The TOPHITS Model for Higher-Order Web Link Analysis*

Tamara Kolda† Brett Bader‡

Abstract

As the size of the web increases, it becomes more and more important to analyze link structure while also considering context. Multilinear algebra provides a novel tool for incorporating anchor text and other information into the authority computation used by link analysis methods such as HITS. Our recently proposed TOPHITS method uses a higher-order analogue of the matrix singular value decomposition called the PARAFAC model to analyze a three-way representation of web data. We compute hubs and authorities together with the terms that are used in the anchor text of the links between them. Adding a third dimension to the data greatly extends the applicability of HITS because the TOPHITS analysis can be performed in advance and offline. Like HITS, the TOPHITS model reveals latent groupings of pages, but TOPHITS also includes latent term information. In this paper, we describe a faster mathematical algorithm for computing the TOPHITS model on sparse data, and Web data is used to compare HITS and TOPHITS. We also discuss how the TOPHITS model can be used in queries, such as computing context-sensitive authorities and hubs. We describe different query response methodologies and present experimental results.

Keywords

PARAFAC, multilinear algebra, link analysis, higher-order SVD

1 Introduction

1.1 Overview As the size of the web continues to grow, link analysis methods must continue to advance. Topical HITS (TOPHITS) [31] is a higher-order generalization of the well-known HITS model of Kleinberg [27]. TOPHITS adds a third dimension to form an adjacency tensor that incorporates anchor text information; see Figure 1. This additional information provides a way of incorporating context into the calculation of authorities and hubs, which is accomplished via a three-way Parallel Factors (PARAFAC) decomposition [7, 23], a higher-order analogue of the singular value decomposition (SVD) [21]. By including anchor text in a third dimension, this approach also has some connections to Latent Semantic Indexing (LSI) [17, 4, 16], which is a popular method in text retrieval that uses dimensionality reduction to improve search.

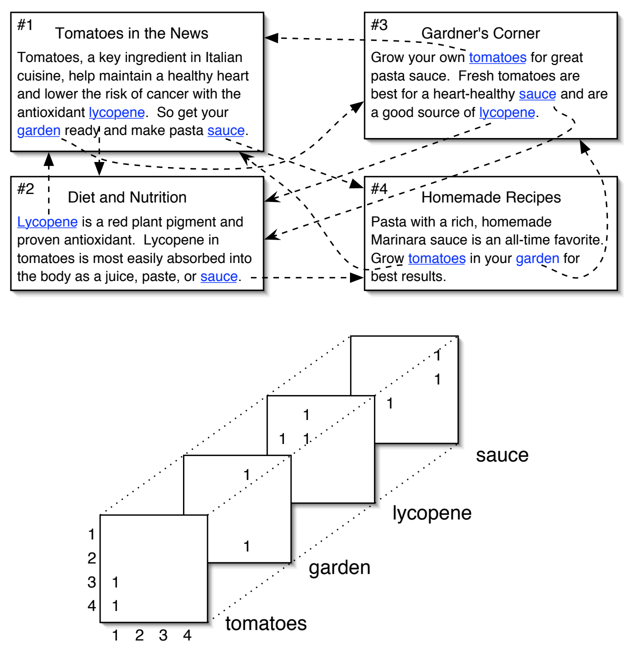

Figure 1: TOPHITS analyzes a three-way tensor representing a collection of web pages.

1.2 Notation Scalars are denoted by lowercase letters, e.g., a. Vectors are denoted by boldface lowercase letters, e.g., $\mathbf{a}$. The ith entry of $\mathbf{a}$ is denoted by a_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 a_{ij}. Tensors, i.e., multi-way arrays, are denoted by boldface Euler script letters, e.g., $\boldsymbol{\mathscr{X}}$. Element (i,j,k) of a 3rd-order tensor $\boldsymbol{\mathscr{X}}$ is denoted by x_{ijk}. The symbol $\circ$ denotes the outer product of vectors; for example, if $\mathbf{a} \in \mathbb{R}^I$, $\mathbf{b} \in \mathbb{R}^J$, $\mathbf{c} \in \mathbb{R}^K$, then $\boldsymbol{\mathscr{X}} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ if and

*This research was sponsored by the United States Department of Energy and by Sandia National Laboratory, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE–AC04–94AL85000.

†Sandia Natl. Labs, Livermore, CA, tgkolda@sandia.gov

‡Sandia Natl. Labs, Albuquerque, NM, bwbader@sandia.gov

only if $x_{ijk} = a_i b_j c_k$ for all $1 \le i \le I$, $1 \le j \le J$, $1 \le k \le K$. The symbol $\otimes$ denotes the Kronecker product of vectors; for example, $\mathbf{x} = \mathbf{a} \otimes \mathbf{b}$ means $x_\ell = a_i b_j$ with $\ell = j + (i-1)(J)$ for all $1 \le i \le I$, $1 \le j \le J$. The symbol $*$ denotes the Hadamard, i.e., elementwise, matrix product. 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 $\mathcal{Y}$ of size $I_1 \times I_2 \times \cdots \times I_N$,

$$\| \mathcal{Y} \|^2 \equiv \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} (y_{i_1 i_2 \cdots i_N})^2.$$

This is the higher-order analogue of the matrix Frobenius norm.

1.3 HITS and TOPHITS Many methods for analyzing the web, like PageRank [43] and HITS [27], are based on the adjacency matrix of a graph of a collection of web pages; see, e.g., Langville and Meyer [33, 34] for a general survey of these methods. PageRank scores are given by the entries of the principal eigenvector of a Markov matrix of page transition probabilities, i.e., a normalized version of the adjacency matrix plus a random-surfer component. HITS, on the other hand, computes both hub and authority scores for each node, and they correspond to the principal left and right singular vectors of the adjacency matrix (though it can also be modified to include a type of random-surfer component [15]). Other methods adhere to the same basic theme. For example, SALSA is a variant on HITS that uses a stochastic iteration matrix [36].

An interesting feature of HITS, which is not shared with PageRank, is that multiple pairs of singular vectors can be considered [27]. Consider a collection of I web pages. In HITS, the $I \times I$ adjacency matrix $\mathbf{X}$ is defined as

(1.1)
$$x_{ij} = \begin{cases} 1 & \text{if page } i \text{ points to page } j \\ 0 & \text{otherwise} \end{cases} \quad \text{for } 1 \le i, j \le I.$$

The HITS method can be thought of as follows. It uses the matrix SVD [21] to compute a rank-R approximation of $\mathbf{X}$:

$$\mathbf{X} \approx \mathbf{H} \mathbf{\Sigma} \mathbf{A}^{\mathsf{T}} \equiv \sum_{r=1}^{R} \sigma_r \, \mathbf{h}_r \circ \mathbf{a}_r. \tag{1.2}$$

Here $\mathbf{\Sigma} = \operatorname{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_R\}$ and we assume $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_R > 0$. The matrices $\mathbf{H}$ and $\mathbf{A}$ are each of size $I \times R$ and have orthonormal columns. We can view this as approximating the matrix $\mathbf{X}$ by the sum of R rank-1 outer products, as shown in Figure 2. The principal pair of singular vectors, $\mathbf{h}_1$ and $\mathbf{a}_1$, provide, respectively, hub and authority scores for the *dominant* topic in the web page collection. In other words, the pages that have the largest scores in $\mathbf{h}_1$ are the best hubs for the dominant topic; likewise, the pages that have the largest scores in $\mathbf{a}_1$ are the corresponding best authorities. Moreover, subsequent pairs of singular vectors reveal hubs and authorities for subtopics in the collection [27]. In fact, finding the appropriate pair of singular vectors for a given topic of interest is an open research question [13], and several groups of researchers have investigated how to incorporate content information into the HITS method [5, 10].

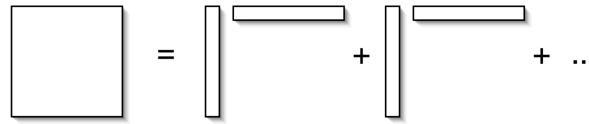

Figure 2: In HITS, the SVD provides a 2-way decomposition that yields hub and authority scores.

In previous work [31], we proposed the TOPHITS method, which is based on a three-way representation of the web where the third dimension encapsulates the anchor text. Let K be the number of terms used as anchor text. In TOPHITS, the $I \times I \times K$ adjacency tensor $\mathcal{X}$ is defined as

(1.3)
$$x_{ijk} = \begin{cases} 1 & \text{if page } i \text{ points to page } j \text{ using term } k \\ 0 & \text{otherwise.} \end{cases}$$
$$\text{for } 1 \le i, j \le I, \quad 1 \le k \le K.$$

Note that anchor text is useful for web search because it behaves as a consensus title [18]. The TOPHITS method uses the PARAFAC model [7, 23] (see §2.1) to generate a rank-R approximation of the form

$$\mathcal{X} \approx \boldsymbol{\lambda} \, [\![\mathbf{H}, \mathbf{A}, \mathbf{T}]\!] \equiv \sum_{r=1}^{R} \lambda_r \, \mathbf{h}_r \circ \mathbf{a}_r \circ \mathbf{t}_r. \tag{1.4}$$

Here we assume that $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_R$. The matrices $\mathbf{H}$, $\mathbf{A}$, $\mathbf{T}$ have columns of length one; but, in contrast to the solution provided by the SVD, these columns are not generally orthonormal [29]. The PARAFAC decomposition approximates the tensor $\mathcal{X}$ by the sum of R rank-1 outer products, as shown in Figure 3.

The principal triplet of PARAFAC vectors, $\mathbf{h}_1$, $\mathbf{a}_1$ and $\mathbf{t}_1$, provide, respectively, hub, authority, and term scores for the dominant topic (or grouping) in the web page collection. In other words, the pages that have the largest scores in $\mathbf{h}_1$ are the best hubs for the dominant grouping; likewise, the pages that have the largest scores

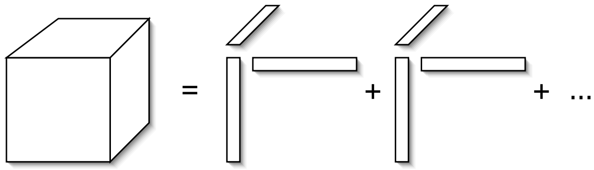

Figure 3: In TOPHITS, the PARAFAC decomposition provides a 3-way decomposition that yields hub, authority, and term scores.

in $\mathbf{a}_1$ are the corresponding best authorities and the terms that have the largest scores in $\mathbf{t}_1$ are the most descriptive terms.

1.4 Related work The problem of improving and extending web link analysis methods by incorporating anchor text or page content has received much attention in other work. For example, the problem of topic drift in HITS, which TOPHITS addresses via the third term dimension, has alternatively been solved by using a weighted adjacency matrix that increases the likelihood that the principal singular vectors relate to the query. The Clever system [8, 9] uses the content of the anchors and surrounding text to give more weight to those pages that are linked using terms in the search query, while Bharat and Henzinger [5] and Li et al. [37] incorporate weighting based on the content of the web pages. Henzinger et al. [26] recommend using text analysis of anchor text in conjunction with information obtained from the web graph for a better understanding of the nature of the links. Rafiei and Mendelzon [44] modify the page transition probabilities for PageRank based on whether or not a term appears in the page. Further, they derive a propagation model for HITS and adapt the same modification in that context. Richardson and Domingos [45] propose a general model that incorporates a term-based relevance function into PageRank. The relevance function can be defined in many ways, such as defining it to be 1 for any page that includes the term, and 0 otherwise. In an approach that is very similar in spirit to ours, though different in the mathematical implementation, Cohn and Hofmann [11] combine probabilistic LSI (PLSI) and probabilistic HITS (PHITS) so that terms and links rely on a common set of underlying factors.

The use of multidimensional models is relatively new in the context and web and data mining. Sun et al. [47] apply a 3-way Tucker decomposition [50] to the analysis of user × query-term × web-page data in order to personalize web search. In [1], various tensor decompositions of user × keyword × time data are used to separate different streams of conversation in chatroom data. Our contribution in [31] was the use of a "greedy" PARAFAC decomposition [23] on a web-page × web-page × anchor-text sparse, three-way tensor representing the web graph with anchor-text-labeled edges. To the best of our knowledge, this was the first use of PARAFAC for analyzing semantic graphs as well as the first instance of applying PARAFAC to sparse data. The history of tensor decompositions in general goes back forty years [50, 23, 7], and they have been used extensively in other domains ranging from chemometrics [46] to image analysis [51].

1.5 Our contribution Here we revisit the problem of how to compute the PARAFAC decomposition on large, sparse data in order to generate the TOPHITS model. In §2, we discuss two different methods for computing PARAFAC decompositions and in particular how those are applied to sparse data. To the best of our knowledge, we are the first to consider the problem of applying tensor decompositions to sparse, multidimensional data; therefore, the details of the implementation are relevant because they have not been presented before.

We also investigate ways in which the TOPHITS model can be used as the basis of a query system in §3. As has been observed many times, see, e.g., [27, 24], HITS is query-dependent. The TOPHITS method extends the applicability of HITS to any collection of web pages, not just a focused subgraph that is derived from a given query. In fact, the TOPHITS model can be computed offline and in advance, making it a viable tool for web analysis. Like PageRank [43], it is entirely query independent; however, its multiple sets of scores provide context sensitivity. Moreover, TOPHITS can be used for other types of queries as well, such as finding pages or terms that are most similar.

In §4, we present numerical results on sample data. We compare different PARAFAC algorithms for computing the TOPHITS model on our sample data and conclude that the ALS method is faster than the greedy PARAFAC method we used in [31]. We also compare the groupings discovered by HITS and TOPHITS, and show that TOPHITS finds similar groupings but adds context information via the terms. This additional information can be used in query systems. We show examples of the different types of query results one can obtain.

2 Computing the TOPHITS model

The idea underlying TOPHITS is as follows. Suppose that we analyze a collection of I web pages having a total of K terms in the anchor text of all hyperlinks. Then the $I \times I \times K$ adjacency tensor $\mathcal{X}$ is defined elementwise as in (1.3). Note that the tensor $\mathcal{X}$ is generally

Algorithm 1 Greedy PARAFAC

in: Tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$.
in: Desired rank $R > 0$.
for $r = 1, \ldots, R$ **do** {outer loop}
  Set $\mathbf{v}^{(n)}$ to be a vector of all ones of length I_n for $n = 1, \ldots, N$.
  repeat {middle loop}
    for $n = 1, \ldots, N$ **do** {inner loop}
      Set $\mathbf{w} = \mathbf{X}_{(n)}\mathbf{z}^{(n)} - \sum_{i=1}^{r-1}\left(\mathbf{u}_i^{(n)} \prod_{\substack{m=1\\m\neq n}}^{N} (\mathbf{v}^{(n)})^{\mathsf{T}}\mathbf{u}_i^{(m)}\right)$ where $\mathbf{z}^{(n)} \equiv \mathbf{v}^{(1)} \otimes \cdots \otimes \mathbf{v}^{(n-1)} \otimes \mathbf{v}^{(n+1)} \otimes \cdots \otimes \mathbf{v}^{(N)}$.
      Set $\lambda_r = \|\mathbf{w}\|$.
      Set $\mathbf{v}^{(n)} = \mathbf{w}/\lambda_r$.
    end for
  until the fit ceases to improve or the maximum number of middle-loop iterations has been exceeded.
  Set $\mathbf{u}_r^{(n)} = \mathbf{v}^{(n)}$ for $n = 1, \ldots, N$.
end for
out: $\boldsymbol{\lambda} \in \mathbb{R}^R$ and $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$.

extremely sparse because most pages only point to a few other pages in the collection and each link only uses a few terms. Thus, it is reasonable to expect that the number of nonzeros in $\mathcal{X}$ is $O(I)$.

Given a value $R > 0$ (loosely corresponding to the number of distinct groupings in our data), the TOPHITS algorithm finds matrices $\mathbf{H}$ and $\mathbf{A}$, both of size $I \times R$, and a matrix $\mathbf{T}$, of size $K \times R$, to yield (1.4). Each triad $\{\mathbf{h}_r, \mathbf{a}_r, \mathbf{t}_r\}$, for $r = 1, \ldots, R$, defines a grouping of hubs, authorities, and terms by considering the entries with the highest scores in each vector; the value λ_r defines the weight of the grouping. (Without loss of generality, we assume the columns of our matrices are normalized to have unit length.)

In the remainder of this section, we describe the general N-way PARAFAC model (our problem is a 3-way problem) and how to compute it, with special emphasis on the fact that $\mathcal{X}$ is sparse.

2.1 The PARAFAC model The three-way decomposition of interest was proposed simultaneously by Harshman [23], using the name Parallel Factors or PARAFAC, and Carroll and Chang [7], using the name Canonical Decomposition or CANDECOMP. The PARAFAC decomposition should not be confused with the Tucker decomposition [50]. The goal is to decompose a given N-way array as a sum of vector outer products as shown in Figure 3.

Mathematically, the problem is stated as follows. Suppose we are given a tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$ and a desired approximation rank R. Then we wish to find matrices $\mathbf{U}^{(n)}$ of size $I_n \times R$, for $n = 1, \ldots, N$, and a weighting vector $\boldsymbol{\lambda}$ of length R, such that

$$\mathcal{X} \approx \boldsymbol{\lambda}\, [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!].$$

The *Kruskal operator* $[\![\cdot]\!]$ is shorthand for the sum of the rank one outer-products of the columns [32, 30]; in other words,

$$\boldsymbol{\lambda}\, [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!] \equiv \sum_{r=1}^{R} \lambda_r\ \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(N)}.$$

Without loss of generality, we assume that $\|\mathbf{u}_r^{(n)}\| = 1$ for all $r = 1, \ldots, R$ and $n = 1, \ldots, N$. Moreover, we typically re-order the final solution so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_R$.

Our goal is to solve the minimization problem:

$$\begin{aligned} &\min \left\| \mathcal{X} - \boldsymbol{\lambda}\, [\![\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!] \right\|^2 \\ &\text{subject to } \boldsymbol{\lambda} \in \mathbb{R}^R, \\ &\qquad \mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R} \text{ for } n = 1, \ldots, N. \end{aligned}$$

In the case of TOPHITS, $\mathcal{X}$ is a three-way array, so $N = 3$ and

$$\mathbf{H} \equiv \mathbf{U}^{(1)},\ \mathbf{A} \equiv \mathbf{U}^{(2)},\ \text{and } \mathbf{T} \equiv \mathbf{U}^{(3)}.$$

2.2 Greedy PARAFAC The notation $\mathbf{X}_{(n)}$ represents the nth *unfolding* of the tensor $\mathcal{X}$; see, e.g., [14, 3, 46]. In other words, $\mathbf{X}_{(n)}$ is simply a rearrangement of the entries of $\mathcal{X}$ into a matrix of size $I_n \times J$ with $J = \prod_{\substack{k=1\\k\neq n}}^{N} I_k$ so that the "fibers" in dimension n are arranged as the columns of the matrix. Mathemat-

Algorithm 2 Alternating Least Squares (ALS) for N-way arrays

in: Tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$.
in: Desired rank $R > 0$.
Initialize $\mathbf{U}^{(n)}$ for $n = 1, \ldots, N$ (see §2.4).
repeat {outer loop}
  for $n = 1, \ldots, N$ **do** {inner loop}

$$\mathbf{V} = \mathbf{X}_{(n)} \mathbf{Z}^{(n)} \mathbf{Y}^{(n)}, \tag{2.5}$$

$$\text{where } \mathbf{Z}^{(n)} \equiv \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \otimes \ldots \otimes \mathbf{u}_r^{(n-1)} \otimes \mathbf{u}_r^{(n+1)} \otimes \ldots \otimes \mathbf{u}_r^{(N)}, \tag{2.6}$$

$$\text{and } \mathbf{Y}^{(n)} \equiv \left(\mathbf{U}^{(1)\mathsf{T}} \mathbf{U}^{(1)} * \cdots * \mathbf{U}^{(n-1)\mathsf{T}} \mathbf{U}^{(n-1)} * \mathbf{U}^{(n+1)\mathsf{T}} \mathbf{U}^{(n+1)} * \cdots * \mathbf{U}^{(N)\mathsf{T}} \mathbf{U}^{(N)} \right)^{-1}. \tag{2.7}$$

    for r=1,…,R **do** {Assign $\mathbf{U}^{(n)}$}
      Set $\lambda_r = \|\mathbf{v}_r\|$
      Set $\mathbf{u}_r^{(n)} = \mathbf{v}_r / \lambda_r$.
    end for
  end for
until the fit ceases to improve or the maximum number of outer iterations is exceeded.
out: $\boldsymbol{\lambda} \in \mathbb{R}^R$ and $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$.

ically, we have

$$\left[\mathbf{X}_{(n)}\right]_{ij} = x_{i_1 i_2 \cdots i_N} \tag{2.8}$$
$$\text{with } i = i_n \text{ and } j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_n - 1) \prod_{\substack{\ell=1 \\ \ell \neq n}}^{k-1} I_\ell$$
$$\text{for } 1 \leq i \leq I_n, 1 \leq j \leq J.$$

In our previous work [31], we presented a greedy algorithm for computing the 3-way PARAFAC model of large, sparse tensors. Here we present the method for a general N-way array in Algorithm 1. Each outer loop iteration computes a single factor, $\{\mathbf{u}_r^{(1)}, \ldots, \mathbf{u}_r^{(N)}\}$. To compute this factor, at outer iteration r, the middle loop is an alternating least squares method that approximately minimizes

$$\left\| \left(\mathcal{X} - \sum_{i=1}^{r-1} \lambda_i \, \mathbf{u}_i^{(1)} \circ \cdots \circ \mathbf{u}_i^{(N)} \right) - \left(\mathbf{v}^{(1)} \circ \cdots \circ \mathbf{v}^{(N)} \right) \right\|$$

with respect to vectors $\mathbf{v}^{(n)} \in \mathbb{R}^{I_n}$ for $n = 1, \ldots, N$.

2.3 Alternating least squares for PARAFAC

A more common approach to solving the PARAFAC model is the use of alternating least squares (ALS) [23, 19, 49], presented in Algorithm 2. At each inner iteration, we compute the entire nth matrix $\mathbf{U}^{(n)}$ while holding all the other matrices fixed.

The $\mathbf{V}$ that is computed at each inner iteration is the solution of the following minimization problem:

$$\min_{\mathbf{V}} \left\| \mathcal{X} - [\![\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(n-1)}, \mathbf{V}, \mathbf{U}^{(n+1)}, \ldots, \mathbf{U}^{(N)}]\!] \right\|^2. \tag{2.9}$$

This can be rewritten in *matrix* form as a least squares problem [19]:

$$\min_{\mathbf{V}} \left\| \mathbf{X}_{(n)} - \mathbf{V} \mathbf{Z}^{(n)\mathsf{T}} \right\|^2. \tag{2.10}$$

Here $\mathbf{X}_{(n)}$ is the nth unfolding of the tensor $\mathcal{X}$ as shown in (2.8). The matrix $\mathbf{Z}^{(n)}$ is of size $J \times R$ and defined by (2.6). The least squares solution for (2.10) involves the pseudo-inverse of $\mathbf{Z}^{(n)}$:

$$\mathbf{V} = \mathbf{X}_{(n)} (\mathbf{Z}^{(n)\mathsf{T}})^{\dagger}.$$

Conveniently, the pseudo-inverse of $\mathbf{Z}^{(n)}$ has special structure [48, 30]. Let the $R \times R$ symmetric matrix $\mathbf{Y}^{(n)}$ be as in (2.7). Then it can be shown that [46]:

$$(\mathbf{Z}^{(n)\mathsf{T}})^{\dagger} = \mathbf{Z}^{(n)} \mathbf{Y}^{(n)\mathsf{T}}.$$

Therefore, the solution to (2.10) is given by (2.5). Thus, computing $\mathbf{U}_{(n)}$ essentially reduces to inverting the special $R \times R$ matrix $\mathbf{Y}^{(n)}$.

2.4 Initializing PARAFAC

In the large-scale case, the choice of initialization in Algorithm 2 can affect both the fit and speed of convergence. We will consider three choices for initialization.

Choice 1: Greedy PARAFAC initialization. We use Algorithm 1 to generate an initial guess that is used for Algorithm 2.

Choice 2: Random initialization. We start with a set of random values for each matrix.

Choice 3: HOSVD initialization. In this case, we consider the tensor $\mathcal{X}$ mode-by-mode. For each mode, we compute the R vectors that best span the column space of the matrix $\mathbf{X}_{(n)}$ as defined above in (2.8). This is known as the higher-order SVD, or HOSVD [14].

We compare these choices in §4.2.

2.5 Special considerations for sparse data

As we discussed at this beginning of §2, the tensor $\mathcal{X}$ is extremely sparse. Consequently, its unfolded representation $\mathbf{X}_{(n)}$ (which has the same nonzeros but reshaped) is a sparse matrix. The matrix $\mathbf{Z}^{(n)}$ from (2.6) should not be formed explicitly because it would be a dense matrix of size $I_n \times J$ where $J = \prod_{k=1, k\neq n}^{N} I_n$. Instead, the calculation of

$$\mathbf{X}_{(n)}\mathbf{Z}^{(n)}$$

needed for (2.5) must be computed specially, exploiting the inherent Kronecker product structure in $\mathbf{Z}^{(n)}$, to retain sparseness. The final result is of size $I_n \times R$ and so can be stored as a dense matrix. One method for computing this product efficiently is shown in Algorithm 3.

Algorithm 3 Computing the sparse product $\mathbf{X}_{(n)}\mathbf{Z}^{(n)}$

in: Tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$ with Q nonzeros. Let the index of the qth nonzero be $(k_{1_q}, k_{2_q}, \ldots, k_{N_q})$ and its value be given by v_q.
in: Index n and matrices $\mathbf{U}^{(m)}$ for $1 \leq m \leq N, m \neq n$.
for $r = 1\ldots, R$ **do**
  for $q = 1, \ldots, Q$ **do**
    Compute $w_q = v_q \prod_{\substack{m=1 \\ m\neq n}}^{N} u^{(m)}_{k_{m_q},r}$
  end for
  for $i = 1, \ldots, I_n$ **do** {Compute rth column of $\mathbf{P}$}
    Set $p_{ir} = \sum_{\substack{q=1 \\ k_{n_q}=i}}^{Q} w_q$.
  end for
end for
out: $\mathbf{P} = \mathbf{X}_{(n)}\mathbf{Z}^{(n)}$

3 TOPHITS and queries

Once we have computed a TOPHITS model of rank R,

$$\mathcal{X} = \boldsymbol{\lambda}[\![\mathbf{H}, \mathbf{A}, \mathbf{T}]\!],$$

we can use it for understanding the data in a variety of ways. Looking at the largest values of each triplet $\{\mathbf{h}_r, \mathbf{a}_r, \mathbf{t}_r\}$ provides a grouping of web page hubs, web page authorities, and descriptive terms, and the multiplier λ_r provides the relative weight of the grouping.

One question we can consider is the basic web search question: find all pages related to a particular term or set of terms. Consider a query vector $\mathbf{q}$ of length K (where K is the number of terms) as

$$q_k = \begin{cases} 1 & \text{if term } k \text{ is in the query,} \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } k = 1, \ldots, K.$$

Note that there is no reason to restrict ourselves to queries on terms. We can also ask the related question: find web pages and/or terms related to a particular web page or set of pages.

3.1 Finding matching groups

Rather than just returning a list of ranked pages, TOPHITS provides the option of identifying groupings that are relevant to a given query. We can create a group score vector $\mathbf{s}$ of length R that contains the score of each grouping, based on the $\mathbf{T}$ matrix from the PARAFAC model:

$$\mathbf{s} = \boldsymbol{\Lambda}\mathbf{T}^{\mathsf{T}}\mathbf{q} \quad \text{with } \boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda}). \tag{3.11}$$

Entry s_r gives the score of the rth group, and higher-scoring groupings are considered to be more relevant.

Alternatively, we can constuct a query vector based on web pages, $\hat{\mathbf{q}} \in \mathbb{R}^I$, and compute group scores as:

$$\hat{\mathbf{s}} = \boldsymbol{\Lambda}\mathbf{A}^{\mathsf{T}}\hat{\mathbf{q}} \quad \text{with } \boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda}). \tag{3.12}$$

3.2 Finding a single set of authorities

It is also possible to return a traditional ranked list of possibilities. We can combine all the information in the TOPHITS model to return a set of ranked authorities and/or hubs. Once again, let $\mathbf{s}$ be defined as in (3.11). The *combined* authorities are then given by:

$$\mathbf{a}^* = \mathbf{A}\mathbf{s} = \sum_{r=1}^{R} s_r\, \mathbf{a}_r.$$

Sorting the entries in $\mathbf{a}^*$ provides a ranked list of authorities. Likewise, the combined hubs are given by:

$$\mathbf{h}^* = \mathbf{H}\mathbf{s} = \sum_{r=1}^{R} s_r\, \mathbf{h}_r.$$

4 Experimental results

4.1 Data We generated data to test our method by using a web crawler that collected anchor text as well as link information. We started the crawler from the URLs listed in Table 1 and allowed it to crawl up to 1000 hosts and up to 500 links per page. It traversed 122,196 hyperlinks, visiting 4986 unique URLs, and identified 8109 unique anchor text terms (standard stop words were omitted). Links with no text were associated with the catch-all term "no-anchor-text."

`http://www.fivestarproduce.com/links.htm`
`http://www.tomatonet.org/news.htm`
`http://www.netweed.com/film/`
`http://infohost.nmt.edu/~armiller/food.htm`

Table 1: Seed URLs for web crawl

For simplicity, we consider host-to-host data rather than page-to-page. From our original set of 1000 hosts, we removed two sets of hosts that seemingly only had interconnections within their own sets: any host containing "`craigslist`" and any host containing "`thecityof`." Finally, we replaced any term that only appeared once in the host-to-host graph with the term "no-anchor-text." Our final host graph had 787 cross-linked hosts and 533 terms, which resulted in a sparse tensor $\mathfrak{X}$ of size $787 \times 787 \times 533$ with 3583 nonzeros. We scaled the entries so that

$$x_{ijk} = \begin{cases} \frac{1}{\log(w_k+1)} & \text{if host } i \text{ links to } j \text{ with term } k, \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } 1 \le i,j \le I = 787, \quad 1 \le k \le K = 533, \tag{4.13}$$

where w_k is the number of distinct pairs (i,j) such that a link from host i to host j uses the term k. This simple weighting reduces the biasing from prevalent terms. Other weightings are possible as well.

For our HITS results, we have a sparse matrix $\mathbf{X}$ of size 787×787 matrix with 1617 nonzeros, defined by

$$x_{ij} = \begin{cases} 1 & \text{if host } i \text{ links to host } j, \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } 1 \le i,j \le I = 787. \tag{4.14}$$

4.2 Computing PARAFAC We compared the performance of greedy PARAFAC (Algorithm 1) and three instances of PARAFAC-ALS (Algorithm 2) using the initialization schemes presented in §2.4. We calculated a rank $R = 50$ model of the tensor $\mathfrak{X}$ defined in (4.13). The *fit* of the model is defined as:

$$\frac{\| \mathfrak{X} - \boldsymbol{\lambda} [\![\mathbf{H}, \mathbf{A}, \mathbf{T}]\!] \|}{\| \mathfrak{X} \|}.$$

We terminated the iterative procedure when the *change* in fit was less than 10^{-4}.

Table 2 shows a comparison of the different methods, including the number of outer iterations for the ALS methods. For PARAFAC-ALS with random initialization, we report average results over 100 runs. All tests were performed using a 3GHz Pentium Xeon desktop computer with 2GB of RAM. Our algorithms were written in MATLAB, using Algorithm 3 for efficient computation, via sparse extensions of our Tensor Toolbox [3]. As these timings are based on prototype code in MATLAB, they are not intended to be scaled directly to estimate the time for solving larger problems. However, they provide some sense of the relative expense of the different methods.

Method	Initializ.	Fit	Time (sec)	Itns
Greedy PARAFAC	—	0.866	18.6	—
PARAFAC-ALS	Greedy	0.859	23.5	18
PARAFAC-ALS	Random	0.863	4.81	22
PARAFAC-ALS	HOSVD	0.855	11.0	15

Table 2: Comparison of different methods for computing the PARAFAC model on sparse data.

The greedy PARAFAC method requires a total of 315 inner iterations (see Algorithm 1), but this iteration count is not comparable to those for PARAFAC-ALS and so is not included in the table itself. Note also that PARAFAC-ALS with the greedy initialization is, in fact, initialized with the output of the greedy PARAFAC; thus, its total time is necessarily greater and its fit is also necessarily as good or better.

All of the methods are approximately equivalent in terms of fit, with a slight advantage going to PARAFAC-ALS with greedy or HOSVD initialization. The real difference is in computation time, and the PARAFAC-ALS methods are much faster than greedy PARAFAC, with the obvious exception being PARAFAC-ALS with greedy initialization. For comparison, using MATLAB's highly optimized `svds` function requires 1.0 seconds to compute a rank-50 SVD for the HITS approach on the matrix $\mathbf{X}$ defined in (4.14). Random initialization is clearly faster than HOSVD initialization, but we have observed that this is not the case with a tighter stopping tolerance (e.g., 10^{-6}).

Because it has the best fit and is relatively fast to compute, we use the results of PARAFAC-ALS with HOSVD initialization in the results that follow.

4.3 TOPHITS groups As in [31], we now compare the groupings found via HITS and TOPHITS, but for a different data set.

Table 3 shows several sets of authorities and hubs derived from the HITS approach [27], using the SVD applied to the matrix **X** from (4.14). We omit negative entries because they tended to be repeats of the previous positive entries.

Authorities	
Score	Host
Grouping 1 (Weight=14.63)	
0.133	www.google.com
0.104	www.yahoo.com
0.093	www.dogpile.com
0.093	www.epinions.com
0.092	dir.yahoo.com
0.091	www.ipl.org
0.066	www.realbeer.com
0.064	www.beerhunter.com
0.064	www.nws.noaa.gov
0.063	www.espressotop50.com
Grouping 2 (Weight=14.11)	
0.088	www.popmatters.com
0.087	www.hiphop-blogs.com
0.086	www.blogarama.com
0.085	pyramids2projects.blogspot.com
0.085	www.bloglet.com
0.084	ulmann.blogspot.com
0.083	news.bbc.co.uk
0.082	differentkitchen.blogspot.com
0.081	www.imdb.com
0.080	www.funkdigital.com
Grouping 3 (Weight=10.84)	
0.329	ve3d.ign.com
0.329	www.gamespyarcade.com
0.311	corp.ign.com
0.310	www.fileplanet.com
0.307	www.rottentomatoes.com
0.306	www.direct2drive.com
0.306	www.gamestats.com
0.286	www.3dgamers.com
0.283	www.gamespy.com
0.281	www.cheatscodesguides.com
Grouping 4 (Weight=9.84)	
0.110	boingboing.net
0.109	www.netweed.com
0.104	www.hiphopdx.com
0.104	www.vibe.com
0.092	www.bbc.co.uk
0.092	blacklogs.com
0.091	www.businesspundit.com
0.091	www.droxy.com
0.090	www.elhide.com
0.090	www.nytimes.com

Table 3: HITS results

Because there is some degree of sign ambiguity in the TOPHITS results, the factors are post-processed as follows. For each vector in a given triad, we looked at the maximum magnitude element. If exactly two of the three largest elements were negative, we swapped the signs of the corresponding two vectors. This means that the largest elements tend to all be positive. The change is mathematically equivalent but affects the interpretation.

Topics		Authorities	
Score	Term	Score	Host
Grouping 1 (Weight=2.37)			
0.373	models	0.997	www.wrh.noaa.gov
0.373	hydrology	0.056	www.nws.noaa.gov
0.259	aviation	0.038	iwin.nws.noaa.gov
0.255	fire	0.031	aviationweather.gov
0.255	radar	0.021	www.weather.gov
0.220	precipitation	0.021	www.goes.noaa.gov
0.213	satellite		
Grouping 2 (Weight=2.34)			
0.375	landscape	1.000	ucce.ucdavis.edu
0.375	rose		
0.375	winter		
0.375	fall		
0.375	sale		
0.326	gardening		
0.273	plant		
0.212	basics		
0.206	garden		
Grouping 3 (Weight=2.31)			
0.590	university	0.804	ucanr.org
0.510	2005	0.592	groups.ucanr.org
0.433	california	0.063	ucce.ucdavis.edu
0.356	jobs		
0.205	uc		
0.101	2003		
0.017	meeting		
0.017	dairy		
0.015	no-anchor-text		
0.014	4-h		
Grouping 10 (Weight=1.85)			
0.475	affiliate	0.996	hotjobs.yahoo.com
0.475	seeker	0.083	ca.hotjobs.yahoo.com
0.475	guidelines	0.031	www.yahoo.com
0.377	program	0.013	www.hotjobs.com
0.296	hotjobs		
0.189	job		
0.172	yahoo		
Grouping 11 (Weight=1.85)			
0.336	software	1.000	www.apple.com
0.336	notice		
0.336	hot		
0.336	support		
0.336	developer		
0.289	itunes		
0.266	pro		
0.266	ipod		
Grouping 13 (Weight=1.81)			
0.367	league	0.945	www.netweed.com
0.361	group	0.148	www.fantasymusicleague.com
0.356	trimedia	0.133	www.nydailynews.com
0.328	line	0.119	www.trimediaent.com
0.326	netweed	0.117	www.allhiphop.com
0.323	logic	0.093	www.hiphop-blogs.com
0.205	hip	0.077	ulmann.blogspot.com
0.200	hop	0.056	www.onlinemusicblog.com
0.198	blogs	0.055	www.lyricalswords.com

Table 4: TOPHITS results

Table 4 shows a sample of groupings and authorities derived from the TOPHITS approach. We omitted repetitive results, including the negative ends of the vectors. For each factor, we get a ranked list of hosts that is associated with a ranked list of terms. Although we are unable to show full results here, they are very similar to what is obtained from HITS, but TOPHITS includes terms that identify the topic of each set of authorities.

4.4 Queries with TOPHITS In this subsection we explore the use of TOPHITS for queries. In §3, we proposed two types of queries, a "max query" to find matching groupings and an "inner product query" to provide cumulative results.

Table 5 shows the results of the max query on the term "California." Three distinct groupings are identified in our data having to do with California; moreover, the score (from $\mathbf{s}$ in (3.11)) of the factor indicates how relevant the grouping is to the query. Table 6 shows the same term with the inner product query, and in this case it muddles the distinct groupings.

Topics		**Authorities**	
Score	Term	Score	Host
Grouping 1 (Score=1.00)			
0.590	university	0.804	ucanr.org
0.510	2005	0.592	groups.ucanr.org
0.433	california	0.063	ucce.ucdavis.edu
0.356	jobs		
0.205	uc		
0.101	2003		
0.017	meeting		
0.017	dairy		
0.015	no-anchor-text		
0.014	4-h		
Grouping 2 (Score=0.49)			
0.532	dui	0.796	www.duicentral.com
0.387	law	0.332	www.duicenter.com
0.374	southern	0.275	www.california-drunkdriving.org
0.352	california	0.188	www.drunkdriving-california.net
0.280	lawyers	0.185	www.california-drunkdriving.com
0.208	lawyer	0.178	www.azduiatty.com
0.183	attorney	0.172	www.california-drunkdriving.net
0.170	defense	0.144	www.dui-dwi.com
0.141	arrests	0.138	guides.california-drunkdriving.org
0.128	attorneys	0.097	www.richardessen.com
Grouping 3 (Score=0.06)			
0.476	no-anchor-text	0.860	www.realbeer.com
0.448	beer	0.344	realbeer.com
0.359	spencer's	0.282	ericsbeerpage.com
0.345	brewpubs	0.101	www.xs4all.nl
0.245	area	0.069	celebrator.com
0.239	country	0.061	worldofbeer.com
0.212	real	0.055	www.nycbeer.org
0.212	pubs	0.055	www.beerinfo.com
0.176	3	0.052	www.allaboutbeer.com
0.167	reviews	0.047	www.virtualbeer.com

Table 5: Max query on "california"

Authorities	
Score	Host
0.692	ucanr.org
0.391	www.duicentral.com
0.163	www.duicenter.com
0.135	www.california-drunkdriving.org
0.092	www.drunkdriving-california.net
0.091	www.california-drunkdriving.com
0.088	www.azduiatty.com
0.084	www.california-drunkdriving.net
0.071	www.dui-dwi.com
0.068	guides.california-drunkdriving.org

Table 6: Inner product query on "california"

Tables 7 and 8 show the results of a query on the terms "job" and "jobs." In this case, the three groupings identified by the max query have relatively similar scores, so it comes as no surprise that the results returned by the inner product query present a good mixture of job-related sites.

Topics		**Authorities**	
Score	Term	Score	Host
Grouping 1 (Score=0.82)			
0.590	university	0.804	ucanr.org
0.510	2005	0.592	groups.ucanr.org
0.433	california	0.063	ucce.ucdavis.edu
0.356	jobs		
0.205	uc		
0.101	2003		
0.017	meeting		
0.017	dairy		
0.015	no-anchor-text		
0.014	4-h		
Grouping 2 (Score=0.43)			
0.510	advice	0.998	content.monster.com
0.484	targeted	0.062	www.fastweb.com
0.441	career	0.011	learning.monster.com
0.400	basics		
0.265	job		
0.235	search		
0.112	home		
0.089	resources		
0.042	span		
0.038	div		
Grouping 3 (Score=0.35)			
0.475	affiliate	0.996	hotjobs.yahoo.com
0.475	seeker	0.083	ca.hotjobs.yahoo.com
0.475	guidelines	0.031	www.yahoo.com
0.377	program	0.013	www.hotjobs.com
0.296	hotjobs		
0.189	job		
0.172	yahoo		
0.157	home		
0.032	canada		
0.031	usa		

Table 7: Max query on "job" and "jobs"

Table 9 shows the results on a query on the terms "tomato" and "tomatoes." The highest scoring grouping is connected with the UC Tomato Genetics Resource Center. The second grouping, with a much

Authorities	
Score	Host
0.569	ucanr.org
0.424	content.monster.com
0.348	hotjobs.yahoo.com
0.215	hiring.monster.com
0.031	www.fastweb.com
0.029	ca.hotjobs.yahoo.com
0.027	my.monster.com
0.020	ucce.ucdavis.edu
0.011	www.allhiphop.com
0.011	www.yahoo.com

Table 8: Inner product query on "job" and "jobs"

lower score, is connected to gaming sites, including the site www.rottentomatoes.com, which is sometimes returned by search engines for a search on the term "tomatoes." The final grouping, with a very low score, is interesting because it picks up a grouping about vegetables in general.

Topics		Authorities	
Score	Term	Score	Host
Grouping 1 (Score=0.50)			
0.765	rick	0.990	tgrc.ucdavis.edu
0.434	center	0.141	wric.ucdavis.edu
0.432	tomato		
0.180	research		
0.068	no-anchor-text		
0.045	weed		
0.027	information		
Grouping 2 (Score=0.02)			
0.575	policy	0.995	corp.ign.com
0.497	privacy	0.063	cheats.ign.com
0.379	ign	0.037	www.fileplanet.com
0.315	0	0.032	www.rottentomatoes.com
0.308	entertainment	0.028	www.gamestats.com
0.286	no-anchor-text	0.023	www.gamespy.com
0.030	cheats	0.022	www.3dgamers.com
0.018	gamestats	0.022	guides.ign.com
0.014	tomatoes	0.020	www.direct2drive.com
0.014	codes	0.020	ve3d.ign.com
Grouping 3 (Score=0.01)			
0.596	vric	0.998	vric.ucdavis.edu
0.458	publications	0.032	www.ctga.org
0.363	vegetable	0.030	www.ag.ohio-state.edu
0.319	current	0.028	www.kdcomm.net
0.312	notes	0.025	www.tomatonews.com
0.258	uc	0.021	ceyolo.ucdavis.edu
0.166	www	0.015	www.wrh.noaa.gov
0.094	no-anchor-text		
0.011	ag		

Table 9: Max query on "tomato" and "tomatoes"

We can adapt the score discussed in §3.1 to input hosts rather than terms, by swapping $\mathbf{T}$ for $\mathbf{A}$. Table 10 shows the results for a "host max query" using the host www.google.com. The primary grouping includes Google sites as well as sites about Google.

Topics		Authorities	
Score	Term	Score	Host
Grouping 1 (Score=1.08)			
0.962	google	0.989	www.google.com
0.165	programs	0.071	google.blogspace.com
0.133	haiku	0.051	www.seochat.com
0.116	home	0.046	catalogs.google.com
0.073	no-anchor-text	0.045	www.google-watch.org
0.062	business	0.045	www.researchbuzz.org
0.056	search	0.045	www.lagcc.cuny.edu
0.041	page	0.045	www.googlealert.com
0.032	site	0.045	www.googlefight.com
0.029	http	0.027	news.google.com

Table 10: Max query on "www.google.com"

5 Conclusions & future work

TOPHITS is an extension of HITS [27] that incorporates anchor text into a third dimension. In this paper, we have shown the following:

- The TOPHITS factors can be calculated efficiently by careful implementation of sparse tensor operations.
- TOPHITS provides grouping information that can be used as part of a query-response system. Moreover, the groupings in the TOPHITS model provide a natural grouping of results.

Like HITS [27], TOPHITS produces both positive and negative entries in its factors. In these results, the negative factors have proved to be insignificant; however, more sophisticated techniques for handling the negative entries is needed. The three-way nature of the decomposition means that there is ambiguity in terms of the negativity that can not be easily resolved. We have experimented with non-negative factorizations for tensors [35, 39] but found them to be ineffective on our data. We conjecture that better methods for calculating non-negative factors may produce better results.

We will need to investigate the stability of TOPHITS under small perturbations to the hyperlink patterns, as has been done by Ng et al. [40, 41] for PageRank and HITS. Moreover, we would add the question of stability with respect to the rank R of the TOPHITS model (1.4), which can have a profound effect on the PARAFAC model [19].

Many existing methods could potentially be extended to the multidimensional case. For example, enhancements for HITS and PageRank could also be extended to TOPHITS, including hub and authority thresholding for HITS [6] and optimizations for accelerating computation of the PageRank score [38]. In terms of applications, TOPHITS may be useful, in the same way as HITS, in partitioning the web into tightly inter-connected groupings [20, 28, 25]. Alternatively, multidimensional models of trust could extend the trust propagation work of Guha et al. [22]. We may also exploit the LSI-like features of TOPHITS. Dasgupta et al. [12] developed a query-dependent version of LSI; in principal, their adaptation of LSI could be applied to TOPHITS to improve its responsiveness to queries.

There is also no reason why TOPHITS need be restricted to anchor text. More complex structure information could be incorporated, especially semantic structure [2, 42]. The third dimension can be used alternatively to capture other types information such as the *type* of connection, which might be available in a semantic web setting. Furthermore, we are not limited to three dimensions but may use as many dimensions as needed.

Acknowledgments

We gratefully acknowledge Ken Kolda for writing the *Web Krawler* application that we have used for our data collection.

References

[1] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, and B. Yener. Modeling and multiway analysis of chatroom tensors. In *Intelligence and Security Informatics: IEEE Intl. Conf. on Intelligence and Security Informatics, ISI 2005 19-20, 2005*, LNCS 3495, pp. 256–268, 2005.

[2] K. Anyanwu, A. Maduko, and A. Sheth. SemRank: ranking complex relationship search results on the semantic web. In *WWW '05*, pp. 117–127. ACM Press, 2005.

[3] B. W. Bader and T. G. Kolda. MATLAB tensor classes for fast algorithm prototyping. Technical Report SAND2004-5187, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, Oct. 2004.

[4] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4):573–595, 1995.

[5] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR '98*, pp. 104–111. ACM Press, 1998.

[6] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *WWW '01*, pp. 415–429. ACM Press, 2001.

[7] 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.

[8] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *WWW7*, pp. 65–74. Elsevier, 1998.

[9] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the Web's link structure. *Computer*, 32(8):60–67, 1999.

[10] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *ICML '00: Proc. 17th Int. Conf. on Machine Learning*, pp. 167–174. Morgan Kaufmann Publishers Inc., 2000.

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

[12] A. Dasgupta, R. Kumar, P. Raghavan, and A. Tomkins. Variable latent semantic indexing. In *KDD '05*, pp. 13–21. ACM Press, 2005.

[13] B. D. Davison, A. Gerasoulis, K. Kleisouris, Y. Lu, H.-J. Seo, W. Wang, and B. Wu. DiscoWeb: applying link analysis to web search. Poster at WWW8, May 1999. Available from `http://www.cse.lehigh.edu/~brian/pubs/1999/www8/`.

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. A.*, 21(4):1253–1278, 2000.

[15] C. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon. PageRank, HITS and a unified framework for link analysis. In *SIGIR '02*, pp. 353–354. ACM Press, 2002.

[16] S. T. Dumais. Latent semantic analysis. *Annu. Rev. Inform. Sci.*, 38:189–230, 2004.

[17] 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: CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 281–285. ACM Press, 1988.

[18] N. Eiron and K. S. McCurley. Analysis of anchor text for web search. In *SIGIR '03*, pp. 459–460. ACM Press, 2003.

[19] N. K. M. Faber, R. Bro, and P. K. Hopke. Recent developments in CANDECOMP/PARAFAC algorithms: a critical review. *Chemometrics and Intelligent Laboratory Systems*, 65(1):119–137, Jan. 2003.

[20] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *HYPERTEXT '98: Proceedings of the ninth ACM conference on Hypertext and hypermedia*, 1998.

[21] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Univ. Press, 1996.

[22] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW '04*, pp. 403–412, New York, NY, USA, 2004. ACM Press.

[23] R. A. Harshman. Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-modal factor analysis. *UCLA working papers in phonetics*, 16:1–84, 1970.

[24] M. R. Henzinger. Hyperlink analysis for the web. *IEEE Internet Comput.*, 5(1):45–50, 2001.

[25] M. R. Henzinger. Algorithmic challenges in web search

engines. *J. Internet Mathematics*, 1(1):115–126, 2003.

[26] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[27] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[28] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The web as a graph: measurements, models, and methods. In *Proceedings of Computing and Combinatorics. 5th Annual International Conference, COCOON'99*, pp. 1–17. Springer-Verlag, 1999.

[29] T. G. Kolda. Orthogonal tensor decompositions. *SIAM J. Matrix Anal. A.*, 23(1):243–255, July 2001.

[30] T. G. Kolda. Multilinear operators for higher-order decompositions. in preparation, 2006.

[31] T. G. Kolda, B. W. Bader, and J. P. Kenny. Higher-order web link analysis using multilinear algebra. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 242–249, 2005. In press.

[32] J. B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.*, 18(2):95–138, 1977.

[33] A. N. Langville and C. D. Meyer. Deeper inside PageRank. *J. Internet Mathematics*, 1(3):335–380, 2005.

[34] A. N. Langville and C. D. Meyer. A survey of eigenvector methods for web information retrieval. *SIAM Rev.*, 47(1):135–161, 2005.

[35] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 21 Oct. 1999.

[36] R. Lempel and S. Moran. SALSA: the stochastic approach for link-structure analysis. *ACM Trans. Inf. Syst.*, 19(2):131–160, 2001.

[37] L. Li, Y. Shang, and W. Zhang. Improvement of HITS-based algorithms on web documents. In *WWW '02*, pp. 527–535. ACM Press, 2002.

[38] F. McSherry. A uniform approach to accelerated PageRank computation. In *WWW '05*, pp. 575–582, New York, NY, USA, 2005. ACM Press.

[39] M. Mørup, L. K. Hansen, and S. M. Arfred. Decomposing the inter trial phase coherence of EEG in time-frequency plots using non-negative matrix and multi-way factorization. Submitted to *Human Brain Mapping*, 2005.

[40] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Link analysis, eigenvectors and stability. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 903–910, 2001.

[41] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *SIGIR '01*, pp. 258–266. ACM Press, 2001.

[42] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: bringing order to web objects. In *WWW '05*, pp. 567–574. ACM Press, 2005.

[43] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the Web. Technical Report 1999-66, Stanford Digital Library Technologies Project, 1999.

[44] D. Rafiei and A. O. Mendelzon. What is this page known for? Computing Web page reputations. *Comput. Networks*, 33(1-6):823–835, 2000.

[45] M. Richardson and P. Domingos. The intelligent surfer: probabilistic combination of link and content information in PageRank. In *Advances in Neural Information Processing Systems 14*, pp. 1441–1448. MIT Press, 2001.

[46] A. Smilde, R. Bro, and P. Geladi. *Multi-way analysis: applications in the chemical sciences*. Wiley, 2004.

[47] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen. CubeSVD: a novel approach to personalized Web search. In *WWW '05*, pp. 382–390, 2005.

[48] G. Tomasi and R. Bro. A comparison of algorithms for fitting the PARAFAC model. *Computational Statistics & Data Analysis*, 2005.

[49] G. Tomasi and R. Bro. PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems*, 75(2):163–180, Feb. 2005.

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

[51] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In *ECCV'02*, LNCS 2350, pp. 447–460. Springer-Verlag, 2002.