al. 2005].

Link analysis typically focuses on a single link type. For example, both PageRank [Brin & Page 1998] and HITS [Kleinberg 1999] consider the structure of the web and decompose the adjacency matrix of a graph representing the hyperlink structure. Instead of decomposing an adjacency matrix that represents a single matrix, our approach is to decompose an adjacency *tensor* that represents multiple link types.

A tensor is a multidimensional, or N-way, array. For multiple linkages, a three-way array can be used, where each two-dimensional *frontal slice* represents the adjacency matrix for a single link type. If there are N nodes and K link types, then the data can be represented as a three-way tensor of size $N \times N \times K$ where the (i, j, k) entry is nonzero if node i is connected to node j by link type k. In the example of Adamic and Adar [Adamic & Adar 2005] discussed above, there are two links types: organization connections versus email communication connections. For bibliometric data, the five different link types mentioned above correspond to (frontal) slices in the tensor; see Figure 7.1.

The CANDECOMP/PARAFAC (CP) tensor decomposition (see, for instance, [Carroll & Chang 1970, Harshman 1970]) is a higher order analog of the matrix singular value decomposition (SVD). The CP decomposition applied to the adjacency tensor of a multi-link graph leads to the following types of analysis.

- The CP decomposition reveals "communities" within the data and how they are connected. For example, a particular factor may be connected primarily by title similarity while another may depend mostly on citations.
- The CP decomposition also generates feature vectors for the nodes in the graph, which can be compared directly to get a similarity score that combines the multiple linkage types.

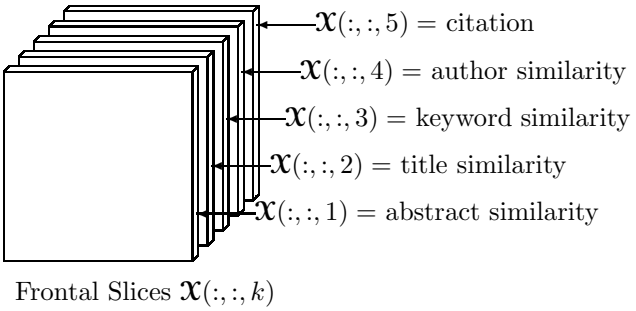


Figure 7.1. Tensor slices.
Slices of a third-order tensor representing a multi-link graph.

• The average of a set of feature vectors represents a *body of work*, e.g., by a given author, and can be used to find the most similar papers in the larger collection.

• The feature vectors can be used for disambiguation. In this case, the feature vectors associated with the body of work for two or more authors indicate whether they are the same authors or not. For example, is H. SIMON the same as H. S. SIMON?

• By inputting the feature vectors to a supervised learning method (decision trees and ensembles), the publication journal for each paper can be predicted.

This chapter is organized as follows. A description of the CP tensor decomposition and how to compute it is provided in Section 7.2. We discuss the properties of the data and how they are represented as a sparse tensor in Section 7.3. Numerical results are provided in Section 7.4. Related work is discussed in Section 7.5. Conclusions and ideas for future work are discussed in Section 7.6.

7.2 Tensors and the CANDECOMP/PARAFAC decomposition

This section provides a brief introduction to tensors and the CP tensor decomposition. For a survey of tensors and their decompositions, see [Kolda & Bader 2009].

7.2.1 Notation

Scalars are denoted by lowercase letters, e.g., c. Vectors are denoted by boldface lowercase letters, e.g., $\mathbf{v}$. The ith entry of $\mathbf{v}$ is denoted by $\mathbf{v}(i)$. Matrices are denoted by boldface capital letters, e.g., $\mathbf{A}$. The jth column of $\mathbf{A}$ is denoted by $\mathbf{A}(:,j)$ and element (i,j) by $\mathbf{A}(i,j)$. Tensors (i.e., N-way arrays) are denoted by boldface Euler script letters, e.g., $\mathcal{X}$. Element (i,j,k) of a third-order tensor $\mathcal{X}$ is denoted by $\mathcal{X}(i,j,k)$. The kth frontal slice of a three-way tensor is denoted by $\mathcal{X}(:,:,k)$; see Figure 7.1.

7.2.2 Vector and matrix preliminaries

The symbol $\otimes$ denotes the *Kronecker product of vectors*; for example

$$\mathbf{x} = \mathbf{a} \otimes \mathbf{b} \quad \Leftrightarrow \quad \mathbf{x}(\ell) = \mathbf{a}(i)\mathbf{b}(j) \\ \text{where } \ell = j + (i-1)(J) \text{ for all } 1 \le i \le I,\ 1 \le j \le J$$

This is a special case of the Kronecker product of matrices.

The symbol $.*$ denotes the *Hadamard matrix product*. This is the element-wise product of two matrices of the same size.

The symbol $\odot$ denotes the *Khatri–Rao product* (or column wise Kronecker product) of two matrices [Smilde et al. 2004]. For example, let $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$. Then

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{A}(:,1) \otimes \mathbf{B}(:,1) & \mathbf{A}(:,2) \otimes \mathbf{B}(:,2) & \cdots & \mathbf{A}(:,K) \otimes \mathbf{B}(:,K) \end{bmatrix}$$

is a matrix of size $(IJ) \times K$.

7.2.3 Tensor preliminaries

The norm of a tensor is given by the square root of the sum of the squares of all its elements; i.e., for a tensor $\mathcal{X}$ of size $I \times J \times K$

$$\| \mathcal{X} \|^2 \equiv \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \mathcal{X}(i,j,k)^2$$

This is the higher order analog of the Frobenius matrix norm.

The symbol $\circ$ denotes the *outer product of vectors*. For example, let $\mathbf{a} \in \mathbb{R}^I$, $\mathbf{b} \in \mathbb{R}^J$, $\mathbf{c} \in \mathbb{R}^K$. Then

$$\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \quad \Leftrightarrow \quad \mathcal{X}(i,j,k) = \mathbf{a}(i)\mathbf{b}(j)\mathbf{c}(k) \\ \text{for all} \quad 1 \le i \le I,\ 1 \le j \le J,\ 1 \le k \le K$$

A *rank-one tensor* is a tensor that can be written as the outer product of vectors. For $\boldsymbol{\lambda} \in \mathbb{R}^R$, $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$, the *Kruskal operator* [Kolda 2006] denotes a sum of rank-one tensors

$$[\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \equiv \sum_{r=1}^{R} \boldsymbol{\lambda}(r)\ \mathbf{A}(:,r) \circ \mathbf{B}(:,r) \circ \mathbf{C}(:,r) \in \mathbb{R}^{I \times J \times K}$$

If $\boldsymbol{\lambda}$ is a vector of ones, then $[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is used as shorthand.

Matricization, also known as *unfolding* or *flattening*, is the process of reordering the elements of an N-way array into a matrix; in particular, the mode-n matricization of a tensor $\mathcal{X}$ is denoted by $\mathbf{X}_{(n)}$; see, e.g., [Kolda 2006]. For a three-way tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, the mode-$n$ unfoldings are defined as follows

$$\mathbf{X}_{(1)}(i,p) = \mathcal{X}(i,j,k) \qquad \text{where } p = j + (k-1)(J) \tag{7.1}$$

$$\mathbf{X}_{(2)}(j,p) = \mathcal{X}(i,j,k) \qquad \text{where } p = i + (k-1)(I) \tag{7.2}$$

$$\mathbf{X}_{(3)}(k,p) = \mathcal{X}(i,j,k) \qquad \text{where } p = i + (j-1)(I) \tag{7.3}$$

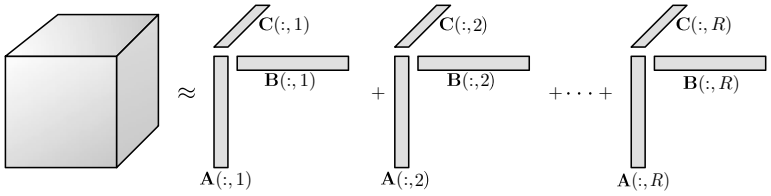


Figure 7.2. CP decomposition.
Approximates a tensor by a sum of rank-one factors.

7.2.4 The CP tensor decomposition

The CP decomposition, first proposed by Hitchcock [Hitchcock 1927] and later rediscovered simultaneously by Carroll and Chang [Carroll and Chang 1970] and Harshman [Harshman 1970], is a higher order analog of the matrix SVD. It should not be confused with the Tucker decomposition [Tucker 1966], a different higher order analog of the SVD.

CP decomposes a tensor into a sum of rank-one tensors. Let $\mathcal{X}$ be a tensor of size $I \times J \times K$. A CP decomposition with R factors approximates the tensor $\mathcal{X}$ as

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{A}(:,r) \circ \mathbf{B}(:,r) \circ \mathbf{C}(:,r) \equiv [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

where $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$. The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are called the *component matrices*. Figure 7.2 illustrates the decomposition.

It is useful to normalize the columns of the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ to length one and rewrite the CP decomposition as

$$\mathcal{X} \approx \sum_{r=1}^{R} \boldsymbol{\lambda}(r)\ \mathbf{A}(:,r) \circ \mathbf{B}(:,r) \circ \mathbf{C}(:,r) \equiv [\![\boldsymbol{\lambda}\ ; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{R}$. In contrast to the solution provided by the SVD, the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ do not have orthonormal columns [Kolda 2001, Kolda & Bader 2009].

Each rank-one factor, $\boldsymbol{\lambda}(r)\ \mathbf{A}(:,r) \circ \mathbf{B}(:,r) \circ \mathbf{C}(:,r)$, represents a "community" within the data; see Section 7.4.1. The number of factors in the approximation, R, should loosely reflect the number of *communities* in the data. Often some experimentation is required to determine the most useful value of R.

7.2.5 CP-ALS algorithm

A common approach to fitting a CP decomposition is the ALS (alternating least squares) algorithm [Carroll & Chang 1970, Harshman 1970]; see also, [Tomasi 2006, Faber et al. 2003, Tomasi & Bro 2006]. At each inner iteration, the CP-ALS

algorithm solves for one-component matrix while holding the others fixed. For example, it solves for the matrix $\mathbf{C}$ when $\mathbf{A}$ and $\mathbf{B}$ are fixed, i.e.,

$$\min_{\mathbf{C}} \| \mathcal{X} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \| \tag{7.4}$$

In this case, $\boldsymbol{\lambda}$ is omitted because it will just be absorbed into the lengths of the columns of $\mathbf{C}$ when the computation is complete. Equation (7.4) can be rewritten as a matrix problem (see, e.g., [Smilde et al. 2004])

$$\min_{\mathbf{C}} \left\| \mathbf{X}_{(3)} - \mathbf{C} \left(\mathbf{B} \odot \mathbf{A} \right)^{\mathsf{T}} \right\| \tag{7.5}$$

Here $\mathbf{X}_{(3)}$ is the mode-3 matricization or unfolding from equation 7.3.

Solving this problem makes use of the pseudoinverse of a Khatri–Rao product, given by

$$(\mathbf{B} \odot \mathbf{A})^{\dagger} = \left((\mathbf{B}\mathbf{B}) \,.*\, \left(\mathbf{A}^{\mathsf{T}}\mathbf{A} \right) \right)^{\dagger} (\mathbf{B} \odot \mathbf{A})^{\mathsf{T}}$$

Note that only the pseudoinverse of an $R \times R$ matrix needs to be calculated rather than that of an $IJ \times R$ matrix [Smilde et al. 2004].

The optimal $\mathbf{C}$ is the least squares solution to equation (7.5)

$$\mathbf{C} = \mathbf{X}_{(3)} \left[(\mathbf{B} \odot \mathbf{A})^{\mathsf{T}} \right]^{\dagger} = \mathbf{X}_{(3)} (\mathbf{B} \odot \mathbf{A}) \left(\left(\mathbf{B}^{\mathsf{T}}\mathbf{B} \right) .* \left(\mathbf{A}^{\mathsf{T}}\mathbf{A} \right) \right)^{\dagger}$$

which can be computed efficiently thanks to the properties of the Khatri–Rao product. The other component matrices can be computed in an analogous fashion using mode-1 and mode-2 matricizations of $\mathcal{X}$ in solving for $\mathbf{A}$ and $\mathbf{B}$, respectively.

It is generally efficient to initialize the ALS algorithm with the R leading eigenvectors of $\mathbf{X}_{(n)}\mathbf{X}_{(n)}^{\mathsf{T}}$ for the nth component matrix as long as the nth dimension of $\mathcal{X}$ is at least as big as R; see, e.g., [Kolda & Bader 2009]. Otherwise, random initialization can be used. Only two of the three initial matrices need to be computed since the other is solved for in the first step. The CP-ALS algorithm is presented in Algorithm 7.1.

Algorithm 7.1. CP-ALS.
CP decomposition via an alternating least squares. $\mathcal{X}$ is a tensor of size $I \times J \times K$, $R > 0$ is the desired number of factors in the decomposition, $M > 0$ is the maximum number of iterations to perform, and $\epsilon > 0$ is the stopping tolerance.

CP-ALS $(\mathcal{X}, R, M, \epsilon)$

$m = 0$
$\mathbf{A} = R$ principal eigenvectors of $\mathbf{X}_{(1)}\mathbf{X}_{(1)}^{\mathsf{T}}$
$\mathbf{B} = R$ principal eigenvectors of $\mathbf{X}_{(2)}\mathbf{X}_{(2)}^{\mathsf{T}}$
repeat
  $m = m + 1$
  $\mathbf{C} = \mathbf{X}_{(3)}(\mathbf{B} \odot \mathbf{A}) \left((\mathbf{B}^{\mathsf{T}}\mathbf{B}) .* (\mathbf{A}^{\mathsf{T}}\mathbf{A})\right)^{\dagger}$
  Normalize columns of $\mathbf{C}$ to length 1
  $\mathbf{B} = \mathbf{X}_{(2)}(\mathbf{C} \odot \mathbf{A}) \left((\mathbf{C}^{\mathsf{T}}\mathbf{C}) .* (\mathbf{A}^{\mathsf{T}}\mathbf{A})\right)^{\dagger}$
  Normalize columns of $\mathbf{B}$ to length 1
  $\mathbf{A} = \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B}) \left((\mathbf{C}^{\mathsf{T}}\mathbf{C}) .* (\mathbf{B}^{\mathsf{T}}\mathbf{B})\right)^{\dagger}$
  Store column norms of $\mathbf{A}$ in $\boldsymbol{\lambda}$ and
  normalize columns of $\mathbf{A}$ to length 1
until $m > M$ or $\| \mathcal{X} - [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \| < \epsilon$
return $\boldsymbol{\lambda} \in \mathbb{R}^{R}$; $\mathbf{A} \in \mathbb{R}^{I \times R}$; $\mathbf{B} \in \mathbb{R}^{J \times R}$; $\mathbf{C} \in \mathbb{R}^{K \times R}$
  such that $\mathcal{X} \approx [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$

In the discussion that follows, $\boldsymbol{\Lambda}$ denotes the $R \times R$ diagonal matrix whose diagonal is $\boldsymbol{\lambda}$.

All computations were performed using the Tensor Toolbox for MATLAB (see [Bader & Kolda 2006, Bader & Kolda 2007]), which was appropriate because of its ability to handle large-scale, sparse tensors.

7.3 Data

The data consist of publication metadata from eleven SIAM journals as well as SIAM proceedings for the period 1999–2004. There are 5022 articles; the number of articles per publication is shown in Table 7.1. The names of the journals used throughout this paper are their ISI abbreviations* and "SIAM PROC" is used to indicate the proceedings.

7.3.1 Data as a tensor

The data are represented as an $N \times N \times K$ tensor where $N = 5022$ is the number of documents and $K = 5$ is the number of link types. The five link types are described below; see also Figure 7.1.

(1) The first slice $(\mathcal{X}(:, :, 1))$ represents abstract similarity; i.e., $\mathcal{X}(i, j, 1)$ is the cosine similarity of the abstracts for documents i and j. The Text to Matrix Generator (TMG) v2.0 [Zeimpekis & Gallopoulos 2006] was used to generate a term-document matrix, $\mathbf{T}$. All words appearing on the default TMG stopword list as well as words starting with a number were removed. The matrix was weighted using term frequency and inverse document frequency local and global weightings

*http://www.isiknowledge.com/.

Table 7.1. SIAM publications.
Names of the SIAM publications along with the number of articles of each used as data for the experiments.

Journal Name	Articles
SIAM J APPL DYN SYST	32
SIAM J APPL MATH	548
SIAM J COMPUT	540
SIAM J CONTROL OPTIM	577
SIAM J DISCRETE MATH	260
SIAM J MATH ANAL	420
SIAM J MATRIX ANAL APPL	423
SIAM J NUMER ANAL	611
SIAM J OPTIM	344
SIAM J SCI COMPUT	656
SIAM PROC	469
SIAM REV	142

(tf.idf); this means that

$$\mathbf{T}(i,j) = f_{ij} \log_2(N/N_i)$$

where f_{ij} is the frequency of term i in document j and N_i is the number of documents that term i appears in. Each column of $\mathbf{T}$ is normalized to length one (for cosine scores). Finally

$$\mathcal{X}(:,:,1) = \mathbf{T}^{\mathsf{T}}\mathbf{T}$$

Because they are cosine scores, all are in the range $[0,1]$. In order to sparsify the slice, only scores greater than 0.2 (chosen heuristically to reduce the total number of nonzeros in all three text similarity slices to approximately 250,000) are retained.

(2) The second slice ($\mathcal{X}(:,:,2)$) represents title similarity; i.e., $\mathcal{X}(i,j,2)$ is the cosine similarity of the titles for documents i and j. It is computed in the same manner as the abstract similarity slice.

(3) The third slice ($\mathcal{X}(:,:,3)$) represents author-supplied keyword similarity; i.e., $\mathcal{X}(i,j,3)$ is the cosine similarity of the keywords for documents i and j. It is computed in the same manner as the abstract similarity slice.

(4) The fourth slice ($\mathcal{X}(:,:,4)$) represents author similarity; i.e., $\mathcal{X}(i,j,4)$ is the similarity of the authors for documents i and j. It is computed as follows. Let $\mathbf{W}$ be the author-document matrix such that

$$\mathbf{W}(i,j) = \begin{cases} 1/\sqrt{M_j} & \text{if author } i \text{ wrote document } j, \\ 0 & \text{otherwise} \end{cases}$$

where M_j is the number of authors for document j. Then

$$\mathcal{X}(:,:,4) = \mathbf{W}^{\mathsf{T}}\mathbf{W}$$

(5) The fifth slice ($\mathcal{X}(:,:,5)$) represents citation information; i.e.,

$$\mathcal{X}(i,j,5) = \begin{cases} 2 & \text{if document } i \text{ cites document } j, \\ 0 & \text{otherwise} \end{cases}$$

For this document collection, a weight of 2 was chosen heuristically so that the overall *slice weight* (i.e., the sum of all the entries in $\mathcal{X}(:,:,k)$, see Table 7.3) would not be too small relative to the other slices. The interpretation is that there are relatively few connections in this slice, but each citation connection indicates a strong connection. In future work, we would like to consider less ad hoc ways of determining the value for citation links.

Each slice is an adjacency matrix of a particular graph. The first four slices are symmetric and correspond to undirected graphs; the fifth slice is asymmetric and corresponds to a directed graph. These graphs can be combined into a multi-link graph and a corresponding tensor representation since they are all on the same set of nodes.

These choices for link types are examples of what can be done—many other choices are possible. For instance, asymmetric similarity weights are an option; e.g., if document i is a subset of document j, the measure might say that document i is very similar to document j, but document j is not so similar to document i. Other symmetric measures include co-citation or co-publication in the same journal.

7.3.2 Quantitative measurements on the data

Table 7.2 shows overall statistics on the data set. Note that some of the documents in this data set have empty titles, abstracts, or keywords; the averages shown in the table are not adjusted for the lack of data for those documents. Recall that Table 7.1 shows the number of articles per journal. In Table 7.2, the citations are counted only when both articles are in the data set and reflect the number of citations *from* each article. The maximum citations *to* a single article is 15.

Table 7.3 shows the number of nonzero entries and the sums of the entries for each slice. The text similarity slices ($k = 1, 2, 3$) have large numbers of nonzeros but low average values, the author similarity slice has few nonzeros but a higher average value, and the citation slice has the fewest nonzeros but all values are equal to 2.

7.4 Numerical results

The results use a CP decomposition of the data tensor $\mathcal{X} \in \mathbb{R}^{N \times N \times K}$

$$\mathcal{X} \approx [\![\boldsymbol{\lambda} \, ; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

where $\boldsymbol{\lambda} \in \mathbb{R}^R$, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$. Using $R = 30$ factors worked well for the experiments and is the default value unless otherwise noted.

Table 7.2. SIAM journal characteristics.
Characteristics of the SIAM journal and proceedings data (5022 documents in total).

	Total in Collection	Per Document	
		Average	Maximum
Unique terms	16617	148.32	831
abstracts	15752	128.06	802
titles	5164	10.16	33
keywords	5248	10.10	40
Authors	6891	2.19	13
Citations (within collection)	2659	0.53	12

Table 7.3. SIAM journal tensors.
Characteristics of the tensor representation of the SIAM journal and proceedings data.

Slice (k)	Description	Nonzeros	$\sum_i \sum_j \mathcal{X}(i,j,k)$
1	Abstract Similarity	28476	7695.28
2	Title Similarity	120236	33285.79
3	Keyword Similarity	115412	16201.85
4	Author Similarity	16460	8027.46
5	Citation	2659	5318.00

7.4.1 Community identification

The rank-one CP factors (see Figure 7.2) reveal communities within the data. The largest entries for the vectors in each factor

$$(\mathbf{A}(:,r), \mathbf{B}(:,r), \mathbf{C}(:,r))$$

correspond to interlinked entries in the data. For the rth factor, high-scoring nodes in $\mathbf{A}(:,r)$ are connected to high-scoring nodes in $\mathbf{B}(:,r)$ with the high-scoring link types in $\mathbf{C}(:,r)$. Recall that the fifth link type, representing citations, is asymmetric; when that link type scores high in $\mathbf{C}(:,r)$, then the highest-scoring nodes in $\mathbf{A}(:,r)$ can be thought of as papers that cite the highest-scoring nodes in $\mathbf{B}(:,r)$.

For example, consider the first factor ($r = 1$). The link scores from $\mathbf{C}(:,1)$ are shown in Table 7.4. Title and keyword similarities are strongest. In fact, the top three link types are based on text similarity and so are symmetric. Therefore, it is no surprise that the highest-scoring nodes in $\mathbf{A}(:,1)$ and $\mathbf{B}(:,1)$, also shown in Table 7.4, are nearly identical. This community is related primarily by text similarity and is about the topic "conservation laws."

On the other hand, the tenth factor ($r = 10$) has citation as the dominant link type; see Table 7.5. Citation links are asymmetric, so the highest-scoring nodes in $\mathbf{A}(:,10)$ and $\mathbf{B}(:,10)$ are not the same. This is a community that is linked primarily

Table 7.4. First community in CP decomposition.
Community corresponding to the first factor ($r = 1$) of the CP tensor decomposition with $R = 30$ factors.

Link scores in C(:, 1)	
Score	**Link Type**
0.95	Title Similarity
0.28	Keyword Similarity
0.07	Abstract Similarity
0.06	Citation
0.06	Author Similarity

Paper node scores in A(:, 1) (top 10)

Score	Title
0.18	On the boundary control of systems of conservation laws
0.17	On stability of conservation laws
0.16	Two a posteriori error estimates for 1D scalar conservation laws
0.16	A free boundary problem for scalar conservation laws
0.15	Convergence of SPH method for scalar nonlinear conservation laws
0.15	Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic ...
0.15	High-order central schemes for hyperbolic systems of conservation laws
0.15	Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws

Paper node scores in B(:, 1) (top 10)

Score	Title
0.18	On the boundary control of systems of conservation laws
0.18	On stability of conservation laws
0.16	Two a posteriori error estimates for one-dimensional scalar conservation laws
0.16	A free boundary problem for scalar conservation laws
0.16	Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic ...
0.16	Convergence of SPH method for scalar nonlinear conservation laws
0.15	Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws
0.14	High-order central schemes for hyperbolic systems of conservation laws

because the high-scoring papers in $\mathbf{A}(:, 10)$ cite the high-scoring papers in $\mathbf{B}(:, 10)$. The topic of this community is "preconditioning," though the third paper in $\mathbf{B}(:, 10)$ is not about preconditioning directly but rather a graph technique that can be used by preconditioners—that is why it is on the "cited" side.

The choice to have symmetric or asymmetric connections affects the interpretation of the CP model. In this case, the tensor has four symmetric slices and one asymmetric slice. If all of the slices were symmetric, then this would be a special case of the CP decomposition called the INDSCAL decomposition [Carroll & Chang 1970] where $\mathbf{A} = \mathbf{B}$. In related work, Selee et al. [Selee et al. 2007] have investigated this situation.

7.4.2 Latent document similarity

The CP component matrices $\mathbf{A}$ and $\mathbf{B}$ provide latent representations (i.e., feature vectors) for each document node. These feature vectors can, in turn, be used to

Table 7.5. Tenth community in CP decomposition.
Community corresponding to the tenth factor ($r = 10$) of the CP tensor decomposition with $R = 30$ factors.

Link scores in C(:, 10)	
Score	**Link Type**
0.96	Citation
0.19	AuthorSim
0.16	TitleSim
0.10	KeywordSim
0.06	AbstractSim

Paper node scores in A(:, 10) (top 10)

Score	**Title**
0.36	Multiresolution approximate inverse preconditioners
0.20	Preconditioning highly indefinite and nonsymmetric matrices
0.16	A factored approximate inverse preconditioner with pivoting
0.16	On two variants of an algebraic wavelet preconditioner
0.14	A robust and efficient ILU that incorporates the growth of the inverse triangular factors
0.11	An algebraic multilevel multigraph algorithm
0.11	On algorithms for permuting large entries to the diagonal of a sparse matrix
0.11	Preconditioning sparse nonsymmetric linear systems with the Sherman–Morrison formula

Paper node scores in B(:, 10) (top 10)

Score	**Title**
0.27	Ordering anisotropy and factored sparse approximate inverses
0.25	Robust approximate inverse preconditioning for the conjugate gradient method
0.23	A fast and high-quality multilevel scheme for partitioning irregular graphs
0.20	Orderings for factorized sparse approximate inverse preconditioners
0.19	The design and use of algorithms for permuting large entries to the diagonal of ...
0.17	BILUM: Block versions of multielimination and multilevel ILU preconditioner ...
0.16	Orderings for incomplete factorization preconditioning of nonsymmetric problems
0.15	Preconditioning highly indefinite and nonsymmetric matrices

compute document similarity scores inclusive of text, authorship, and citations. Since there are two applicable component matrices, $\mathbf{A}$, $\mathbf{B}$, or some combination can be used. For example

$$\mathbf{S} = \frac{1}{2}\mathbf{A}\mathbf{A}^\mathsf{T} + \frac{1}{2}\mathbf{B}\mathbf{B}^\mathsf{T} \tag{7.6}$$

Here $\mathbf{S}$ is an $N \times N$ similarity matrix where the similarity for documents i and j is given by $\mathbf{S}(i, j)$.

It may also be desirable to incorporate $\mathbf{\Lambda}$, e.g.,

$$\mathbf{S} = \frac{1}{2}\mathbf{A}\mathbf{\Lambda}\mathbf{A}^\mathsf{T} + \frac{1}{2}\mathbf{B}\mathbf{\Lambda}\mathbf{B}^\mathsf{T}$$

This issue is reminiscent of the choice facing users of latent semantic indexing (LSI) [Dumais et al. 1988] which uses the SVD of a term-document matrix, producing term and document matrices. In LSI, there is a choice of how to use the diagonal scaling for the queries and comparisons [Berry et al. 1995].

Table 7.6. Articles similar to *Link Analysis ....*
Comparison of most similar articles to *Link Analysis: Hubs and Authorities on the World Wide Web* using different numbers of factors in the CP decomposition.

R = 10

Score	Title
0.000079	Ordering anisotropy and factored sparse approximate inverses
0.000079	Robust approximate inverse preconditioning for the conjugate gradient method
0.000077	An interior point algorithm for large-scale nonlinear programming
0.000073	Primal-dual interior-point methods for semidefinite programming in finite precision
0.000068	Some new search directions for primal-dual interior point methods in semidefinite ...
0.000068	A fast and high-quality multilevel scheme for partitioning irregular graphs
0.000067	Reoptimization with the primal-dual interior point method
0.000065	Superlinear convergence of primal-dual interior point algorithms for nonlinear ...
0.000064	A robust primal-dual interior-point algorithm for nonlinear programs
0.000063	Orderings for factorized sparse approximate inverse preconditioners

R = 30

Score	Title
0.000563	Skip graphs
0.000356	Random lifts of graphs
0.000354	A fast and high-quality multilevel scheme for partitioning irregular graphs
0.000322	The minimum all-ones problem for trees
0.000306	Rankings of directed graphs
0.000295	Squarish k-d trees
0.000284	Finding the k-shortest paths
0.000276	On floor-plan of plane graphs
0.000275	1-Hyperbolic graphs
0.000269	Median graphs and triangle-free graphs

As an example of how these similarity measures can be used, consider the paper *Link analysis: Hubs and authorities on the World Wide Web* by Ding et al., which presents an analysis of an algorithm for web graph link analysis. Table 7.6 shows the most similar articles to this paper based on equation (7.6) for *two different* CP decompositions with $R = 10$ and $R = 30$ factors. In the $R = 10$ case, the results are not very good because the "most similar" papers include several papers on interior point methods that are not related. The results for $R = 30$ are all focused on graphs and are therefore related. Observe that there is also a big difference in the magnitude of the similarity scores in the two different cases. This example illustrates that, just as with LSI, choosing the number of factors of the approximation (R) is heuristic and affects the similarity scores.

In the next section, feature vectors from the CP factors are combined to represent a body of work.

7.4.3 Analyzing a body of work via centroids

Finding documents similar to a body of work may be useful in a literature search or in finding other authors working in a given area. This subsection and the next discuss two sets of experiments using centroids, corresponding to a term or an author, respectively, to analyze a body of work.

Consider finding collections of articles containing a particular term (or phrase). All articles containing the term in either the title, abstract, or keywords are identified and then the centroids $\mathbf{g}_A$ and $\mathbf{g}_B$ are computed using the columns of the matrices $\mathbf{A}$ and $\mathbf{B}$, respectively, for the identified articles. The similarity scores for all documents to the body of work are then computed as

$$\mathbf{s} = \frac{1}{2}\mathbf{A}\mathbf{g}_A + \frac{1}{2}\mathbf{B}\mathbf{g}_B \tag{7.7}$$

Consequently, $\mathbf{s}(i)$ is the similarity of the ith document to the centroid.

Table 7.7 shows the results of a search on the term "GMRES," which is an iterative method for solving linear systems. The table lists the top-scoring documents using a combination of matrices $\mathbf{A}$ and $\mathbf{B}$. In order not to overemphasize the papers that cite many of the papers about GMRES (i.e., using only the components from $\mathbf{A}$) or those which are most cited (i.e., using only the components from $\mathbf{B}$), combining the two sets of scores takes into account the content of the papers (i.e., abstracts, titles, and keywords) as an average of these two extremes. Thus, the average scores result in a more balanced look at papers about GMRES.

Similarly, centroids were used to analyze a body of work associated with a particular author. All of the articles written by an author were used to generate a centroid and similarity score vector as above. Table 7.8 shows the most similar papers to the articles written by V. Kumar, a researcher who focuses on several research areas, including graph analysis. In these ten articles in the table, only three papers (including the two authored by V. Kumar) are explicitly linked to V. Kumar by coauthorship or citations. Furthermore, several papers that are closely related to those written by V. Kumar focused on graph analysis, while some are not so obviously linked. Table 7.8 lists the authors as well to illustrate that such results could be used as a starting point for finding authors related to V. Kumar that are not necessarily linked by coauthorship or citation. In this case, the author W. P. Tang appears to be linked to V. Kumar.

Analysis of centroids derived from tensor decompositions can be useful in understanding small collections of documents. For example, such analysis could be useful for matching referees to papers. In this case, program committee chairs could create a centroid for each member on a program committee, and work assignments could be expedited by automatically matching articles to the appropriate experts.

As a segue to the next section, note that finding a set of documents associated with a particular author is not always straightforward. In fact, in the example above, there is also an author named V. S. A. Kumar, and it is not clear from article titles alone that this author is not the same one as V. Kumar. The next section discusses the use of the feature vectors produced by tensor decompositions for solving this problem of author disambiguation.

7.4.4 Author disambiguation

A challenging problem in working with publication data is determining whether two authors are in fact a single author using multiple aliases. Such problems are often caused by incomplete or incorrect data or varying naming conventions for authors

Table 7.7. Articles similar to *GMRES*.
Articles similar to the centroid of articles containing the term *GMRES* using the component matrices of a CP tensor decomposition to compute similarity scores.

Highest-scoring nodes using $\frac{1}{2}\mathbf{Ag}_A + \frac{1}{2}\mathbf{Bg}_B$

Score	Title
0.0134	FQMR: A flexible quasi-minimal residual method with inexact ...
0.0130	Flexible inner-outer Krylov subspace methods
0.0114	Adaptively preconditioned GMRES algorithms
0.0112	Truncation strategies for optimal Krylov subspace methods
0.0093	Theory of inexact Krylov subspace methods and applications to ...
0.0086	Inexact preconditioned conjugate gradient method with inner-outer iteration
0.0085	Flexible conjugate gradients
0.0078	GMRES with deflated restarting
0.0065	A case for a biorthogonal Jacobi–Davidson method: Restarting and ...
0.0062	On the convergence of restarted Krylov subspace methods

Highest-scoring nodes using $\mathbf{Ag}_A$

Score		
$\mathbf{Ag}_A$	$\mathbf{Bg}_B$	Title
0.0240	0.0019	Flexible inner-outer Krylov subspace methods
0.0185	0.0082	FQMR: A flexible quasi-minimal residual method with inexact ...
0.0169	0.0017	Theory of inexact Krylov subspace methods and applications to ...
0.0132	0.0024	GMRES with deflated restarting
0.0127	0.0003	A case for a biorthogonal Jacobi–Davidson method: Restarting and ...
0.0107	0.0010	A class of spectral two-level preconditioners
0.0076	0.0011	An augmented conjugate gradient method for solving consecutive ...

Highest-scoring nodes using $\mathbf{Bg}_B$

Score		
$\mathbf{Bg}_B$	$\mathbf{Ag}_A$	Title
0.0217	0.0011	Adaptively preconditioned GMRES algorithms
0.0158	0.0014	Inexact preconditioned conjugate gradient method with inner-outer iteration
0.0149	0.0074	Truncation strategies for optimal Krylov subspace methods
0.0113	0.0056	Flexible conjugate gradients
0.0082	0.0185	FQMR: A flexible quasi-minimal residual method with inexact ...
0.0080	0.0007	Linear algebra methods in a mixed approximation of magnetostatic problems
0.0063	0.0060	On the convergence of restarted Krylov subspace methods

used by different publications (e.g., J. R. Smith versus J. Smith). In the SIAM articles, there are many instances where two or more authors share the same last name and at least the same first initial, e.g., V. Torczon and V. J. Torczon. In these cases, the goal is to determine which names refer to the same person.

The procedure for solving this author disambiguation problem works as follows. For each author name of interest, we extract all the columns from the matrix $\mathbf{B}$ corresponding to the articles written by that author name. Recall that the matrix $\mathbf{B}$ comes from the $R = 30$ CP decomposition. Because of the directional citation links in $\mathcal{X}(:,:,5)$, using the matrix $\mathbf{B}$ slightly favors author names that are co-cited (i.e., their papers are cited together in papers), whereas using $\mathbf{A}$ would have slightly

Table 7.8. Similarity to V. Kumar.
Papers similar to those by V. KUMAR using a rank $R = 30$ CP tensor decomposition.

Score	Authors	Title
0.0645	Karypis G, Kumar V	A fast and high-quality multilevel scheme for partitioning ...
0.0192	Bank RE, Smith RK	The incomplete factorization multigraph algorithm
0.0149	Tang WP, Wan WL	Sparse approximate inverse smoother for multigrid
0.0115	Chan TF, Smith B, Wan WL	An energy-minimizing interpolation for robust methods ...
0.0114	Henson VE, Vassilevski PS	Element-free AMGe: General algorithms
0.0108	Hendrickson B, Rothberg E	Improving the run-time and quality of nested dissection ...
0.0092	Karypis G, Kumar V	Parallel multilevel k-way partitioning scheme for irregular ...
0.0091	Tang WP	Toward an effective sparse approximate inverse preconditioner
0.0085	Saad Y, Zhang J	BILUM: Block versions of multielimination and multilevel ...
0.0080	Bridson B, Tang WP	A structural diagnosis of some IC orderings

favored author names that co-cite (i.e., their papers cite the same papers). The centroid of those columns from $\mathbf{B}$ is used to represent the author name. Two author names are compared by computing the cosine similarity of their two centroids, resulting in a value between -1 (least similar) and 1 (most similar). In the example above, the similarity score of the centroids for V. TORCZON and V. J. TORCZON is 0.98, and thus there is a high confidence that these names both refer to the same person (verified by manual inspection of the articles).

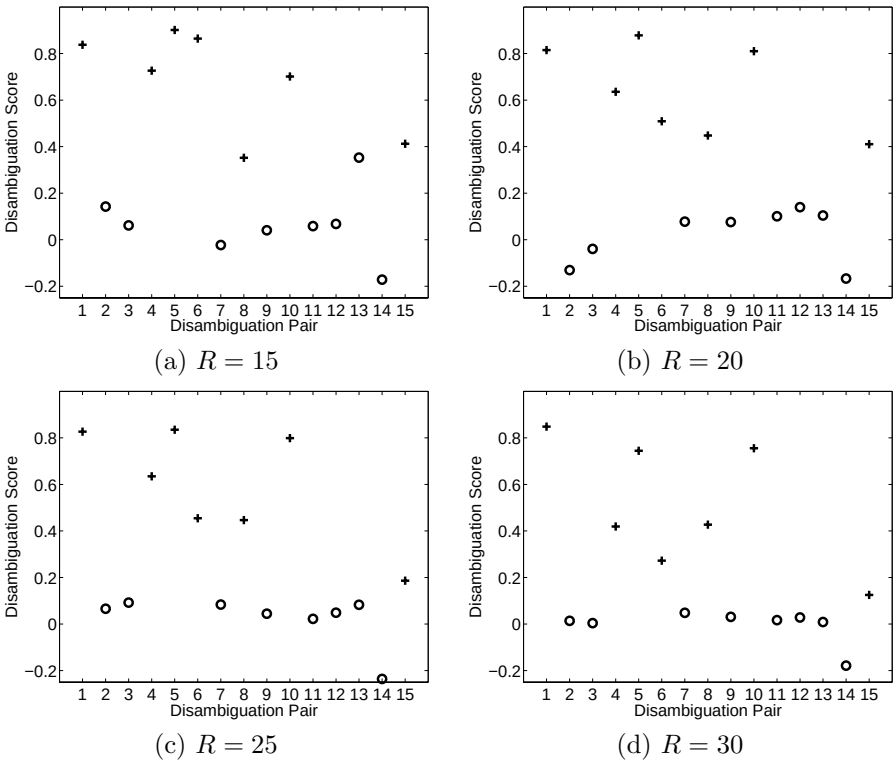
As an example use of author disambiguation, the following experiment was performed. (i) The top 40 author names of papers in the data set were selected, i.e., those with the most papers. (ii) For each author name in the top 40, all papers in the full document collection with any name sharing the same first initial and last name were retrieved. (iii) Next the centroids for each author name as in Section 7.4.3 were computed. (iv) The combined similarity scores using the formula in (7.7) were calculated for all papers of author names sharing the same first initial and last name. (v) Finally, the resulting scores were compared to manually performed checks to see which matches are correct.

According to the above criteria, there are a total of 15 pairs of names to disambiguate. Table 7.9 shows all the pairs and whether or not each is a correct match, which was determined manually.

Figure 7.3 presents plots of the similarity scores for these 15 pairs of author names using CP decompositions with $R = 15, 20, 25, 30$. The scores denoted by $+$ in the figure are those name pairs that refer to the same person, whereas those pairs denoted by $\circ$ refer to different people. Ideally, there will be a distinct cutoff between correct and incorrect matches. The figure shows that, in general, most correct matches have higher scores than the incorrect ones. However, there are several instances where there is not a clear separation between pairs in the two

Table 7.9. Author disambiguation.
Author name pairs to be disambiguated.

Pair	Name 1	Name 2	Same Person?
1	T. Chan	T. F. Chan	Yes
2	T. Chan	T. M. Chan	No
3	T. F. Chan	T. M. Chan	No
4	T. Manteuffel	T. A. Manteuffel	Yes
5	S. McCormick	S. F. McCormick	Yes
6	G. Golub	G. H. Golub	Yes
7	X. L. Zhou	X. Y. Zhou	No
8	R. Ewing	R. E. Ewing	Yes
9	S. Kim	S. C. Kim	No
10	S. Kim	S. D. Kim	Yes
11	S. Kim	S. J. Kim	No
12	S. C. Kim	S. D. Kim	No
13	S. C. Kim	S. J. Kim	No
14	S. D. Kim	S. J. Kim	No
15	J. Shen	J. H. Shen	Yes



(a) $R = 15$ (b) $R = 20$

(c) $R = 25$ (d) $R = 30$

Figure 7.3. Disambiguation scores.
Author disambiguation scores for various CP tensor decompositions (+ = correct; ∘ = incorrect).

Table 7.10. Disambiguation before and after.
Authors with most papers before and after disambiguation.

Before Disambuguation		After Disambuguation	
Papers	**Author**	**Papers**	**Author**
17	Q. Du	17	Q. Du
15	K. Kunisch	16	T. F. Chan
15	U. Zwick	16	T. A. Manteuffel
14	T. F. Chan	16	S. F. McCormick
13	A. Klar	15	K. Kunisch
13	T. A. Manteuffel	15	U. Zwick
13	S. F. McCormick	13	A. Klar
13	R. Motwani	13	R. Motwani
12	G. H. Golub	13	G. H. Golub
12	M. Y. Kao	12	M. Y. Kao
12	S. Muthukrishnan	12	S. Muthukrishnan
12	D. Peleg	12	D. Peleg
11	H. Ammari	12	S. D. Kim
11	N. J. Higham	11	H. Ammari
11	K. Ito	11	N. J. Higham
11	H. Kaplan	11	K. Ito
11	L. Q. Qi	11	H. Kaplan
11	A. Srinivasan	11	L. Q. Qi
11	X. Y. Zhou	11	A. Srinivasan
10	N. Alon	11	X. Y. Zhou

sets—e.g., pairs 8, 13, and 15 in Figure 7.3(a). The CP decomposition with $R = 20$ clearly separates the correct and incorrect matches. Future work in this area will focus on determining if there is an optimal value for R for the task of predicting cutoff values for separating correct and incorrect matches.

Table 7.10 shows how correctly disambiguating authors can make a difference in publication counts. The left column shows the top 20 authors before disambiguation, and the right column shows the result afterward. There are several author names—T. F. Chan, T. A. Manteuffel, S. F. McCormick, G. H. Golub, and S. D. Kim—that move up (some significantly) in the list when the ambiguous names are resolved correctly.

One complication that has not yet been addressed is that two different people may be associated with the same author name. This is particularly likely in the case that the name has only a single initial and a common last name. Consider the name Z. Wu—there are two papers in the collection with this author name and five others with author names with the same first initial and a different second initial. Table 7.11 lists the papers by these authors along with the full first name of the author, which was determined by manual inspection.

Two approaches for solving this name resolution problem are considered: treating Z. Wu as a single author and taking the centroid of the two papers and treating each paper as separate. In Table 7.12(a), Z. Wu, as the author of two papers, appears most similar to author 3. Separating the articles of Z. Wu and recomputing the scores provides much stronger evidence that authors 1b and 3 are the same author, and that author 1a is most likely not an alias for one of the other authors; see Table 7.12(b).

Table 7.11. Data used in disambiguating the author Z. Wu.

ID	Author	Title(s)
1a	Wu Z (Zhen)	Fully coupled forward-backward stochastic differential equations and ...
1b	Wu Z (Zili)	Sufficient conditions for error bounds
2	Wu ZJ (Zhijun)	A fast Newton algorithm for entropy maximization in phase determination
3	Wu ZL (Zili)	First order and second order conditions for error bounds
3	Wu ZL (Zili)	Weak sharp solutions of variational inequalities in Hilbert spaces
4	Wu ZN (Zi-Niu)	Steady and unsteady shock waves on overlapping grids
4	Wu ZN (Zi-Niu)	Efficient parallel algorithms for parabolic problems

Table 7.12. Disambiguation of author Z. Wu.

(a) Combination of all ambiguous authors.

	1	2	3	4
1	1.00	0.18	0.79	0.03
2		1.00	0.06	0.06
3			1.00	0.01
4				1.00

(b) Separation of all ambiguous authors.

	1a	1b	2	3	4
1a	1.00	0.01	0.21	0.03	0.07
1b		1.00	0.09	**0.90**	0.00
2			1.00	0.06	0.06
3				1.00	0.01
4					1.00

Manual inspection of all the articles by this group of authors indicates that authors 1b and 3 are in fact the same person, Zili Wu, and that author 1a is not an alias of any other author in this group. The verified full name of each author is listed in parentheses in Table 7.11.

The experiments and results presented in this section suggest several ways that tensor decompositions can be used for resolving ambiguity in author names. In particular, the use of centroids for characterizing a body of work associated with an author shows promise for solving this problem. In the next set of experiments, though, it can be observed that the utility of centroids may be limited to small, cohesive collections, as they fail to produce useful results for the problem of predicting which journal an article may appear in.

7.4.5 Journal prediction via ensembles of tree classifiers

Another analysis approach, supervised machine learning with the feature vectors obtained in Section 7.4.2, may be used to predict the journal that a given paper is published in.

Table 7.13. Summary journal prediction results.

ID	Journal Name	Size	Correct	Mislabeled as	
1	SIAM J APPL DYN SYST	1%	0%	2	(44%)
2	SIAM J APPL MATH	11%	58%	6	(10%)
3	SIAM J COMPUT	11%	56%	11	(20%)
4	SIAM J CONTROL OPTIM	11%	60%	2	(10%)
5	SIAM J DISCRETE MATH	5%	15%	3	(47%)
6	SIAM J MATH ANAL	8%	26%	2	(29%)
7	SIAM J MATRIX ANAL APPL	8%	56%	10	(19%)
8	SIAM J NUMER ANAL	12%	50%	10	(16%)
9	SIAM J OPTIM	7%	66%	4	(16%)
10	SIAM J SCI COMPUT	13%	36%	8	(21%)
11	SIAM PROC	9%	32%	3	(38%)
12	SIAM REV	3%	5%	2	(34%)

The approach from Section 7.4.3 of considering the centroid of a body of work does not yield useful results in the case of journals because the centroids are not sufficiently distinct. Therefore, classifiers trained on subsets of the data are used to predict the journals in which the articles not included in those training sets are published. The feature vectors were based on the matrix $\mathbf{A}$ from a CP decomposition with $R = 30$ components. Thus, each document is represented by a length-30 feature vector, and the journal in which it is published is used as the label value, i.e., the classification. The 5022 labeled feature vectors were split into ten disjoint partitions, stratified so that the relative proportion of each journal's papers remained constant across the partitions. Ten-fold cross validation was used, meaning that each one of the ten partitions (10% of the data) was used once as testing data and the remaining nine partitions (90% of the data) were used to train the classifier. This computation was done using OpenDT [Banfield et al. 2004] to create bagged ensembles [Dietterich 2000] of C4.5 decision trees. The ensemble size was 100; larger ensembles did not improve performance.

Table 7.13 provides an overview of the results giving, for each journal, its identification number, its size relative to the entire collection, the percentage of its articles that were correctly classified, and the journal that it was most often mislabeled as and how often that occurred. For instance, articles in journal 2, make up 11% of the total collection, are correctly identified 58% of the time, and are confused with journal 6 most often (10% of the time). The overall "confusion matrix" is given in Table 7.14; this matrix is obtained by combining the confusion matrices generated for each of the ten folds.

Figure 7.4 shows a graphical representation of the confusion matrix. Each journal is represented as a node, and the size of the node corresponds to the percentage of its articles that were correctly labeled (0–66%). There is a directed edge from journal i to journal j if journal i's articles were mislabeled as article j. A Barnes–Hut forced directed method (using the weighted edges) was used to determine the positions of the nodes [Beyer 2007]. Only those edges corresponding to mislabeling percentages of 5% or higher are actually shown in the image (though all were used for the layout); the thicker the edge, the greater the proportion of mislabeled articles.

Table 7.14. Predictions of publication.
Confusion matrix of predictions of publication of articles in the different SIAM publications. A classifier based on bagging and using decision trees as weak learners was used in this experiment. The bold entries are correct predictions.

	Predicted Journal											
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
1	**0**	14	4	1	1	4	0	3	1	1	2	1
2	1	**318**	19	46	3	54	13	31	7	41	12	3
3	0	29	**303**	24	29	5	15	8	7	10	109	1
4	0	57	21	**346**	2	34	20	12	51	22	11	1
5	0	12	122	9	**40**	4	15	2	1	2	53	0
6	0	120	19	56	1	**108**	15	58	3	34	5	1
7	0	23	11	22	5	8	**235**	18	18	81	2	0
8	0	56	13	47	0	37	37	**304**	13	98	5	1
9	0	10	19	55	1	4	10	5	**228**	1	10	1
10	0	77	7	32	0	36	98	135	23	**237**	7	4
11	0	37	176	21	34	12	9	8	7	13	**149**	3
12	1	48	13	12	2	13	16	6	6	10	8	**7**

Figure 7.4. Journals linked by mislabeling.

The automatic layout generated by the Barnes–Hut algorithm visually yields four clusters, and the nodes in Figure 7.4 are color-coded according to their cluster labels. These journals along with their descriptions are presented in Table 7.15, and they are clearly clustered by overlap in topics. Observe that, for example,

Table 7.15. Journal clusters.
Journals grouped by how they are generally confused, with descriptions.

ID	Topic
	Red-Colored Nodes: Dynamical Systems
2	SIAM J APPL MATH: scientific problems using methods that are of mathematical interest such as asymptotic methods, bifurcation theory, dynamical systems theory, and probabilistic and statistical methods
6	SIAM J MATH ANAL: partial differential equations, the calculus of variations, functional analysis, approximation theory, harmonic or wavelet analysis, or dynamical systems; applications to natural phenomena
1	SIAM J APPL DYN SYST: mathematical analysis and modeling of dynamical systems and its application to the physical, engineering, life, and social sciences
12	SIAM REV: articles of broad interest
	Green-Colored Nodes: Optimization
4	SIAM J CONTROL OPTIM: mathematics and applications of control theory and on those parts of optimization theory concerned with the dynamical systems
9	SIAM J OPTIM: theory and practice of optimization
	Purple-Colored Nodes: Discrete Math and Computer Science
3	SIAM J COMPUT: mathematical and formal aspects of computer science and nonnumerical computing
5	SIAM J DISCRETE MATH: combinatorics and graph theory, discrete optimization and operations research, theoretical computer science, and coding and communication theory
11	SIAM PROC: Conference proceedings including SIAM Data Mining, ACM-SIAM Symposium on Discrete Algorithms, Conference on Numerical Aspects of Wave Propagation, etc.
	Cyan-Colored Nodes: Numerical Analysis
7	SIAM J MATRIX ANAL APPL: matrix analysis and its applications
8	SIAM J NUMER ANAL: development and analysis of numerical methods including convergence of algorithms, their accuracy, their stability, and their computational complexity
10	SIAM J SCI COMPUT: numerical methods and techniques for scientific computation

the scope of SIAM J COMPUT (3) includes everything in the scope of SIAM J DISCRETE MATH (5), so it is not surprising that many of the latter's articles are misidentified as the former. In cases where there is little overlap in the stated scope, there seems to be less confusion. For instance, articles from the SIAM J OPTIM (9) are correctly labeled 66% of the time and the only other journal it is confused with more than 5% of the time is the other optimization journal represented in the collection: SIAM J CONTROL OPTIM (4). Note that the SIAM J CONTROL OPTIM (4) does include dynamical systems in its description and is, in fact, linked to the "dynamical systems" cluster.

7.5 Related work

7.5.1 Analysis of publication data

Researchers look at publication data to understand the impact of individual authors and who is collaborating with whom, to understand the type of information being

published and by which venues, and to extract "hot topics" and understand trends [Boyack 2004].

As an example of the interest in this problem, the 2003 KDD Cup challenge brought together 57 research teams from around the world to focus on the analysis of publication data for citation prediction (i.e., implicit link detection in a citation graph), citation graph creation, and usage estimation (downloads from a server of preprint articles) [Gehrke et al. 2003]. The data were from the high-energy physics community (a portion of the arXiv preprint server collection*). For this challenge, McGovern et al. looked at a number of questions related to the analysis of publication data [McGovern et al. 2003]. Of particular relevance to this paper, they found that clustering papers based only on text similarity did not yield useful clusters. Instead, they applied spectral-based clustering to a citation graph where the edges were weighted by the cosine similarity of the paper abstracts—combining citation and text information into one graph. Additionally, for predicting in which journal an article will be published, they used relational probability trees (see Section 7.5.3).

In other work, Barábasi et al. [Barábasi et al. 2002] considered the social network of scientific collaborations based on publication data, particularly the properties of the entire network and its evolution over time. In their case, the data were from publications in mathematics and neuroscience. The nodes correspond to authors and the links to coauthorship.

Hill and Provost [Hill & Provost 2003] used only citation information to predict authorship with an accuracy of 45%. They created a profile on each author based on his/her citation history (weighting older citations less). This profile can then be used to predict the authorship of a paper where only the citation information is known but not the authors. They did not use any text-based matching but observe that using such methods may improve accuracy.

Jo, Lagoze, and Giles [Jo et al. 2007] used citation graphs to determine topics in a large-scale document collection. For each term, the documents (nodes in the citation graph) were down-selected to those containing a particular term. The interconnectivity of those nodes within the "term" subgraph was used to determine whether or not it comprises a topic. The intuition of their approach was that, if a term represents a topic, the documents containing that term will be highly interconnected; otherwise, the links should be random. They applied their method to citation data from the arXiv (papers in physics) and Citeseer† papers in computer science) preprint server collections.

7.5.2 Higher order analysis in data mining

Tensor decompositions such as CP [Carroll & Chang 1970, Harshman 1970] and Tucker [Tucker 1966] (including HO-SVD [De Lathauwer et al. 2000] as a special case) have been in use for several decades in psychometrics and chemometrics and have recently become popular in signal processing, numerical analysis, neuroscience,

*http://www.arXiv.org/.
†http://citeseer.ist.psu.edu/.

computer vision, and data mining. See [Kolda & Bader 2009] for a comprehensive survey.

Recently, tensor decompositions have been applied to data-centric problems including analysis of click-through data [Sun et al. 2005] and chatroom analysis [Acar et al. 2005, Acar et al. 2006]. Liu et al. [Liu et al. 2005] presented a tensor space model which outperforms the classical vector space model for the problem of classification of Internet newsgroups. In the area of web hyperlink analysis, the CP decomposition has been used to extend the well-known HITS method to incorporate anchor text information [Kolda et al. 2005, Kolda & Bader 2006]. Bader et al. [Bader et al. 2007a, Bader et al. 2007b] used tensors to analyze the communications in the Enron e-mail data set. Sun et al. [Sun et al. 2006a, Sun et al. 2006b] dynamically updated Tucker models for detecting anomalies in network data. Tensors have also been used for multiway clustering, a method for clustering entities of different types based on both entity attributes as well as the connections between the different types of entities [Banerjee et al. 2007].

7.5.3 Other related work

Cohn and Hofmann [Cohn & Hofmann 2001] developed a joint probability model that combines text and links, with an application to categorizing web pages. Relational probability trees (RPTs) [Getoor et al. 2003, Getoor & Diehl 2005] offer a technique for analyzing graphs with different link and node types, with the goal of predicting node or link attributes.

For the problem of author disambiguation, addressed in this paper, Bekkerman and McCallum [Bekkerman & McCallum 2005] have developed an approach called multiway distributional clustering (MDC) that clusters data of several types (e.g., documents, words, and authors) based on interactions between the types. They used an instance of this method for disambiguation of individuals appearing in pages on the web.

7.6 Conclusions and future work

Multiple similarities between documents in a collection are represented as a three-way tensor ($N \times N \times K$), the tensor is decomposed using the CP-ALS algorithm, and relationships between the documents are analyzed using the CP component matrices. How to best choose the weights of the entries of the tensor is an open topic of research—the ones used here were chosen heuristically.

Different factors from the CP decomposition are shown to emphasize different link types; see Section 7.4.1. Moreover, the highest-scoring components in each factor denote an interrelated community. The component matrices ($\mathbf{A}$ and $\mathbf{B}$) of the CP decomposition can be used to derive feature vectors for latent similarity scores. However, the number of components (R) of the CP decomposition can strongly influence the quality of the matches; see Section 7.4.2. The choice of the number of components (R) and exactly how to use the component matrices are open questions, including how to combine these matrices, how to weight or normalize the features, and whether or not to incorporate the factor weightings, i.e., $\boldsymbol{\lambda}$.

This brings us to two disadvantages of the CP model. First, the factor matrices are not orthogonal, in contrast to the matrix SVD. A possible remedy for this is to instead consider the Tucker decomposition [Tucker 1966], which produces orthogonal component matrices and, moreover, can have a different number of columns for each component matrix; unfortunately, the Tucker decomposition is not unique and does not produce rank-one components like CP. Second, the best decomposition with R components is not the same as the first R factors of the optimal decomposition with $S > R$ components, again in contrast to the SVD [Kolda 2001]. This means that we cannot determine the optimal R by trial-and-error without great expense.

The centroids of feature vectors from the component matrices of the CP decomposition can be used to represent a small body of work (e.g., all the papers with the phrase "GMRES") in order to find related works. As expected, the feature vectors from the different component matrices produce noticeably different answers, either one of which may be more or less useful in different contexts; see Section 7.4.3. Combining these scores can be used to provide a ranked list of relevant work, taking into account the most relevant items from each of the component matrices.

A promising application of the similarity analysis is author disambiguation, where centroids are compared to predict which authors with similar names are actually the same. The technique is applied to the subset of authors with the most papers authored in the entire data set and affects the counts for the most published authors; see Section 7.4.4. In future work, we will consider the appropriate choice of the number of components (R) for disambiguation, identify how to choose the disambiguation similarity threshold, and perform a comparison to other approaches.

Using the feature vectors, it is possible to predict which journal each article was published in; see Section 7.4.5. Though the accuracy was relatively low, closer inspection of the data yielded clues as to why. For example, two of the publications were not focused publications. Overall, the results revealed similarities between the different journals. In future work, we will compare the results of using ensembles of decision trees to other learning methods (e.g., k-nearest neighbors, perceptrons, and random forests).

We also plan to revisit the representation of the data on two fronts. First, we wish to add authors as nodes. Hendrickson [Hendrickson 2007] observes that term-by-document matrices can be expanded to be (term *plus* document)-by-(term *plus* document) matrices so that term-term and document-document connections can be additionally encoded. Therefore, we intend to use a (document *plus* author) dimension so that we can explicitly capture connections between documents and authors as well as the implicit connections between authors, such as colleagues, conference co-organizers, etc. Second, in order to make predictions or analyze trends over time, we intend to incorporate temporal information using an additional dimension for time.

Though the CP decomposition has indications of the importance of each link in the communities it identifies (see Section 7.4.1), we do not exploit this information in reporting or computing similarities. As noted in [Ramakrishnan et al. 2005], understanding *how* two entities are related is an important issue and a topic for future work.

The reasons that the spectral properties of adjacency matrices aid in clustering are beginning to be better understood; see, e.g., [Brand & Huang 2003]. Similar analyses to explain the utility of the CP model for higher order data are needed.

7.7 Acknowledgments

Data used in this paper were extracted from the Science Citation Index Expanded, Thomson ISI, Philadelphia, PA, USA. We gratefully acknowledge Brett Bader for his work on the MATLAB tensor toolbox [Bader & Kolda 2006, Bader & Kolda 2007], which was used in our computations, and for providing the image used in Figure 7.2; the TMG Toolbox creators for providing a helpful tool for generating term-document matrices in MATLAB [Zeimpekis & Gallopoulos 2006]; Ann Yoshimura for TaMALE, which was used to create Figure 7.4; and Dirk Beyer for providing the code CCVisu* to generate the Barnes–Hut layouts.

References

[Acar et al. 2005] E. Acar, S.A. Çamtepe, M.S. Krishnamoorthy, and B. Yener. Modeling and multiway analysis of chatroom tensors. In *ISI 2005: Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, vol. 3495 of *Lecture Notes in Computer Science*, 256–268, New York: Springer, 2005.

[Acar et al. 2006] E. Acar, S.A. Çamtepe, and B. Yener. Collective sampling and analysis of high order tensors for chatroom communications. In *ISI 2006: Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, vol. 3975 of *Lecture Notes in Computer Science*, 213–224, New York: Springer, 2006.

[Adamic & Adar 2005] L. Adamic and E. Adar. How to search a social network. *Social Networks*, 27:187–203, 2005.

[Bader et al. 2007a] B.W. Bader, M.W. Berry, and M. Browne. Discussion tracking in Enron email using PARAFAC. In M.W. Berry and M. Castellanos, eds., *Survey of Text Mining: Clustering, Classification, and Retrieval, Second Edition*, 147–162, New York: Springer, 2007.

[Bader et al. 2007b] B.W. Bader, R.A. Harshman, and T.G. Kolda. Temporal analysis of semantic graphs using ASALSAN. In *ICDM 2007: Proceedings of the 7th IEEE International Conference on Data Mining*, 33–42, 2007.

[Bader & Kolda 2006] B.W. Bader and T.G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32:635–653, 2006.

*http://www.sosy-lab.org/~dbeyer.

[Bader & Kolda 2007] B.W. Bader and T.G. Kolda. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30:205–231, 2007.

[Banerjee et al. 2007] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *SDM07: Proceedings of the 2007 SIAM International Conference on Data Mining*, http://www.siam.org/proceedings/datamining/2007/dm07.php, 145–156, 2007.

[Banfield et al. 2004] R. Banfield et al. OpenDT Web Page. http://opendt.sourceforge.net/, 2004.

[Barábasi et al. 2002] A.L. Barábasi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A*, 311:590–614, 2002.

[Bekkerman & McCallum 2005] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *WWW 2005: Proceedings of the 14th International Conference on World Wide Web*, 463–470, ACM Press, 2005.

[Berry et al. 1995] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.

[Beyer 2007] D. Beyer. CCVisu: A tool for co-change visualization and general force-directed graph layout, version 1.0. http://www.sosy-lab.org/~dbeyer, 2007.

[Boyack 2004] K.W. Boyack. Mapping knowledge domains: Characterizing PNAS. *Proceedings of the National Academy of Sciences*, 101:5192–5199, 2004.

[Brand & Huang 2003] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[Brin & Page 1998] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *WWW7: Proceedings of the Seventh International World Wide Web Conference*, 107–117, Elsevier, 1998.

[Carroll & Chang 1970] J.D. Carroll and J.J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35:283–319, 1970.

[Cohn & Hofmann 2001] D. Cohn and T. Hofmann. The missing link—a probabilistic model of document content and hypertext connectivity. In *NIPS 2000: Advances in Neural Information Processing Systems*, 13:460–436, 2001.

[De Lathauwer et al. 2000] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21:1253–1278, 2000.

[Dietterich 2000] T.G. Dietterich. Ensemble methods in machine learning. In Josef Kittler and Fabio Roli, eds., *First International Workshop on Multiple Classifier Systems*, no. 1857 in *Lecture Notes in Computer Science*, 1–15. New York: Springer, 2000.

[Dumais et al. 1988] S.T. Dumais, G.W. Furnas, T.K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *CHI '88: Proceedings of the SIGCHI Conference on Fuman Factors in Computing Systems*, 281–285, ACM Press, 1988.

[Faber et al. 2003] N.M. Faber, R. Bro, and P.K. Hopke. Recent developments in CANDECOMP/PARAFAC algorithms: A critical review. *Chemometrics and Intelligent Laboratory Systems*, 65:119–137, 2003.

[Gehrke et al. 2003] J. Gehrke, P. Ginsparg, and J. Kleinberg. Overview of the 2003 KDD cup. *ACM SIGKDD Explorations Newsletter*, 5:149–151, 2003.

[Getoor et al. 2003] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3:679–707, 2003.

[Getoor & Diehl 2005] L. Getoor and C.P. Diehl. Link mining: A survey. *ACM SIGKDD Explorations Newsletter*, 7:3–12, 2005.

[Harshman 1970] R.A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970. Available at http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf.

[Hendrickson 2007] B. Hendrickson. Latent semantic analysis and Fiedler retrieval. *Linear Algebra and Its Applications*, 421:345–355, 2007.

[Hill & Provost 2003] S. Hill and F. Provost. The myth of the double-blind review?: Author identification using only citations. *ACM SIGKDD Explorations Newsletter*, 5:179–184, 2003.

[Hitchcock 1927] F.L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6:164–189, 1927.

[Jo et al. 2007] Y. Jo, C. Lagoze, and C.L. Giles. Detecting research topics via the correlation between graphs and texts. In *KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 370–379, ACM Press, 2007.

[Kleinberg 1999] J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999.

[Kolda 2001] T.G. Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23:243–255, 2001.

[Kolda 2006] T.G. Kolda. Multilinear operators for higher-order decompositions. Technical Report SAND2006-2081, Sandia National Laboratories, Albuquerque, New Mexico and Livermore, Calif., April 2006.

[Kolda & Bader 2006] T.G. Kolda and B.W. Bader. The TOPHITS model for higher-order web link analysis. In *Workshop on Link Analysis, Counterterrorism and Security*, 2006.

[Kolda & Bader 2009] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM Review*, 51:455–500, 2009.

[Kolda et al. 2005] T.G. Kolda, B.W. Bader, and J.P. Kenny. Higher-order web link analysis using multilinear algebra. In *ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining*, 242–249, IEEE Computer Society, 2005.

[Liu et al. 2005] N. Liu, B. Zhang, J. Yan, Z. Chen, W. Liu, F. Bai, and L. Chien. Text representation: From vector to tensor. In *ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining*, 725–728, IEEE Computer Society, 2005.

[McGovern et al. 2003] A. McGovern, L. Friedland, M. Hay, B. Gallagher, A. Fast, J. Neville, and D. Jensen. Exploiting relational structure to understand publication patterns in high-energy physics. *ACM SIGKDD Explorations Newsletter*, 5:165–172, 2003.

[Ramakrishnan et al. 2005] C. Ramakrishnan, W.H. Milnor, M. Perry, and A.P. Sheth. Discovering informative connection subgraphs in multi-relational graphs. *ACM SIGKDD Explorations Newsletter*, 7:56–63, 2005.

[Selee et al. 2007] T.M. Selee, T.G. Kolda, W.P. Kegelmeyer, and J.D. Griffin. Extracting clusters from large datasets with multiple similarity measures using IMSCAND. In *CSRI Summer Proceedings 2007*, M. L. Parks and S. S. Collis, eds., Technical Report SAND2007-7977, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 87–103, December 2007.

[Smilde et al. 2004] A. Smilde, R. Bro, and P. Geladi. *Multi-Way Analysis: Applications in the Chemical Sciences*. West Sussex, England: Wiley, 2004.

[Sun et al. 2005] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen. CubeSVD: A novel approach to personalized web search. In *WWW 2005: Proceedings of the 14th International Conference on World Wide Web*, 382–390, ACM Press, 2005.

[Sun et al. 2006a] J. Sun, S. Papadimitriou, and P.S. Yu. Window-based tensor analysis on high-dimensional and multi-aspect streams. In *ICDM 2006: Proceedings of the 6th IEEE Conference on Data Mining*, 1076–1080, IEEE Computer Society, 2006.

[Sun et al. 2006b] J. Sun, D.Tao, and C. Faloutsos. Beyond streams and graphs: Dynamic tensor analysis. In *KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 374–383, ACM Press, 2006.

[Tomasi 2006] G. Tomasi. *Practical and computational aspects in chemometric data analysis*. PhD thesis, Department of Food Science, The Royal Veterinary and Agricultural University, Frederiksberg, Denmark, 2006. Available at http://www.models.life.ku.dk/theses/.

[Tomasi & Bro 2006] G. Tomasi and R. Bro. A comparison of algorithms for fitting the PARAFAC model. *Computational Statistics & Data Analysis*, 50:1700–1734, 2006.

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

[Zeimpekis & Gallopoulos 2006] D. Zeimpekis and E. Gallopoulos. TMG: A MATLAB toolbox for generating term-document matrices from text collections. In Jacob Kogan, Charles Nicholas, and Marc Teboulle, eds., *Grouping Multidimensional Data: Recent Advances in Clustering*, 187–210, New York: Springer, 2006.