Higher-Order Web Link Analysis Using Multilinear Algebra

Tamara G. Kolda, Brett W. Bader, and Joseph P. Kenny
Sandia National Laboratories
Livermore, CA and Albuquerque, NM
{tgkolda,bwbader,jpkenny}@sandia.gov

Abstract

Linear algebra is a powerful and proven tool in web search. Techniques, such as the PageRank algorithm of Brin and Page and the HITS algorithm of Kleinberg, score web pages based on the principal eigenvector (or singular vector) of a particular non-negative matrix that captures the hyperlink structure of the web graph. We propose and test a new methodology that uses multilinear algebra to elicit more information from a higher-order representation of the hyperlink graph. We start by labeling the edges in our graph with the anchor text of the hyperlinks so that the associated linear algebra representation is a sparse, three-way tensor. The first two dimensions of the tensor represent the web pages while the third dimension adds the anchor text. We then use the rank-1 factors of a multilinear PARAFAC tensor decomposition, which are akin to singular vectors of the SVD, to automatically identify topics in the collection along with the associated authoritative web pages.

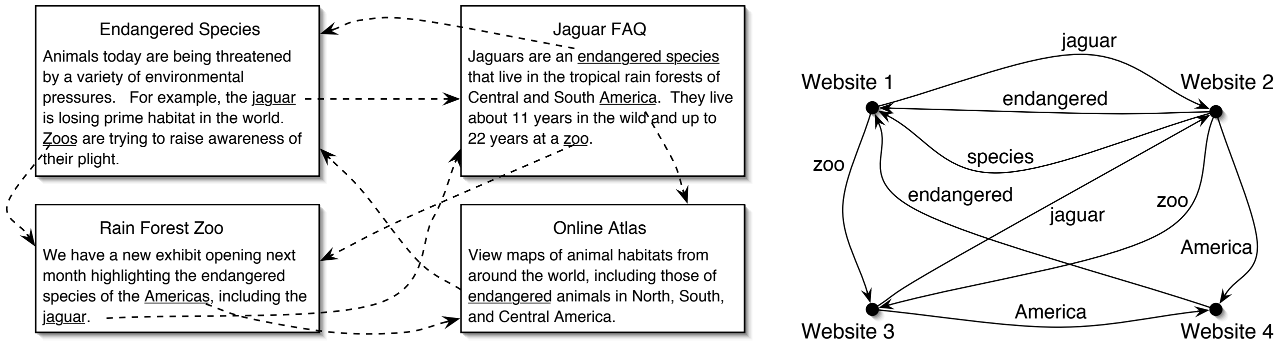
1. Introduction

PageRank [6, 32], which underlies the Google and Yahoo! search engines, and HITS [24] are two significant algorithms for determining the importance of web pages. The PageRank [6, 32] scores are given by the entries of the principal eigenvector of a Markov matrix of page transition probabilities across the entire web (see, e.g., [26] for a detailed description of PageRank). Thus, PageRank is a global score that depends only on the topology of the Web and does not take page content or the query into account. Query responses are compiled by combining the PageRank score with other heuristics that ensure a good term match. Occasionally, this can lead to peculiar query responses; for example, the top site currently returned by Google for a search on "tomatoes" is `http://www.rottentomatoes.com`, a website that rates movies.

HITS [24], on the other hand, first compiles a focused subgraph of the Web that is assumed to be "rich in relevant pages." The principal singular vectors of the adjacency matrix of the focused subgraph define the best authorities and hubs for the query. The HITS score is query-specific in that it computes the authority scores of the pages after it compiles a subset of web pages. Unfortunately, Kleinberg [24] and others [4, 10] have observed that the authorities and hubs do not always match the original query due to "topic drift," i.e., nodes in the focused subgraph are not related to the query topic. Appropriate authorities and hubs generally appear in some pair of singular vectors, but Davison et al. [12] note that selecting the appropriate singular vectors is an open research question.

Both PageRank [6, 32] and HITS [24] use appropriate eigenvectors (or singular vectors) to compute the authority of web pages and can be considered as members of the same family [14]. Other methods adhere to the same basic theme. For example, SALSA is a variant on HITS that uses a stochastic iteration matrix [27].

In this paper, we propose a new method called Topical Hypertext Induced Topic Selection (TOPHITS), following Kleinberg [24]. This new technique analyzes a semantic graph that combines anchor text with the hyperlink structure of the web. Anchor text is useful for web search because it behaves as a "consensus title" [18]. Fig. 1 shows four hypothetical web pages and the corresponding semantic graph. The adjacency structure of a semantic graph cannot be modeled as a matrix without losing edge type information. Instead, it is modeled by a *three-way tensor* containing both hyperlink and anchor text information; see Fig. 2. Then we apply the Parallel Factors (PARAFAC) decomposition [22], which is a higher-order analogue of the SVD, to get the most significant factors that are akin to singular vectors. Instead of pairs of vectors containing authority and hub scores, we produce triplets of vectors with authority and hub scores for the pages as well as topic scores for the terms. This is an extension of Kleinberg's HITS algorithm [24], which uses the singular vectors of the hyperlink matrix (a two-way tensor) to produce multiple sets of hubs and authorities. The addition of the topic vector means that determining which set of singular vectors contains the answer to the query is just a



Endangered Species

Animals today are being threatened by a variety of environmental pressures. For example, the jaguar is losing prime habitat in the world. Zoos are trying to raise awareness of their plight.

Jaguar FAQ

Jaguars are an endangered species that live in the tropical rain forests of Central and South America. They live about 11 years in the wild and up to 22 years at a zoo.

Rain Forest Zoo

We have a new exhibit opening next month highlighting the endangered species of the Americas, including the jaguar.

Online Atlas

View maps of animal habitats from around the world, including those of endangered animals in North, South, and Central America.

Figure 1. The web pages on the left yield the semantic graph on the right. The edges of the graph are labeled with the anchor text of the links.

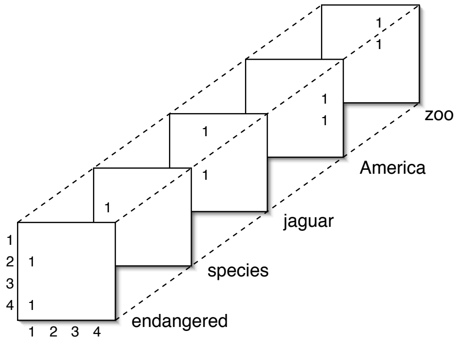


Figure 2. A three-way tensor that models the semantic graph in Fig. 1.

matter of looking at which topic vectors have a high score for the query terms and then considering the corresponding hubs and authorities. Like PageRank, TOPHITS is query-independent because the computation of the significant vectors can potentially be done in advance and off-line. This approach incorporates ideas from Latent Semantic Analysis (LSA) [17, 3, 16, 15], a popular method in text retrieval that uses dimensionality reduction to improve search. LSA has been used in many domains, including term suggestions for online advertisers [20].

2. Related Work

2.1. Topic Drift

The problem of topic drift in HITS has been addressed 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 [4] and Li et al. [28] incorporate weighting based on the content of the web pages.

2.2. Incorporating Text Information

We are not the first to propose the simultaneous analysis of hyperlink structure and anchor text or page content. Diligenti et al. [13] propose a modification of PageRank that uses a topic classifier instead of the random surfer model. Rafiei and Mendelzon [33] 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. Haveliwala [23] introduced a topic-sensitive PageRank that pre-computes several PageRank vectors that are biased towards particular topics. Richardson and Domingos [34] 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.

2.3. Our Contribution

Our contribution is the use of a PARAFAC decomposition [22] (also known as the Canonical Decomposition or CANDECOMP decomposition [7]) on a three-way tensor representing the web graph with anchor-text-labeled edges. Tensor decompositions have a long history and have been used in applications ranging from chemometrics [36] to image analysis [39]. Recently, they have been applied to data-centric problems including analysis of clickthrough data using an alternate decomposition known as Tucker [37], and chatroom analysis comparing different tensor decompositions [1].

3. Problem Setting & Evaluation

3.1. HITS

We briefly review the HITS [24] method. Let n denote the number of pages in our web (sub-)graph. Every page has a hub score $(\mathbf{h})$ and an authority score $(\mathbf{a})$, which are computed iteratively as follows:

$$\begin{aligned} \mathbf{h}_i^{(t+1)} &= \sum_{i \to j} \mathbf{a}_j^{(t)} \qquad \text{for} \quad i = 1, \ldots, n, \quad \text{and} \\ \mathbf{a}_j^{(t+1)} &= \sum_{i \to j} \mathbf{h}_i^{(t+1)} \quad \text{for} \quad j = 1, \ldots, n, \end{aligned} \tag{1}$$

The iterates $\mathbf{h}$ and $\mathbf{a}$ are normalized after each iteration. In words, the hub score of page i is equal to the sum of the authority scores of all the pages to which it points; conversely, the authority score of page i is equal to the sum of the hub scores of all pages that point to it.

Equivalently, these equations can be expressed in matrix form. Let $\mathbf{A}$ denote the $n \times n$ adjacency matrix of our graph, defined by

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } i \to j, \\ 0 & \text{otherwise}, \end{cases}$$

where $i \to j$ denotes that page i links to page j. The equations in (1) become

$$\mathbf{h}^{(t+1)} = \mathbf{A}\, \mathbf{a}^{(t)}, \quad \text{and} \quad \mathbf{a}^{(t+1)} = \mathbf{A}^T\, \mathbf{h}^{(t+1)}. \tag{2}$$

Under appropriate conditions, these iterates converge to the principal singular vectors of the adjacency matrix, cf. [21].

Recall that the first p factors of the singular value decomposition (SVD) of $\mathbf{A}$ yield the best rank-p approximation, assuming $p < \text{rank}(A)$ [21]. Thus, we can approximate $\mathbf{A}$ as

$$\mathbf{A} \approx \sum_{i=1}^{p} \sigma^{(i)}\, \mathbf{u}^{(i)} \circ \mathbf{v}^{(i)}. \tag{3}$$

Here $\sigma^{(1)} \geq \sigma^{(2)} \geq \cdots \geq \sigma^{(p)} > 0$ are the first p singular values, and $\mathbf{u}^{(i)}$ and $\mathbf{v}^{(i)}$ are the corresponding singular vectors. The notation $\mathbf{a} \circ \mathbf{b}$ denotes the vector outer product so that $(\mathbf{a} \circ \mathbf{b})_{ij} = \mathbf{a}_i \mathbf{b}_j$. See Fig. 3 for an illustration of the SVD.

As mentioned above, the iterates defined in (2) converge to the principal singular vectors:

$$\mathbf{h}^{(t)} \to \mathbf{h}^* = \mathbf{u}^{(1)} \quad \text{and} \quad \mathbf{a}^{(t)} \to \mathbf{a}^* = \mathbf{v}^{(1)}.$$

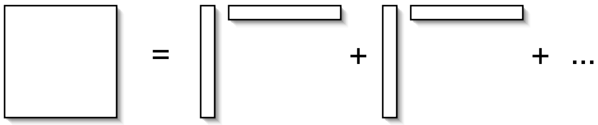


Figure 3. In HITS model, the SVD provides a 2-way decomposition that yields authority and hub scores.

Furthermore, each pair $\{\mathbf{u}^{(i)}, \mathbf{v}^{(i)}\}$ identifies a set of related authorities and hubs for the graph [24]. Our new method, described in the next section, discovers triplets of vectors that identify a topic (described by key terms) along with its associated hubs and authorities.

3.2. TOPHITS

The TOPHITS method produces sets of triplets $\{\mathbf{u}^{(i)}, \mathbf{v}^{(i)}, \mathbf{w}^{(i)}\}$ where the $\mathbf{u}$ and $\mathbf{v}$ vectors contain hub and authority scores for the web pages as in HITS, and the $\mathbf{w}$ vector contains topic scores for the terms.

Just like HITS, these scores can be computed iteratively. Let n denote the number of pages and m the number of terms. The hub, authority, and topic scores are updated as follows:

$$\begin{aligned} \mathbf{h}_i^{(t+1)} &= \sum_{i \xrightarrow{k} j} \mathbf{a}_j^{(t)} \mathbf{t}_k^{(t)} \qquad \text{for} \quad i = 1, \ldots, n, \\ \mathbf{a}_j^{(t+1)} &= \sum_{i \xrightarrow{k} j} \mathbf{h}_i^{(t+1)} \mathbf{t}_k^{(t)} \qquad \text{for} \quad j = 1, \ldots, n, \\ \mathbf{t}_k^{(t+1)} &= \sum_{i \xrightarrow{k} j} \mathbf{a}_j^{(t+1)} \mathbf{h}_i^{(t+1)} \quad \text{for} \quad k = 1, \ldots, m. \end{aligned} \tag{4}$$

Here, the notation $i \xrightarrow{k} j$ means page i links to page j with anchor text k. As with HITS, we normalize after each iteration. In words, the hub score of page i is the sum of authority scores for pages that i points to multiplied by the corresponding topic scores of the terms in the anchor text. Similarly, the authority score of page j is the sum of hub scores of all pages that point to j multiplied by the topic scores of the corresponding terms in the anchor text. The topic score of term k is the sum of hub scores for page i multiplied by the authority scores for page j over all hyperlinks $i \to j$ that involve term k in the anchor text.

This can be written in tensor form as follows. Let $\mathbf{A}$ denote the $n \times n \times m$ adjacency *tensor* of a web (sub-)graph, defined by

$$\mathbf{A}_{ijk} = \begin{cases} 1 & \text{if } i \to j \text{ with anchor text } k, \\ 0 & \text{otherwise.} \end{cases}$$

Then the equations in (4) can be expressed as:

$$\begin{aligned}\mathbf{h}^{(t+1)} &= \mathbf{A}\,\bar{\times}_2\,\mathbf{a}^{(t)}\,\bar{\times}_3\,\mathbf{t}^{(t)},\\ \mathbf{a}^{(t+1)} &= \mathbf{A}\,\bar{\times}_1\,\mathbf{h}^{(t+1)}\,\bar{\times}_3\,\mathbf{t}^{(t)},\\ \mathbf{t}^{(t+1)} &= \mathbf{A}\,\bar{\times}_1\,\mathbf{h}^{(t+1)}\,\bar{\times}_2\,\mathbf{a}^{(t+1)}.\end{aligned}\tag{5}$$

The notation $\mathbf{A}\,\bar{\times}_i\,\mathbf{x}$ indicates that the tensor $\mathbf{A}$ should be multiplied by the vector $\mathbf{x}$ in dimension i. For example,

$$\mathbf{h} = \mathbf{A}\,\bar{\times}_2\,\mathbf{a}\,\bar{\times}_3\,\mathbf{t}$$

says to multiply $\mathbf{A}$ by $\mathbf{a}$ in the second dimension and by $\mathbf{t}$ in the third dimension. The result is

$$\mathbf{h}_i = \sum_{j=1}^{n}\sum_{k=1}^{m} \mathbf{A}_{ijk}\mathbf{a}_j\mathbf{t}_k \quad \text{for } i = 1,\ldots,n.$$

(See [2] for more details on notation.) Under appropriate conditions, these iterations will converge to the best rank-1 approximation of $\mathbf{A}$.

In Section 4.3, we describe a method for computing a PARAFAC decomposition [22] of $\mathbf{A}$ which yields a rank-p approximation of a tensor $\mathbf{A}$ of the form

$$\mathbf{A} \approx \sum_{i=1}^{p} \sigma^{(i)}\,\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)} \circ \mathbf{w}^{(i)},\tag{6}$$

where $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ indicated a three-way outer product so that $(\mathbf{a} \circ \mathbf{b} \circ \mathbf{c})_{ijk} = \mathbf{a}_i\mathbf{b}_j\mathbf{c}_k$. Fig. 4 shows an illustration of the PARAFAC decomposition.

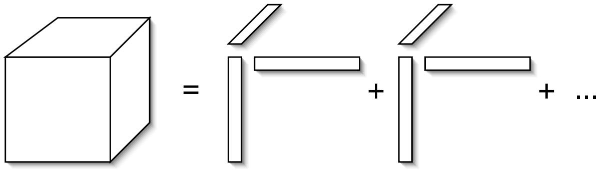


Figure 4. In TOPHITS, the PARAFAC model provides a 3-way decomposition that yields authority, hub, and topic scores.

Unlike the SVD, there is no guarantee that the rank-p PARAFAC approximation will be optimal [25]. Furthermore, the PARAFAC vectors are not orthogonal; i.e., $\mathbf{u}^{(1)}$ is not orthogonal to $\mathbf{u}^{(2)}$, as would be the case for the SVD.

However, the algorithm in Section 4.3, under suitable conditions, computes the best rank-1 approximation of $\mathbf{A}$ as the first factor so that

$$\mathbf{h}^{(t)} \to \mathbf{h}^* = \mathbf{u}^{(1)}, \quad \mathbf{a}^{(t)} \to \mathbf{a}^* = \mathbf{v}^{(1)}, \quad \mathbf{t}^{(t)} \to \mathbf{t}^* = \mathbf{w}^{(1)}.$$

The largest entries in $\mathbf{w}^{(1)}$ define the dominant topic terms for the first factor, while the largest entries in the $\mathbf{u}^{(1)}$ and $\mathbf{v}^{(1)}$ vectors define the dominant hubs and authorities for the topic. Each factor of (6), i.e., $\{\mathbf{u}^{(i)}, \mathbf{v}^{(i)}, \mathbf{w}^{(i)}\}$, yields another topic and corresponding hubs and authorities.

4. Methodology

4.1. The Data

We tested our technique on a subset of web data, generated using an in-house web crawler that includes anchor text in its output. Stop words, punctuation, and non-integer numbers were removed. Any hyperlink without anchor text was assigned the term “no-anchor-text”. In order to avoid the edge effects inherent in small web crawls, it was assumed that URLs with no recorded outlinks were never crawled and so were excluded from the data set. Finally, we considered only host-to-host links (rather than page-to-page links) and removed all self-links that point from one host to the same host.

The three-way tensor is computed by counting the number of links from host i to host j with term k and storing the result as $\mathbf{C}_{ijk}$. We perform an element-wise scaling of $\mathbf{C}$, which attenuates the influence of highly linked hosts:

$$\mathbf{A}_{ijk} = \begin{cases} 1 + \log(\mathbf{C}_{ijk}) & \text{if } \mathbf{C}_{ijk} \neq 0,\\ 0 & \text{otherwise.}\end{cases}\tag{7}$$

4.2. Working with Sparse Tensors

Working with multi-way data is a challenge due to the lack of available software. Although a few packages do exist for working with dense tensors (see, e.g., [2]), nothing is available for sparse tensors. Our web graph data is extremely sparse. For example, storing a host graph with 10,000 hosts and 10,000 terms in a dense tensor storage format would require one trillion entries, which rules out any type of dense storage format. Thus, in order to work with this data in sparse form, we developed the capability to mathematically manipulate sparse, large-scale tensors.

We implemented our methods in MATLAB by extending our existing toolbox of dense tensor classes [2], details of which will be in a forthcoming report. We have created a `sparse_tensor` object (or class) in MATLAB that stores the data in sparse format and can efficiently manipulate it. We support multiplication, scaling, accumulation across dimensions, operations on individual elements, and permutations in addition to standard operations like adding, subtracting, etc. For example, we have been able to run the greedy PARAFAC algorithm (described in Section 4.3) to work with data sets as large as 50,000 by 50,000 by 50,000 with 500,000 nonzeros on a laptop. In addition, to the `sparse_tensor` class, we have a separate class for storing a PARAFAC decomposition.

Efficiency is achieved by carefully selecting a storage format and using built-in MATLAB functions to avoid any loops. We use a coordinate-based storage scheme in which each non-zero is stored along with its indices; e.g., we store

i, j, k, and A_{ijk}. This proved to be more feasible than any type of compressed format, as is often used for sparse matrices, because tensor manipulations require indexing by each dimension.

4.3. Algorithm

To determine the leading factors of our TOPHITS method, we compute a low-rank, approximate PARAFAC decomposition (6) of the sparse tensor $\mathbf{A}$. As outlined above, we use the iteration defined by (5), which is called the higher-order power method [36], to compute the best rank-1 tensor that minimizes the Frobenius norm of the difference from $\mathbf{A}$. Computing an approximation to the best rank-p decomposition is simply a matter of iterating on this concept. To compute the kth rank-1 tensor, we apply the higher-order method to the current residual:

$$\mathbf{R}^{(k)} = \mathbf{A} - \sum_{i=1}^{k-1} \sigma^{(i)}\ \mathbf{u}^{(i)} \circ \mathbf{v}^{(i)} \circ \mathbf{w}^{(i)}.$$

We avoid computing the residual explicitly by instead computing the products (e.g., $\mathbf{R} \mathbin{\bar{\times}_1} \mathbf{x} \mathbin{\bar{\times}_2} \mathbf{y}$) on each term individually. Thus, each iteration of the power method on $\mathbf{R}^{(k)}$ involves three tensor-vector-vector products with $\mathbf{A}$ and then $4(k-1)$ vector inner products.

```
In: $\mathbf{A}$ of size $n \times n \times m$.
Out: Rank-$p$ approximation of $\mathbf{A}$, returned as $p$ triplets
$\{\mathbf{u}^{(i)}, \mathbf{v}^{(i)}, \mathbf{w}^{(i)}\}$ plus weights $\sigma^{(i)}$ for $i = 1, \ldots, p$.
For $k = 1, 2, \ldots, p$, do:
    Initialize $\mathbf{x}, \mathbf{y}, \mathbf{z}$ to be vectors of all ones of length $n, n, m$, resp.
    Repeat:
        $\mathbf{x} = \mathbf{A} \mathbin{\bar{\times}_2} \mathbf{y} \mathbin{\bar{\times}_3} \mathbf{z} - \sum_{i=1}^{k-1} \sigma^{(i)} \mathbf{u}^{(i)} (\mathbf{y}^T \mathbf{v}^{(i)})(\mathbf{z}^T \mathbf{w}^{(i)})$
        $\mathbf{y} = \mathbf{A} \mathbin{\bar{\times}_1} \mathbf{x} \mathbin{\bar{\times}_3} \mathbf{z} - \sum_{i=1}^{k-1} \sigma^{(i)} \mathbf{v}^{(i)} (\mathbf{x}^T \mathbf{u}^{(i)})(\mathbf{z}^T \mathbf{w}^{(i)})$
        $\mathbf{z} = \mathbf{A} \mathbin{\bar{\times}_1} \mathbf{x} \mathbin{\bar{\times}_2} \mathbf{y} - \sum_{i=1}^{k-1} \sigma^{(i)} \mathbf{w}^{(i)} (\mathbf{x}^T \mathbf{u}^{(i)})(\mathbf{y}^T \mathbf{v}^{(i)})$
        $\lambda = \|\mathbf{x}\|\ \|\mathbf{y}\|\ \|\mathbf{z}\|$, and normalize $\mathbf{x}, \mathbf{y}, \mathbf{z}$
    Until the change in $\lambda$ is small.
    Set $\mathbf{u}^{(k)} = \mathbf{x}$, $\mathbf{v}^{(k)} = \mathbf{y}$, $\mathbf{w}^{(k)} = \mathbf{z}$, $\sigma^{(k)} = \lambda$
End do.
```

Figure 5. Greedy PARAFAC Algorithm

The complete algorithm is in Fig. 5. We call this procedure the greedy PARAFAC decomposition because it calculates a rank-1 factor to $\mathbf{R}^{(k)}$ without considering changes to the factors previously computed. An alternative method employs a different alternating least squares approach that simultaneously solves for all vectors in the same mode (e.g., all $\mathbf{u}^{(i)}, 1 \leq i \leq p$) [36]. However, in our experiences, such an approach is slow to converge and does not yield any significant improvements in our results over the greedy approach.

5. Results

We started our web crawler from the following URL: `http://www-neos.mcs.anl.gov/neos` (an optimization web page) and allowed it to crawl 4700 pages, resulting in 560 cross-linked hosts.

Fig. 6 shows the authorities derived from the HITS approach [24], using the SVD applied to the standard adjacency matrix (i.e., $A_{ij} = 1$ if $i \rightarrow j$). We show results from the first several singular vectors, omitting negative entries because they were repeats of earlier sets of authorities and other sets that were also repeats (e.g., the fifth singular vector contained repeats from several of the first four vectors).

Using our greedy PARAFAC algorithm from Fig. 5 on the tensor $\mathbf{A}$ defined by (7), we computed the first twenty factors of the scaled adjacency tensor. The cost of each iteration is $O(N)$ where N is the number of nonzeros in the tensor $\mathbf{A}$. This is approximately the same cost of each iteration of the power method for computing the SVD because the number of nonzeros in the tensor representation is not much more than that in the matrix representation. Fig. 7 shows that we only require a few iterations for each factor.

Figures 8 and 9 show sets of topics and authorities derived from the TOPHITS approach. As before, we omitted repetitive results. For each factor, we get a ranked list of hosts that is associated with a ranked list of terms. The results are very similar to what we get from HITS, but TOPHITS includes terms that identify the topic of each set of authorities. In the simplest case, this approach can be used to correct the topic drift problem. Here, for example, we collected pages about optimization as well as other topics. It is easy to find the authorities on optimization by simply searching for key terms (in this case, “optimization” identifies the 12th factor).

The usefulness of this new TOPHITS approach is that it automatically discovers topics along with sets of authorities. This can be used to extend HITS so that it can be used on large, multi-topic data sets.

6. Conclusions

Multi-way data representations and tensor decompositions are a novel technique for web search and related tasks. We have introduced the TOPHITS algorithm, which extends HITS [24] by identifying hubs and authorities that are associated with prominent topics. We accomplish this

Authorities	
SCORE	HOST
1st Singular Vector	
0.97	www.ibm.com
0.24	www.alphaworks.ibm.com
0.08	www-128.ibm.com
0.05	www.developer.ibm.com
0.02	www.research.ibm.com
2nd Singular Vector	
0.99	www.lehigh.edu
0.11	www2.lehigh.edu
0.06	www.lehighalumni.com
0.06	www.lehighsports.com
3rd Singular Vector	
0.75	java.sun.com
0.38	www.sun.com
0.36	developers.sun.com
0.24	see.sun.com
0.16	www.samag.com
0.13	docs.sun.com
0.12	blogs.sun.com
0.08	sunsolve.sun.com
0.08	www.sun-catalogue.com
0.08	news.com.com
4th Singular Vector	
0.60	www.pueblo.gsa.gov
0.45	www.whitehouse.gov
0.35	www.irs.gov
0.31	travel.state.gov
0.22	www.gsa.gov
0.20	www.ssa.gov
0.16	www.census.gov
0.14	www.govbenefits.gov
0.13	www.kids.gov
0.13	www.usdoj.gov
6th Singular Vector	
0.97	mathpost.asu.edu
0.18	math.la.asu.edu
0.17	www.asu.edu
0.04	www.act.org
0.03	www.eas.asu.edu

Figure 6. HITS results

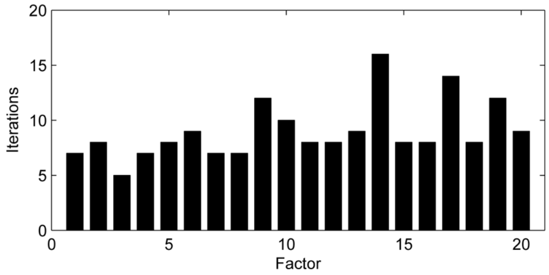


Figure 7. Number of power method iterations per PARAFAC factor

Topics		Authorities	
SCORE	TERM	SCORE	HOST
1st Principal Factor			
0.23	java	0.86	java.sun.com
0.18	sun	0.38	developers.sun.com
0.17	platform	0.16	docs.sun.com
0.16	solaris	0.14	see.sun.com
0.16	developer	0.14	www.sun.com
0.15	edition	0.09	www.samag.com
0.15	download	0.07	developer.sun.com
0.14	info	0.06	sunsolve.sun.com
0.12	software	0.05	access1.sun.com
		0.05	iforce.sun.com
2nd Principal Factor			
0.20	no-anchor-text	0.99	www.lehigh.edu
0.16	faculty	0.06	www2.lehigh.edu
0.16	search	0.03	www.lehighalumni.com
0.16	news		
0.16	libraries		
0.16	computing		
0.12	lehigh		
3rd Principal Factor			
0.15	no-anchor-text	0.97	www.ibm.com
0.15	ibm	0.18	www.alphaworks.ibm.com
0.12	services	0.07	www-128.ibm.com
0.12	websphere	0.05	www.developer.ibm.com
0.12	web	0.02	www.redbooks.ibm.com
0.11	developerworks	0.01	www.research.ibm.com
0.11	linux		
0.11	resources		
0.11	technologies		
0.10	downloads		
4th Principal Factor			
0.26	information	0.87	www.pueblo.gsa.gov
0.24	federal	0.24	www.irs.gov
0.23	citizen	0.23	www.whitehouse.gov
0.22	other	0.19	travel.state.gov
0.19	center	0.18	www.gsa.gov
0.19	languages	0.09	www.consumer.gov
0.15	u.s	0.09	www.kids.gov
0.15	publications	0.07	www.ssa.gov
0.14	consumer	0.05	www.forms.gov
0.13	free	0.04	www.govbenefits.gov
6th Principal Factor			
0.26	president	0.87	www.whitehouse.gov
0.25	no-anchor-text	0.18	www.irs.gov
0.25	bush	0.16	travel.state.gov
0.25	welcome	0.10	www.gsa.gov
0.17	white	0.08	www.ssa.gov
0.16	u.s		
0.15	house		
0.13	budget		
0.13	presidents		
0.11	office		

Figure 8. TOPHITS results

Topics		Authorities	
SCORE	TERM	SCORE	HOST
12th Principal Factor			
0.75	optimization	0.35	`www.palisade.com`
0.58	software	0.35	`www.solver.com`
0.08	decision	0.33	`plato.la.asu.edu`
0.07	neos	0.29	`www.mat.univie.ac.at`
0.06	tree	0.28	`www.ilog.com`
0.05	guide	0.26	`www.dashoptimization.com`
0.05	search	0.26	`www.grabitech.com`
0.05	engine	0.25	`www-fp.mcs.anl.gov`
0.05	control	0.22	`www.spyderopts.com`
0.05	ilog	0.17	`www.mosek.com`
13th Principal Factor			
0.46	adobe	0.99	`www.adobe.com`
0.45	reader		
0.45	acrobat		
0.30	free		
0.30	no-anchor-text		
0.29	here		
0.29	copy		
16th Principal Factor			
0.50	weather	0.81	`www.weather.gov`
0.24	office	0.41	`www.spc.noaa.gov`
0.23	center	0.30	`lwf.ncdc.noaa.gov`
0.19	no-anchor-text	0.15	`www.cpc.ncep.noaa.gov`
0.17	organization	0.14	`www.nhc.noaa.gov`
0.15	nws	0.09	`www.prh.noaa.gov`
0.15	severe	0.07	`aviationweather.gov`
0.15	fire	0.06	`www.nohrsc.nws.gov`
0.15	policy	0.06	`www.srh.noaa.gov`
0.14	climate	0.05	`news.google.com`
19th Principal Factor			
0.22	tax	0.73	`www.irs.gov`
0.17	taxes	0.43	`travel.state.gov`
0.15	child	0.22	`www.ssa.gov`
0.15	retirement	0.08	`www.govbenefits.gov`
0.14	benefits	0.06	`www.usdoj.gov`
0.14	state		
0.14	income		
0.13	service		
0.13	revenue		
0.12	credit		

Figure 9. More TOPHITS results

with a three-way PARAFAC decomposition [22] of the web graph, which provides more information than the two-way SVD used in HITS.

Further differences with HITS are apparent. TOPHITS is not restricted to focused subgraphs. If multiple topics exist in the graph, users can find the appropriate cluster by looking for the topic vectors in which their query terms have a high score. For example, if the vector $\mathbf{q}$ represents the query, then $\mathbf{q}^T\mathbf{w}^{(i)}$ is a measure of the importance of the ith factor to the query. This basic premise can be used potentially to extend TOPHITS to a query-based system. For instance, a query-dependent authority score of all web pages, $\hat{\mathbf{a}}$, could be computed as

$$\hat{\mathbf{a}} = \sum_{i=1}^{p} (\mathbf{q}^T\mathbf{w}^{(i)})\ \mathbf{v}^{(i)}.$$

There are many directions for future research. Currently, we are studying an alternative decomposition to PARAFAC called the Tucker model [38] for applications in information retrieval. We are also looking at even higher order data sets that go beyond three-way models.

Although some accelerations have been proposed (see, e.g., [40]), much work remains to be done on efficient computation of PARAFAC models for large-scale, sparse tensors. As a first step we have created a MATLAB toolbox for working with sparse tensors that can efficiently handle up to one million nonzeros. Extending these techniques to data sets the size of the Web is a topic of future study. While multiple vectors need to be stored, they can be sparsified, which will reduce both the overall storage cost as well as the computational cost for computing them. Further, convergence and stability analysis of TOPHITS should be analyzed in the same way that Ng et al. [30, 31] have analyzed the stability of PageRank and HITS.

Another future topic is the use of tensor decompositions on semantic graphs to measure similarity, analogous to how Blondel et al. use the SVD to measure the similarity between directed graphs [5]. Such techniques can be used in attribute prediction as has already been done using matrix decompositions [35] as well as probabilistic-based approaches [19, 29].

Acknowledgments

This work was funded by Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

We are indebted to Travis Bauer, David Blackledge, and Thomas Cleal for providing the web spider in STANLEY, a text analysis library being developed as part of Sandia's Cognitive Science program. We thank the referees for their advice and pointers to two additional relevant articles.

References

[1] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, and B. Yener. Modeling and multiway analysis of chatroom tensors. In *Proc. ISI 2005*, volume 3495 of *LNCS*, pp. 256–268, Jan. 2005.

[2] B. W. Bader and T. G. Kolda. MATLAB tensor classes for fast algorithm prototyping. Tech. Rep. SAND2004-5187, Sandia Natl. Labs, Oct. 2004. Submitted to *ACM Trans. Math. Soft.*

[3] 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.

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

[5] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. V. Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Rev.*, 46(4):647–666, 2004.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Comput. Networks ISDN*, 30(1–7):107–117, 1998.

[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 *Proc. 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 *Proc. ICML '00*, 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] 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/`.

[13] M. Diligenti, M. Gori, and M. Maggini. Web page scoring systems for horizontal and vertical search. In *Proc. WWW '02*, pp. 508–516. ACM, 2002.

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

[15] C. H. Q. Ding. A probabilistic model for latent semantic indexing. *J. Am. Soc. Inf. Sci. Tec.*, 56(6):597–608, 2005.

[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 *Proc. CHI '88*, pp. 281–285. ACM, 1988.

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

[19] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of link structure. *J. Mach. Learn. Res.*, 3:679–707, 2003.

[20] D. Gleich and L. Zhukov. SVD based term suggestion and ranking system. In *Proc. ICDM 2004*, pp. 391–394, 2004.

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

[22] 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.

[23] T. H. Haveliwala. Topic-sensitive PageRank. In *Proc. WWW '02*, pp. 517–526. ACM, 2002.

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

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

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

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

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

[29] 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 Explor. Newsl.*, 5(2):165–172, 2003.

[30] 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.

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

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

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

[34] 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.

[35] D. Skillicorn. Social network analysis via matrix decompositions: al Qaeda. Available from `http://www.cs.queensu.ca/home/skill/alqaeda.pdf`, Aug. 2004.

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

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

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

[39] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In *Proc. ECCV'02*, volume 2350 of *Lecture Notes in Computer Science*, pp. 447–460. Springer-Verlag, 2002.

[40] T. Zhang and G. H. Golubuhou. Rank-one approximation to high order tensors. *SIAM J. Matrix Anal. A.*, 23(2):534–550, 2001.