AUTHORITY RANKINGS FROM HITS, PAGERANK, AND SALSA: EXISTENCE, UNIQUENESS, AND EFFECT OF INITIALIZATION*†

AYMAN FARAHAT‡, THOMAS LOFARO§¶, JOEL C. MILLER‖, GREGORY RAE**, AND LESLEY A. WARD**

Abstract. Algorithms such as Kleinberg's HITS algorithm, the PageRank algorithm of Brin and Page, and the SALSA algorithm of Lempel and Moran use the link structure of a network of webpages to assign weights to each page in the network. The weights can then be used to rank the pages as authoritative sources. These algorithms share a common underpinning; they find a dominant eigenvector of a non-negative matrix that describes the link structure of the given network and use the entries of this eigenvector as the page weights. We use this commonality to give a unified treatment, proving the existence of the required eigenvector for the PageRank, HITS, and SALSA algorithms, the uniqueness of the PageRank eigenvector, and the convergence of the algorithms to these eigenvectors. However, we show that the HITS and SALSA eigenvectors need not be unique. We examine how the initialization of the algorithms affects the final weightings produced. We give examples of networks that lead the HITS and SALSA algorithms to return non-unique or non-intuitive rankings. We characterize all such networks, in terms of the connectivity of the related HITS authority graph. We propose a modification, Exponentiated Input to HITS, to the adjacency matrix input to the HITS algorithm. We prove that Exponentiated Input to HITS returns a unique ranking, so long as the network is weakly connected. Our examples also show that SALSA can give inconsistent hub and authority weights, due to non-uniqueness. We also mention a small modification to the SALSA initialization which makes the hub and authority weights consistent.

Key words. Link analysis, web search, HITS algorithm, Kleinberg's algorithm, PageRank algorithm, SALSA algorithm, hubs, authorities, networks.

AMS subject classifications. 68W40, 15A18, 68R10.

1. Introduction.

The rapid growth of the world wide web has created a need for search tools. In the past, search engines ranked pages using word frequency or similar measures. Recently, new algorithms have been created that greatly improve rankings, using the network structure of the web. Two prominent ranking algorithms are PageRank [3] (the underlying method of the *Google* search engine) and HITS [9, 14] (Hypertext Induced Topic Search). A third algorithm, SALSA [16] (Stochastic Approach for Link Structure Analysis), combines ideas from PageRank and HITS. Some search tools, such as HITS, first use conventional information-retrieval techniques to identify a subnetwork of pages relevant to a particular query, and then order the results so that the most relevant web pages are presented near the top of the list. Others, such as PageRank, rank the whole web on an absolute scale, and then select those pages relevant to the query, preserving the order.

The common thread running through these methods is linear algebra. For each method, the ranking vector is the dominant eigenvector of some matrix describing

*A brief preliminary version of this work appeared in the Proceedings of the ACM SIGIR 2001 Conference.

†Supported in part by the Harvey Mudd College Mathematics Clinic and by HNC Software, Inc.

‡PARC, 3333 Coyote Hills Rd, Palo Alto, CA 94304, Ayman.Farahat@parc.com

§Department of Mathematics and Computer Science, Gustavus Adolphus College, St. Peter, MN 56082, tlofaro@gustavus.edu

¶Supported in part by a Research, Scholarship and Creativity Grant from Gustavus Adolphus College.

‖Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, United Kingdom, jcm52@cam.ac.uk

**Department of Mathematics, Harvey Mudd College, Claremont, CA 91711, grae@hmc.edu, ward@math.hmc.edu

the network. How this matrix is defined differs in each method. However, this commonality allows us to establish theorems (see Section 3) providing conditions for each method which ensure the convergence of the algorithm, existence of solutions, and uniqueness of solutions. These results also apply to any other methods that reduce to the computation of a dominant eigenvector.

Introducing the HITS algorithm in [14, p.613], Kleinberg writes

> For the sake of simplicity, we will make the following technical assumption $\ldots |\lambda_1(M)| > |\lambda_2(M)|$ [that is, the dominant eigenvalue is simple]. $\ldots$ When the assumption does not hold, the analysis becomes less clean, but it is not affected in any substantial way.

This idea has been repeated in (among others) [2, 5, 6, 11, 17, 25]. See, for example, this passage from [6, p.5]:

> iteration will converge to the principal eigenvector for any 'non-degenerate' choice of initial vector — $\ldots$ for example, for any vector all of whose entries are positive. This says that the hub and authority weights we compute are truly an *intrinsic* feature of the collection of linked pages, not an artifact of our choice of initial weights or the tuning of arbitrary parameters. (Italics in the original.)

In this paper we take up the question of the choice of initial weights. We examine the dependence of the results of the algorithms on changes in initialization. The key point is that the dominant eigenvalue of the appropriate matrix need not be simple. When it is simple, the convergence asserted above for HITS is valid. However, when the dominant eigenvalue is repeated, the ranking vector could be any non-negative vector in the multidimensional dominant eigenspace. (This occurs for HITS in our example in Figure 4.1, Section 4.) In particular, the ranking vector may not be unique, but may depend on the initial seed vector.

Even when the dominant eigenvalue is simple, HITS can return inappropriate zero weights (Figure 4.3, Section 4).

We show that the HITS and SALSA algorithms are 'badly behaved' on certain networks, meaning that (i) they can return ranking vectors that are not unique but depend on the initial seed vector, or (ii) HITS can return ranking vectors that inappropriately assign zero weights to parts of the network. In some cases, this behavior can cause SALSA to return hub and authority weights that are inconsistent.

We characterize all networks where HITS and SALSA are badly behaved in terms of the connectivity of a certain related graph (our *HITS authority graph*, Definition 4.3). We propose a modification, *Exponentiated Input to HITS*, of the adjacency matrix input to the HITS algorithm; our new input matrix adds information about longer paths in the network. We prove that with Exponentiated Input, the HITS algorithm returns a unique eigenvector, without inappropriate zero weights, as long as the network is weakly connected.

Here is an outline of the paper. In Section 2, we review four eigenvector ranking algorithms, namely the PageRank, HITS, and SALSA algorithms, along with our Exponentiated Input modification to the adjacency matrix input to the HITS algorithm. For each algorithm, we identify the non-negative matrix that describes the structure of the network being considered.

In Section 3, we review the Perron-Frobenius theorems which deal with the properties of dominant eigenvalues and eigenvectors of non-negative matrices. We present existence and (where applicable) uniqueness results for the dominant eigenvectors the ranking algorithms find. In particular, we show that Exponentiated Input to HITS

does return a unique ranking (independent of the initialization), without inappropriate zero weights, when the network graph is weakly connected.

In Section 4, we give examples of simple networks for which the HITS and SALSA algorithms return non-unique dominant eigenvectors, and for which HITS assigns inappropriate zero weights to significant parts of the network. We say that an algorithm is *badly behaved* on a network if it has either of these two features. We give a characterization of all networks for which HITS and SALSA are badly behaved in terms of the connectivity of our HITS authority graph G' (Definition 4.3), which is an undirected graph related to the directed graph representing the network.

For our example in Section 4, Figure 4.1, SALSA returns authority and hub vectors which are inconsistent. In Section 4.3 we suggest a modification to the SALSA initialization which makes the vectors consistent.

In Section 5 we discuss related issues, including stability of the algorithms, sample networks for which the dominant and second eigenvalues are arbitrarily close for Exponentiated Input to HITS, and other possible modifications of the adjacency matrix input to HITS. Section 6 considers related work, and Section 7 has our conclusions.

We are grateful for the detailed comments provided by two anonymous reviewers.

2. Eigenvector-based Ranking Algorithms. To prepare our review of the eigenvector-based ranking algorithms (or *linear link analysis algorithms*) PageRank, HITS, Exponentiated Input to HITS, and SALSA, we give some definitions.

The network of web pages and links define a directed graph G, with the web pages defining the nodes $1, \ldots, n$, and the links defining the edges. This graph is described by an $n \times n$ *adjacency matrix* A, where $a_{ij} = 1$ if there is a link from page i to page j and $a_{ij} = 0$ otherwise. We use the following standard definitions from graph theory.

DEFINITION 2.1. *The* indegree [outdegree] *of a node i on a graph G is the number of edges coming into [out of] i.*

DEFINITION 2.2. *A directed graph G is* weakly connected *if any node can be reached from any other node by traversing edges either in their indicated direction or in the opposite direction.*

The notion of dominant eigenvectors and eigenvalues is central to the descriptions of the algorithms in terms of linear algebra.

DEFINITION 2.3. *A* dominant eigenvector *of a matrix is an eigenvector associated with an eigenvalue of largest modulus. An eigenvalue (which may be simple or repeated) of largest modulus is called a* dominant *or* largest *eigenvalue of the matrix.*

We note that some authors use the term *dominant* in a slightly different sense. Also, in [14] the term *principal eigenvector* indicates an eigenvector whose eigenvalue is simple and of *strictly* largest modulus.

With our definitions, we can describe the four linear link analysis algorithms.

2.1. PageRank. The PageRank method, developed by Brin and Page [3], is a random walk with random resets on G. This process describes a Markov chain. Let W be the transition probability matrix describing the Markov chain without reset, so that the (i, j)-entry in W gives the probability of following a link on page i to page j. Assuming all links are equally likely, W is derived from A by

$$w_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{ik}}. \tag{2.1}$$

If at least one node has zero outdegree then the Markov chain is absorbing, and a modification to W is needed. For instance, PageRank temporarily removes the links to all such nodes, then replaces them after the ranking is calculated [23].

The random reset part of PageRank is described by a probability $\varepsilon < 1$, which determines whether we restart, and a (positive) uniform transition probability matrix U having $u_{ij} = 1/n$ for all i and j. (Positive, non-uniform choices for U are also possible.) Combining these two elements we conclude that the PageRank algorithm is described by the stochastic matrix

$$P = \varepsilon U + (1 - \varepsilon)W. \tag{2.2}$$

With probability ε we randomly choose a new page. Otherwise we follow one of the directed edges from our present node.

The relationship between Markov chains such as this and linear algebra is well understood (see Gantmacher [8] for example). The stationary distribution of this Markov chain is simply the dominant eigenvector of the matrix P^T (the transpose of P). We return to this point in Section 3.

2.2. HITS. The HITS algorithm was developed by J. Kleinberg [14], and is now a part of the CLEVER Searching project of the IBM Almaden Research Center [13]. Similar ideas are used in the Teoma search engine, subsequently acquired by Ask Jeeves [24]. The premise of the algorithm is that a web page serves two purposes: to provide information on a topic, and to provide links to other pages giving information on a topic. This gives rise to two ways of categorizing a web page. First, we consider a web page to be an *authority* on a subject if it provides good information about the subject. Second, we consider the web page to be a *hub* if it provides links to good authorities on the subject. The HITS algorithm is an iterative algorithm developed to quantify each page's value as an authority and as a hub.

The HITS algorithm is not applied to the graph representing the whole web, but rather to a subgraph, typically of 1000–5000 nodes, derived from traditional text matching of the query terms in the search topic.

We again consider the directed graph G. Each node i is given an initial authority weight $a_0(i)$ and hub weight $h_0(i)$, although the initial authority weights are not used by the algorithm as presented in [14, p.612]. To begin the algorithm, these nodes are uniformly weighted. The original paper [14, p.614] states "one can compute weights ... starting from any initial [weights]." We will later discuss complications that can arise if the initial weighting is changed, or if the algorithm starts from the initial authority weights instead of the initial hub weights.

The algorithm proceeds iteratively, until values from successive iterations are within a specified error, as follows:

1. In the k^{th} iteration, node i is assigned a new authority weight $a_k(i)$ equal to the sum of $h_{k-1}(j)$, where the sum runs over each node j which points to node i. This is repeated for all nodes in G.

2. The new hub weight $h_k(i)$ is the sum of $a_k(j)$, where the sum runs over the nodes j to which node i points. This is repeated for all nodes in G. Note that the hub weights are computed from the *current* authority weights, which in turn were computed from the *previous* hub weights.

3. Still in the k^{th} iteration, after new weights are computed for all nodes, the weights are normalized so that $\sum_i a_k(i)^2 = \sum_i h_k(i)^2 = 1$.

To construct a linear algebra formulation of this method, we create a vector $\vec{a}_k$ of the authority weights at iteration k and another vector $\vec{h}_k$ of the hub weights, where

$$\vec{a}_k = [a_k(1), a_k(2), \cdots, a_k(n)]^T, \tag{2.3}$$

$$\vec{h}_k = [h_k(1), h_k(2), \cdots, h_k(n)]^T. \tag{2.4}$$

A uniform initialization gives

$$\vec{a}_0 = [1/\sqrt{n},\ 1/\sqrt{n},\ \ldots,\ 1/\sqrt{n}]^T, \tag{2.5}$$
$$\vec{h}_0 = [1/\sqrt{n},\ 1/\sqrt{n},\ \ldots,\ 1/\sqrt{n}]^T, \tag{2.6}$$

up to normalization. When A is the adjacency matrix of the directed graph G, the algorithm becomes

$$\vec{a}_k = \phi_k A^T \vec{h}_{k-1}, \qquad \vec{h}_k = \psi_k A \vec{a}_k, \tag{2.7}$$

where ϕ_k and ψ_k are the normalization constants chosen such that the sum of the squares of the authority weights, as well as that of the hub weights, is 1 in the k^{th} iteration.

Combining the formulas in (2.7), we reach

$$\vec{a}_k = \phi_k \psi_{k-1} A^T A \vec{a}_{k-1}, \qquad \text{for } k > 1; \tag{2.8}$$
$$\vec{h}_k = \psi_k \phi_k A A^T \vec{h}_{k-1}, \qquad \text{for } k > 0. \tag{2.9}$$

When this method converges, the resulting authority and hub vectors satisfy

$$\vec{a}^* = \frac{1}{\lambda^*} A^T A \vec{a}^*, \qquad \vec{h}^* = \frac{1}{\lambda^*} A A^T \vec{h}^*. \tag{2.10}$$

In other words, the authority vector $\vec{a}^*$ is an eigenvector of $A^T A$ and the hub vector $\vec{h}^*$ is an eigenvector of AA^T. It is simple to show (see Section 3.2) that these eigenvectors are, as in the case of PageRank, dominant eigenvectors corresponding to the dominant eigenvalue λ^*. To avoid confusion, it should be noted that throughout this paper, we do not concern ourselves with the eigenvalues of A, but just of $A^T A$ and AA^T.

2.3. Exponentiated Input to HITS. In [20], we showed that certain network topologies caused the standard HITS algorithm to return either non-unique or non-intuitive results (see Section 4). We developed the Exponentiated Input method to address these limitations. HITS' direct dependence on the adjacency matrix A restricts it, in a given iteration step, to considering only paths of length 1. That is, to determine the hub score of a page, at each iteration HITS looks only at the authority scores of adjacent pages. We replace A by a new matrix with information about longer paths.

To consider paths of length greater than 1, we note that given an adjacency matrix A, the number of paths of length m from node i to node j is given by the (i, j)-entry of A^m. We want to consider paths of length 1 to be more important than paths of length 2, 3, and so on. To accomplish this we use the new matrix

$$A + A^2/2! + A^3/3! + \cdots + A^m/m! + \cdots = e^A - I. \tag{2.11}$$

This series converges, since each entry in the m^{th} term of the series can be bounded by $n^m/m!$. While the original adjacency matrix A is binary, $e^A - I$ need not be.

Using this new 'exponentiated' matrix $e^A - I$, the HITS algorithm with exponentiated input can be expressed in linear algebra terms as follows. We initialize the authority and hub vectors so that $a_0(i) > 0$ and $h_0(i) > 0$ for all i, and $\sum_i a_0(i)^2 = \sum_i h_0(i)^2 = 1$. (Again, the uniform initialization $a_0(i) = h_0(i) = 1/\sqrt{n}$ is one possibility, and again, $\vec{a}_0$ is never used.) For $k \geq 1$, we *replace the adjacency matrix A input to the HITS algorithm with the matrix $e^A - I$*, so that we define

$$\vec{a}_k = \phi_k (e^A - I)^T \vec{h}_{k-1}, \qquad \vec{h}_k = \psi_k (e^A - I) \vec{a}_k, \tag{2.12}$$

for $k > 0$, with normalization constants ϕ_k, ψ_k. We will show that the sequence $\{\vec{a}_k\}$ approaches the eigenspace of the largest eigenvalue of $(e^A - I)^T(e^A - I)$, and that the sequence $\{\vec{h}_k\}$ approaches the eigenspace of the largest eigenvalue of $(e^A - I)(e^A - I)^T$.

We prove in Sections 3.2 and 3.3 that given a weakly connected input graph, the Exponentiated Input modification of HITS prevents the possibilities (i) of returning non-unique rankings that depend on the initialization and (ii) of returning inappropriate zero weights. Other matrices such as $A + A^2/2$ or $I + A$ rather than $e^A - I$ would also exclude these possibilities; again, so long as G is weakly connected.

2.4. SALSA. The SALSA (Stochastic Approach for Link Structure Analysis) algorithm, developed by Lempel and Moran [16], combines the random walk idea of PageRank with the hub/authority idea of HITS. Given a graph G, a bipartite undirected graph H is constructed by building the subset V_a of all the nodes with positive in-degree (the potential authorities), and the subset V_h of all the nodes with positive out-degree (the potential hubs). After renumbering, the elements of V_a and V_h become the nodes of H. Since some nodes may have both positive in-degree and out-degree the number m of nodes of H satisfies $m \leq 2n$, where n is the number of nodes of G. The (undirected) edges of H are defined from G as follows: if G has a link from i to j, then we put an edge between the nodes corresponding to i in V_h and j in V_a. The algorithm corresponds to a two-step random walk on the graph H.

If we pick a node in V_a and randomly follow an edge, we traverse to a node in V_h. A second step returns us to V_a. Thus a pair of two-step random walks, one starting in V_a and the other in V_h, can be used to determine the hub and authority vectors. As in the PageRank scheme, each random walk is a Markov chain with a corresponding transition probability matrix. As before, let A be the adjacency matrix of G. Let W_r be the matrix generated from A by dividing each entry of A by its row sum. Similarly define W_c to be generated by dividing each entry of A by its column sum. We arrive at $\vec{a}_k = W_c^T W_r \vec{a}_{k-1}$ and $\vec{h}_k = W_r W_c^T \vec{h}_{k-1}$. Note that the sequence $\vec{a}_k$ depends on the initialization $\vec{a}_0$, and $\vec{h}_k$ depends on the initialization $\vec{h}_0$. This contrasts with HITS where both sequences depend on $\vec{h}_0$, and leads to subtle differences in the behavior of the two algorithms. In particular the SALSA limits $\vec{a}$ and $\vec{h}$ may not satisfy $\vec{a} = W_c^T \vec{h}$.

A significant advantage of SALSA is that the weightings can be computed explicitly without the iterative process described here [16]. Although the initial paper provides an explicit calculation only in the case of uniform initial weights, this quick method of calculating the weightings can be easily generalized to accommodate non-uniform initial weights.

3. Existence and Convergence. Each of the algorithms presented in Section 2 reduces to computing the dominant eigenvector of some matrix that describes the connectivity of the given graph. In this section, we discuss the graph properties that ensure the existence and uniqueness of these eigenvectors. If the dominant eigenvalue is repeated, it will have a full eigenspace, and there does not exist a unique ranking. Therefore, perhaps unexpectedly, the ranking an algorithm returns depends on the initialization used to start the algorithm. We discuss this in greater detail in Section 4.

3.1. The Perron-Frobenius Theorems. We begin by presenting the necessary theorems concerning dominant eigenvalues and eigenvectors of non-negative matrices. These theorems describe why the ranking vectors of the above methods exist and when the rankings are unique. In addition, the different hypotheses allow us to describe shortcomings of HITS and SALSA (see Section 4). The following definitions and theorems (including proofs) can be found in Gantmacher, volume II [8].

DEFINITION 3.1. *A matrix M is* positive *if every entry of M is positive. The matrix is* non-negative *if every entry of M is non-negative.*

DEFINITION 3.2. *A permutation of the rows of M with the same permutation applied to the columns is a* matrix permutation. *We denote by σ_{ij} the fundamental (i,j)-matrix permutation (interchanging rows i and j and columns i and j).*

A matrix permutation applied to the adjacency matrix of a graph corresponds to a relabeling of the nodes. We use the notation $\circ$ for composition of matrix permutations.

DEFINITION 3.3. *A matrix M is* reducible *if there exists a matrix permutation $\sigma = \sigma_{i_1,j_1} \circ \cdots \circ \sigma_{i_m,j_m}$ such that*

$$\sigma(M) = \begin{pmatrix} M_1 & 0 \\ M_2 & M_3 \end{pmatrix}, \tag{3.1}$$

with M_1 and M_3 square matrices and 0 a zero matrix. Otherwise M is irreducible.

A graph is said to be strongly connected if every node can be reached from every other node following the direction of the edges. It can be shown that a graph whose (weighted) adjacency matrix is M is strongly connected if and only if the matrix is irreducible. This is commonly used as an equivalent definition of irreducibility.

The following three theorems concern the existence and multiplicity of a dominant real eigenvalue and the properties of its corresponding eigenvector. For our results, it is particularly important whether or not the dominant eigenvalue is simple. The first theorem, due to Perron, has the most restrictive hypothesis.

THEOREM 3.4. *A positive matrix M always has a real and positive eigenvalue r that is a simple root of the characteristic equation, and that exceeds the moduli of all the other eigenvalues. The eigenvector corresponding to the eigenvalue r can be scaled so that every entry is positive.*

Frobenius generalized Perron's result to irreducible non-negative matrices. The conclusions are naturally weaker in the sense that there may be other eigenvalues having the same modulus as r. However, the dominant real eigenvalue has multiplicity 1.

THEOREM 3.5. *An irreducible non-negative matrix M always has a positive eigenvalue r that is a simple root of the characteristic equation, such that the moduli of all other eigenvalues do not exceed r. The eigenvector corresponding to the eigenvalue r can be scaled so that every entry is positive.*

Finally, if we remove the irreducibility hypothesis, even less can be said. (Consider, for instance, the identity matrix, whose dominant eigenvalue is of course not simple.) However, the dominant real eigenvalue r does exist, is non-negative, and has a non-negative eigenvector.

THEOREM 3.6. *A non-negative matrix M always has a non-negative eigenvalue r such that the moduli of all other eigenvalues do not exceed r. An eigenvector corresponding to the eigenvalue r can be chosen so that every entry is non-negative.*

We give a lemma that will help to show that for HITS, the non-negative eigenvalue found using the above theorems is in fact larger in modulus than all other eigenvalues.

LEMMA 3.7. *If B is a real square matrix, then all eigenvalues of B^TB are real and non-negative with full eigenspace.*

Proof. The eigenvalues are real and have full eigenspaces because $M = B^TB$ is symmetric. To show that they are non-negative, we must show $x^TMx \geq 0$ for any vector x [15]. This is straightforward: $x^TMx = x^TB^TBx = |Bx|^2 \geq 0$. □

3.2. Existence and Uniqueness of Ranking Vectors. In this section, we present existence and uniqueness results for positive or non-negative dominant eigenvectors of the associated matrices for the four algorithms. In Section 3.3, we address

convergence to these eigenvectors. Similar theorems to those below can be found in the original papers, but the results on uniqueness (or lack thereof) in Theorems 3.9 and 3.10 seem to have been largely overlooked.

THEOREM 3.8 (PageRank). *Let G denote a directed graph with adjacency matrix A. If the ε defined in equation (2.2) is positive and U is a positive matrix, then the PageRank vector exists and is unique, and all entries in this vector are positive.*

Here, by the phrase *the PageRank vector exists*, we mean that the PageRank matrix P for G has an eigenvector (which we call the PageRank vector) corresponding to a positive eigenvalue that is at least as large as the modulus of every other eigenvalue of P. Similarly for the HITS, SALSA, and Exponentiated Input to HITS hub and authority vectors in Theorems 3.9, 3.10, and 3.11 below.

Proof of Theorem 3.8. The result follows immediately from Theorem 3.4. If $\varepsilon > 0$ and U is a positive matrix, then the matrix P of equation (2.2) is positive, and hence P^T has a simple dominant eigenvalue, which is positive, with a unique eigenvector having positive entries. Convergence of the PageRank algorithm to this eigenvector follows from the theory of Markov chains. See for instance [18, p.490]. □

THEOREM 3.9 (HITS). *Let G denote a directed graph with adjacency matrix A. The HITS hub and authority vectors exist and have non-negative entries.*

Proof. AA^T and A^TA are non-negative. Theorem 3.6 gives the result. □

THEOREM 3.10 (SALSA). *Let G denote a directed graph with adjacency matrix A. The SALSA hub and authority vectors exist and have non-negative entries.*

Proof. The matrices $W_c^T W_r$ and $W_r W_c^T$ defined in Section 2.4 are non-negative. Theorem 3.6 gives the result. □

We have proved that the eigenvector exists for HITS and SALSA. As we show in Section 4, it may not be unique, since the dominant eigenvalue may be repeated (although all other eigenvalues are smaller in modulus, by Lemma 3.7 for HITS, and by the theory of Markov chains for SALSA as shown in [16]). However, for Exponentiated Input to HITS, the eigenvector is unique, as we now show.

THEOREM 3.11 (Exponentiated Input to HITS). *Let G denote a directed graph with adjacency matrix A. If G is weakly connected, then the normalized Exponentiated Input to HITS authority vector exists and is unique, the authority vector is non-negative, and all nodes with positive indegree receive positive weights. Further, the analogous statements hold for the hub vector: if G is weakly connected, then the normalized Exponentiated Input to HITS hub vector exists and is unique and non-negative, and all nodes with positive outdegree receive positive weights.*

Proof. We prove this result by showing that the authority matrix $(e^A - I)^T(e^A - I)$ has a simple dominant eigenvalue whose eigenvector is positive exactly on those nodes whose indegree is positive. The proof for the hub matrix is analogous.

Without loss of generality we assume that G has n nodes, t of which have indegree equal to zero, and that those nodes with indegree zero are labeled $n - t + 1, \ldots, n$. Then the adjacency matrix is given by

$$A = \begin{pmatrix} \widetilde{A} & 0 \\ \widetilde{B} & 0 \end{pmatrix}, \tag{3.2}$$

where $\widetilde{A}$ is $(n-t) \times (n-t)$ and the matrix $\widetilde{B}$ is $t \times (n-t)$. The last t columns of

$$e^A - I = A + A^2/2! + \cdots + A^m/m! + \cdots \tag{3.3}$$

have only zero entries. Therefore

$$(e^A - I)^T(e^A - I) = \begin{pmatrix} C & 0 \\ 0 & 0 \end{pmatrix}, \tag{3.4}$$

where $C = (e^{\widetilde{A}} - I)^T(e^{\widetilde{A}} - I)$. Moreover, the characteristic polynomial of $(e^A - I)^T(e^A - I)$ is $\lambda^t p_C(\lambda)$ where $p_C(\lambda)$ is the characteristic polynomial of C. Hence if C has a simple, dominant, positive eigenvalue, then the same is true for $(e^A - I)^T(e^A - I)$. Further, if the corresponding eigenvector of C is positive, then the corresponding eigenvector of $(e^A - I)^T(e^A - I)$ is the same vector, padded with 0 entries for all the zero indegree nodes. Lemma 3.7 shows that the eigenvalues of C are real and non-negative. We need to show that C is irreducible so that Theorem 3.5 applies.

Let G' be the graph whose (weighted) adjacency matrix is given by C. This graph G' is the analog, for Exponentiated Input to HITS, of our HITS authority graph; see Section 4.2 and especially Definition 4.3. Note that G' can be considered an undirected graph because C is symmetric. C is irreducible if and only if G' is connected. We seek to show that G' is connected.

The (i, j)-entry of C is nonzero if and only if there is a node k such that the original graph G has a path from k to i and a path from k to j. From this we conclude that all vertices that can be reached from a node k following the links in G must be in the same component of G'. Further, if k has nonzero indegree, it follows that k is in that component of G' as well.

Let us assume that G' is disconnected. Then it can be separated into two components H_1 and H_2 (which may themselves be disconnected). It must be true that there is no link between H_1 and H_2 in G. Since G is weakly connected, there must be a node (with zero indegree) that links to some node l in H_1 and to another node m in H_2. However, that would lead to a connection between these nodes (l and m) in G', contradicting the assumption that G' is disconnected.

Thus G' is connected and C is irreducible. This completes the proof. □

A close inspection of the proof shows that if the more easily computable matrix $A + A^2/2$ were used instead of $e^A - I$, then the result would still hold. The main observation needed is that all nodes that can be reached from a given node i following edges in G are still in the same component of G'. This is because a link from i to j and a link from j to k in G will imply a link between j and k in G'. Consequently the proof above still applies.

Alternatively we could use $I + A$, as the graph G' contains the graph G (ignoring direction) as a subgraph. Consequently, if G is weakly connected, then G' is connected. Since G' is connected, the same arguments then show that the algorithm converges uniquely. The graph G' will be discussed in more detail later.

3.3. Convergence of the Ranking Algorithms. All of the methods described in Section 2 reduce to the computation of a dominant eigenvector. However, PageRank and SALSA compute this eigenvector using a Markov chain method, while HITS and Exponentiated Input to HITS are formulated in terms of summing the weights of linked nodes at each step.

A probability transition matrix is *regular* if some power of the matrix is positive. Moreover, a regular transition matrix has a unique dominant eigenvector that (after scaling) gives the stationary distribution of the associated Markov chain.

The transition matrix P of the PageRank method is positive, and hence regular, if the random reset parameter $\varepsilon > 0$. The ergodic theorem guarantees that the random

walk process that PageRank uses converges to the stationary vector of the transition matrix. Therefore PageRank converges to a unique ranking vector.

SALSA also uses this approach. However, because SALSA does not incorporate random reset, the graph structure affects whether the ranking vector is unique. In particular, if the bipartite graph H is not connected, then the authority and hub vectors are not unique. The original SALSA paper showed, using the ergodic theorem, that each component of the graph has a simple dominant eigenvalue which is equal to one. Each component will converge on its own, and the final result will depend on how much weight each component is given in the initialization.

The HITS and Exponentiated Input to HITS methods differ only in their input matrices, and so the proof of convergence is identical for both methods. We offer a heuristic proof, followed by a short rigorous proof.

It is sufficient to consider the convergence of the hub vector $\vec{h}_k$. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ be the eigenvalues of AA^T. Because AA^T is symmetric, each eigenvalue has a full eigenspace and the eigenvectors can be chosen to form an orthogonal basis for $\mathbf{R}^n$.

We can express $\vec{h}$ as a linear combination of the eigenvectors $\vec{v}_1, \ldots, \vec{v}_n$. So $\vec{h}_1 = \alpha_1\vec{v}_1 + \cdots + \alpha_n\vec{v}_n$, where $\alpha_i = \vec{h}_1 \cdot \vec{v}_i / \|\vec{v}_i\|^2$. Then $\vec{h}_2 = \lambda_1\alpha_1\vec{v}_1 + \cdots + \lambda_n\alpha_n\vec{v}_n$. Suppose $\lambda_1 = \lambda_2 = \cdots = \lambda_r \neq \lambda_{r+1}$. We can rewrite $\vec{h}_2$ as $\lambda_1(\alpha_1\vec{v}_1 + \cdots + \alpha_r\vec{v}_r) + \lambda_{r+1}\alpha_{r+1}\vec{v}_{r+1} + \cdots + \lambda_n\alpha_n\vec{v}_n$. Iterating, we obtain

$$\vec{h}_{k+1} = \xi_k \left(\lambda_1^k [\alpha_1\vec{v}_1 + \cdots + \alpha_r\vec{v}_r] + \sum_{i=r+1}^{n} \lambda_i^k \alpha_i \vec{v}_i \right), \tag{3.5}$$

where $\xi_k = \psi_k\phi_k\xi_{k-1}$ is the normalization constant. As k increases, the λ_1^k term dominates. Thus, $\vec{h}_k \to c\,(\alpha_1\vec{v}_1 + \cdots + \alpha_r\vec{v}_r)$. A closer analysis shows that $\psi_k\phi_k \to \lambda_1$, which gives a numerical test for convergence.

We now offer a rigorous proof of convergence.

THEOREM 3.12. *The sequence $\vec{a}_k$ provided by the HITS algorithm converges to an authority vector, and this authority vector is a non-negative eigenvector of the largest eigenvalue of A^TA. Similarly $\vec{h}_k$ converges to a hub vector, and this hub vector is a non-negative eigenvector of the largest eigenvalue of AA^T.*

The same is true for the Exponentiated Input to HITS algorithm, for the matrices $(e^A - I)^T(e^A - I)$ and $(e^A - I)(e^A - I)^T$.

The HITS algorithm computes all future hub and authority vectors, $\vec{h}_k$ and $\vec{a}_k$, $k \geq 1$, from the initial *hub* vector $\vec{h}_0$; it does not use the initial authority vector $\vec{a}_0$ (Section 2.2). A fine point is that if instead the HITS algorithm were implemented so that it made use of the initial *authority* vector and ignored the initial hub vector, then the resulting authority vectors $\vec{a}_k$ could converge to a *different* vector. Figure 4.1 gives an example where this would happen. Theorem 4.4 provides a sufficient condition for HITS to converge to the same authority vector, regardless of whether HITS makes use of $\vec{h}_0$ or $\vec{a}_0$. If G' is weakly connected, Theorem 3.11 proves that Exponentiated Input to HITS also enjoys that uniqueness property.

Proof of Theorem 3.12. The eigenvalues of AA^T are real and non-negative. It follows that while the eigenvalue of largest modulus may be repeated, all other eigenvalues have strictly smaller modulus. Since AA^T is symmetric, the eigenspaces are orthogonal. In the dominant eigenspace, we can use Gram-Schmidt orthonormalization to choose orthogonal vectors such that one of them is non-negative. Because $\vec{h}_0$ is positive, its dot product with this non-negative vector is positive, and thus $\vec{h}_0$ has

a nontrivial component in the eigenspace of the dominant eigenvalue. This ensures that the algorithm converges to an eigenvector of the largest eigenvalue [10, pp.351–3]. By the construction of the HITS algorithm, the limit cannot have negative entries. Note that although the algorithm converges, it could converge to any non-negative, normalized vector in the eigenspace of the dominant eigenvalue λ_1, depending on the initial choice $\vec{h}_0$ of the hub weights.

The same argument applies to the HITS authority vector, and to the authority and hub vectors for the Exponentiated Input to HITS algorithm. □

We note that one could apply the PageRank idea of adding a matrix εU, where $\varepsilon > 0$ and U is a positive matrix, to the matrices AA^T and A^TA for HITS, $W_rW_c^T$ and $W_c^TW_r$ for SALSA, and $(e^A - I)(e^A - I)^T$ and $(e^A - I)^T(e^A - I)$ for Exponentiated Input to HITS. (For the last-mentioned algorithm, this would only be needed when G is not weakly connected and so the dominant eigenvalue is not already known to be simple.) Since the resulting perturbed matrices are positive, Theorem 3.4 then guarantees the existence and uniqueness of a dominant eigenvector for each perturbed matrix. One could then use perturbation analysis to study the stability of the resulting eigenvector, and how close this eigenvector is to the dominant eigenspace of the original unperturbed matrix; in particular, how close the resulting authority vector is to the possible authority vectors given by the original matrix.

4. Limitations of the HITS and SALSA algorithms. In the previous section, we presented conditions for the existence and uniqueness of ranking vectors. The weakest conclusions were for the HITS and SALSA algorithms. In this section we explore in more detail the limitations of these two methods. The key questions here are whether the largest eigenvalue of A^TA for HITS, or of $W_c^TW_r$ for SALSA, is simple or repeated, and whether its eigenvector has inappropriate zero entries.

4.1. Examples. We begin by considering a few simple examples that illustrate some limitations of the HITS and SALSA algorithms.

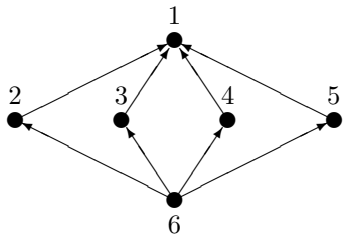

FIG. 4.1. *A network with repeated largest eigenvalue for A^TA. The HITS algorithm with standard input yields different authority vectors for different initial seed vectors, and the SALSA algorithm with standard input gives inconsistent authority and hub vectors.*

We first demonstrate the problem of repeated largest eigenvalue as the initialization changes. Examples of networks with repeated largest eigenvalue for HITS have appeared in [20], and independently in [2, Section 5] (where they were considered in the context of finding communities). The initial SALSA paper [16] mentioned the possibility of a repeated dominant eigenvalue, but did not investigate changes in the initialization.

With the uniform initialization $\vec{h}_0 = [1/\sqrt{n}, 1/\sqrt{n}, \ldots, 1/\sqrt{n}]^T$ for the hub vector, the HITS algorithm applied to the network in Figure 4.1 yields $\vec{a} = [2/\sqrt{5}, 1/2\sqrt{5}, 1/2\sqrt{5}, 1/2\sqrt{5}, 1/2\sqrt{5}, 0]^T$ and $\vec{h} = [0, 1/\sqrt{5}, 1/\sqrt{5}, 1/\sqrt{5}, 1/\sqrt{5}, 1/\sqrt{5}]^T$. This means

that vertex 6 is no better a hub than vertices 2, 3, 4 and 5. In contrast, if the chain of $\vec{h}_i$ and $\vec{a}_i$ were to start from a uniformly initialized $\vec{a}_0$ rather than $\vec{h}_0$, we would find that $\vec{a} = [1/\sqrt{5}, 1/\sqrt{5}, 1/\sqrt{5}, 1/\sqrt{5}, 1/\sqrt{5}, 0]^T$ and $\vec{h} = [0, 1/2\sqrt{5}, 1/2\sqrt{5}, 1/2\sqrt{5}, 1/2\sqrt{5}, 2/\sqrt{5}]^T$ so that vertex 1 is now no better an authority than vertices 2, 3, 4 and 5.

More generally, choosing different (positive) initial seed vectors for the HITS algorithm, the final authority vector for the network in Figure 4.1 can be

$$\vec{a} = [\alpha, \beta, \beta, \beta, \beta, 0]^T, \tag{4.1}$$

for any positive numbers α and β such that $\alpha^2 + 4\beta^2 = 1$. In other words, the possible outcomes for $\vec{a}$ can be *any* normalized, non-negative vector in the two-dimensional eigenspace of the dominant eigenvalue for A^TA except $[1, 0, 0, 0, 0, 0]^T$ and $[0, 1/2, 1/2, 1/2, 1/2, 0]^T$.

Similar behavior occurs when the SALSA algorithm is applied to the same network (in Figure 4.1). The rankings from SALSA result from the limits of two different chains which do not interact. Each of these chains will converge, but their limits will be unrelated.

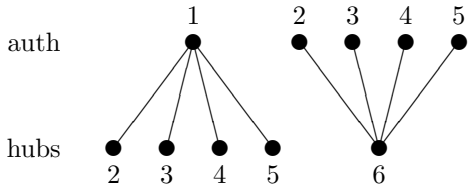

FIG. 4.2. *The SALSA graph H associated with Figure 4.1.*

The random walk performed by the SALSA algorithm is confined to whichever component of Figure 4.2 it begins in. Using the standard uniform initialization, we find that $\vec{a} = [1/5, 1/5, 1/5, 1/5, 1/5, 0]^T$ and $\vec{h} = [0, 1/5, 1/5, 1/5, 1/5, 1/5]^T$, so that all nodes with nonzero outdegree are equally good hubs and all nodes with nonzero indegree are equally good authorities.

These two weightings are inconsistent. Because the premise of HITS and SALSA is that good hubs point to good authorities, if four out of five equally good hubs all point to one node and no others, then we expect that node to be a better authority than any of the four nodes pointed to by the fifth hub. In other words, we expect that $\vec{a} = W_c^T\vec{h}$ and $\vec{h} = W_r\vec{a}$. The fact that this is not the case results from the repeated eigenvalue of $W_c^TW_r$. For arbitrary positive initial vectors, the final authority vector can be any vector of the form shown in (4.1) such that $\alpha, \beta > 0$ and $\alpha + 4\beta = 1$.

The reason for the inconsistency is that the initial hub vector assigns the first component four times as much weight as the second component. In contrast, the authority vector assigns the second component four times as much weight as the first.

For both HITS and SALSA, there are some graphs that give rise to repeated eigenvalues. The output of all these graphs is sensitive to the initial vector chosen. A second problem, inappropriate zero weights, can arise in HITS regardless of the output's dependence on or independence of the initial vector.

Consider the graph in Figure 4.3. If we remove vertex 5 from the graph, obtaining the two-level binary tree, then the characteristic polynomial of A^TA is $(\lambda - 2)^3\lambda^4$ and so A^TA has a repeated dominant eigenvalue leading to non-unique behavior. However,

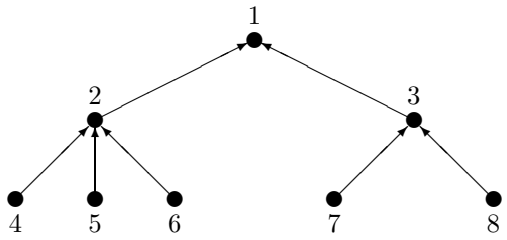

FIG. 4.3. *A tree with simple largest eigenvalue for A^TA.*

the inclusion of vertex 5 creates a significantly different situation. Now A^TA has a simple dominant eigenvalue 3 and repeated eigenvalues of 2 and 0.

In this case, as long as the initial seed vector is positive, the output of HITS is

$$\vec{a} = [0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^T, \tag{4.2}$$

$$\vec{h} = [0,\ 0,\ 0,\ 1/\sqrt{3},\ 1/\sqrt{3},\ 1/\sqrt{3},\ 0,\ 0]^T. \tag{4.3}$$

This means that node 2 gets authority weight 1, while all other nodes get authority weight 0. The majority of the graph is deemed to be completely unimportant. However, we would expect that node 1 should be a better authority than any other node, because it is accessible from every node. Meanwhile, node 3 should be only slightly less important than node 2, since there is only one more node pointing to node 2.

For the same graph (Figure 4.3), SALSA once more has repeated eigenvalues and non-unique rankings. In this case, if we use the uniform initialization, we again have inconsistent results: $\vec{a} \neq W_c^T\vec{h}$. In most cases where the SALSA algorithm has repeated eigenvalues, the uniform initialization will result in this inconsistency between the authority and hub vectors. The following definition introduces terminology to describe these behaviors.

DEFINITION 4.1. *We say that a ranking algorithm is* badly behaved *on a graph G if the final result depends on the initial seed vector, and/or if some node is given authority [hub] weight zero even though it has indegree [outdegree] greater than zero.*

4.2. Characterization of badly-behaved graphs. Theorem 4.2 in this section gives a linear algebra interpretation of what it means for the HITS and SALSA algorithms to be badly behaved on a graph. We also characterize (Theorems 4.4 and 4.5) the graphs for which the HITS and SALSA algorithms are badly behaved.

THEOREM 4.2. *Let A be the adjacency matrix of a graph G. The HITS algorithm is badly behaved if and only if at least one of the following holds:*

1. *AA^T has a repeated largest eigenvalue;*
2. *the dominant eigenvector of AA^T (and hence also A^TA) has zero entries for nodes whose outdegree (indegree) is nonzero.*

Proof. Assume Condition 1 of Theorem 4.2 holds, that is, AA^T has a repeated dominant eigenvalue.

Let the seed vector given to the HITS algorithm be $\vec{h}_0$. Each iteration preserves the direction while changing the magnitude of the component in each eigenspace. Once the non-dominant eigenvectors are effectively scaled to zero, the final result is the (normalized) component of $\vec{h}_0$ in the eigenspace of the dominant eigenvalue. Since the eigenvalue is repeated, there are many possibilities for this component. Conversely, if there are multiple possible outcomes depending on the seed vector, then the component in the eigenspace of the dominant eigenvalue is not unique, and so the eigenspace is multi-dimensional.

Now assume Condition 2 of Theorem 4.2 holds, that is, the dominant eigenvector has a 0 in entry i, corresponding to a node i with positive outdegree. As the iterates converge to the dominant eigenvector, the hub weight of node i converges to 0.

Conversely, if the hub weight of node i converges to 0, then the corresponding entry from the dominant eigenvector must be 0. □

In fact, if the dominant eigenvalue is repeated, the eigenspace is spanned by several non-negative eigenvectors that satisfy Condition 2.

Next we characterize the set of graphs on which the HITS algorithm is badly behaved, and describe some results about its behavior on these graphs.

DEFINITION 4.3. *Given a directed graph G on the vertex set $1, \ldots, n$, we define the* HITS authority graph G' *as follows. The vertex set of G' consists of those vertices of G with positive indegree. We define the (undirected) edge set of G' by letting $\{i, j\} \in E(G')$ if there exists a $k \in G$ such that (k, i) and (k, j) are directed edges of G.*

This graph G' was first described in [20]. Shortly thereafter, it was independently described in [2] in the context of stability (see Section 5.1).

We give some intuition for why the HITS authority graph G' is important for understanding the behavior of the HITS algorithm on G. Under the HITS algorithm, the authority weight of i has an impact on the hub weight of k because (k, i) is an edge of G. Similarly, the hub weight of k affects the authority weight of j. Thus, the edge $\{i, j\}$ represents a relation in two iterations of the algorithm between the authority weights of nodes i and j.

If we start the HITS algorithm with node i having authority weight 1 and all other nodes having authority weight 0, then after two iterations, all nodes joined to i in G' will have nonzero weight. After four iterations, all nodes that can be reached in at most two steps in G' will have nonzero weight. Continuing, all nodes that are in the same component as i will get nonzero weight, but the weight will not leak into any other components.

If we allow repeated edges $\{i, j\}$ in G' whenever there are multiple nodes pointing to i and j in G, and if we allow a loop at i in G' for each k pointing to i in G, then the adjacency matrix of G' is $A^T A$ (up to deletion of zero rows and columns in $A^T A$). Using appropriate matrix permutations, this matrix can be made block diagonal with each block corresponding to a different component of G'. Therefore, the eigenvectors (and eigenvalues) of $A^T A$ can be partitioned into disjoint subsets corresponding to the connected components of the HITS authority graph G'. (Note that allowing repeated edges and loops does not change the connectivity of G'.) This leads to our theorem characterizing the cases where HITS is badly behaved.

THEOREM 4.4. *The HITS algorithm is badly behaved on a graph G if and only if the HITS authority graph G' is disconnected.*

Proof. The "only if" direction of the proof follows from Theorem 3.5 along with Lemma 3.7. If HITS is badly behaved, then the matrix $\widetilde{C}$ defined below must be reducible, and therefore G' is disconnected.

Assume that there are t nodes with positive indegree. Defining $C = A^T A$, and labeling the nodes so that those with 0 indegree have the highest index, we can rewrite C as

$$C = \begin{pmatrix} \widetilde{C} & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.4}$$

where $\widetilde{C}$ is $t \times t$. We are interested in the dominant eigenvalue and eigenvector of $\widetilde{C}$.

If the dominant eigenvalue is not simple, or the eigenvector has zero entries for nodes with nonzero indegree, then $\widetilde{C}$ must be reducible. However, as noted in the proof of Theorem 3.11 above, $\widetilde{C}$ is reducible if and only if G' is disconnected.

To prove the "if" direction, it suffices to show that for each component there is a non-negative eigenvector that is positive exactly on that component. Since the matrix is symmetric, all eigenvectors of different eigenvalues are orthogonal. If there exists a unique dominant eigenvector, it is non-negative, and thus it cannot be orthogonal to all of these. In fact, it must be one of these. Using this and Theorem 4.2, it follows that the HITS algorithm is badly behaved.

(Another way to see this is, for G' with more than one connected component, to relabel the vertices of G' with the vertices of the first component first, then those of the second component, and so on. The matrix $\widetilde{C}$ becomes block-diagonal. Each eigenvector of $\widetilde{C}$ arises as an eigenvector of one of the blocks, padded with zeroes 'outside that block'. So any unique dominant eigenvector $\vec{v}$ of $\widetilde{C}$ is of this special form. But then $\vec{v}$ has zero entries everywhere 'outside that block', and in particular in those positions corresponding to vertices in other components of G'. But these vertices have positive indegree in G. Thus the HITS algorithm on G is badly behaved.) □

There is a strong connection between the HITS authority graph G' and the SALSA bipartite graph H. In particular, each edge of G' corresponds to a "vee" in H. More specifically, if there is an edge $\{i, j\}$ in G', then there exists a node k that links to both i and j. This means that on the graph H there is a hub node corresponding to k that has an edge to both i and j. It follows immediately that G' is disconnected if and only if H is disconnected. This proves the following.

THEOREM 4.5. *The SALSA algorithm is badly behaved on a graph G if and only if the HITS authority graph G' is disconnected.*

This theorem can also be proven easily from the results of Section 6.1 of [16]. It is not mentioned in [16] that multiple components can lead to inconsistent hub and authority rankings.

We do not know how likely it is in practice for the HITS authority graph G' corresponding to a particular query to be disconnected.

4.3. Modification to the SALSA initialization. The results of this section show that the inconsistency encountered with SALSA is due to the SALSA graph being disconnected. In particular, when we uniformly initialize the authority and hub vectors, the total authority weight given to a component may differ from the total hub weight given to that component. In all such cases, the final authority vector will be inconsistent with the hub vector.

This suggests the following modified initialization. To each component C_i we assign a component weight equal to $|C_i|/|H|$, the proportion of the nodes in the SALSA graph H which are in C_i. Note that $|C_i|$ counts nodes in both the authority and hub subsets, V_a and V_h, of H. When we initialize the authority weights in component C_i, we ensure that their sum equals the component weight, and similarly we initialize the hub weights. Then the final hub and authority weights will be consistent.

Although this correction will ensure that the weightings given are consistent, it does not guarantee that the weightings are meaningful in terms of the application. If some components are *a priori* more important than others because of extra information, then that should be considered when assigning the component weights.

When applied to Figure 4.1, this initialization results in $\vec{a} = [1/2, 1/8, 1/8, 1/8, 1/8, 0]^T$ and $\vec{h} = [0, 1/8, 1/8, 1/8, 1/8, 1/2]^T$. This contrasts with the results of the standard initialization, which were that all of the nonzero weights were 1/5.

5. Related Issues. In this section, we discuss the stability of HITS to changes in the network structure; two scenarios involving closely clustered top eigenvalues; and a different modification, *Usage Weighted Input to HITS*, of the adjacency matrix.

5.1. Stability. Recent work in [21, 22] studied the stability of HITS and other algorithms under small changes in the network structure. We can view the network perturbation as a continuous change in the adjacency matrix. An added link can be given weight t and a deleted link can be given weight $1-t$ as t goes from 0 to 1. Thus we move from an original network to the modified network by a continuous process.

If the first two eigenvectors switch, this corresponds to their associated eigenvalues switching order. In order for this to happen, the eigenvalues must be equal at some value t_0. Consider the network structure at that value. The only restriction on the value of the link weights in our proof of Theorem 4.4 was that they be positive. This theorem gives a necessary condition for the dominant eigenvalue to be repeated.

The authors of [21, 22] found examples of real networks on which small changes in the link structure appeared to cause the sort of switch in eigenvalues described above. This suggests that the HITS authority graph G' may have been disconnected to have caused that effect, or nearly so to have caused a similar effect.

More generally, small changes to a matrix tend to result in large changes to eigenvectors only when eigenvalues are close. We would expect the leading eigenvalues to be close together when G' is less tightly connected. This implies a close relationship between the connectivity of G' and the stability of HITS.

Similar results are obtained in [2]. The authors of [2] also describe the HITS authority graph, and focus their attention on cases where it is connected (their "authority-connected graphs").

5.2. Arbitrarily Close Eigenvalues. Although we have shown that for Exponentiated Input to HITS the largest eigenvalue of $(e^A - I)^T(e^A - I)$ is not repeated, a potential problem arises when the second largest eigenvalue is very close to the largest. The number of iterations needed for convergence increases without bound as the eigenvalues get close to one another. The following examples illustrate two scenarios where this problem appears to occur.

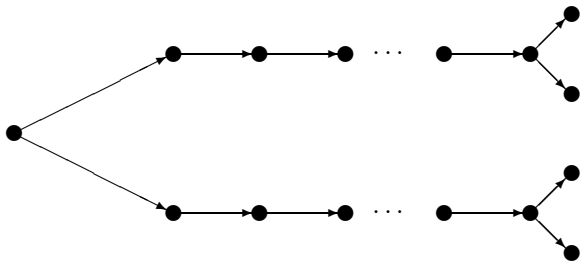

FIG. 5.1. *Broom tree with brush width $b = 2$ and large handle length l.*

Example 5.2.1. We begin by describing a family of graphs where numerical experiments suggest that the dominant and second eigenvalues can be made arbitrarily close. These graphs are of the *'broom'* form shown in Figure 5.1. The graph has a root vertex from which two identical branches begin. Each branch consists of m consecutive nodes of outdegree 1; we call $l = m - 1$ the *handle length* of the graph. The last node in this sequence has outdegree b, producing a *'brush'* at the end of each branch. We call b the *brush width* of the graph.

The ratio between the second largest eigenvalue and the dominant eigenvalue of $(e^A - I)^T(e^A - I)$ is called the *eigenvalue ratio* of the graph. Using a program written in Matlab [19] we fixed the brush width and let the handle length increase from $l = 5$ to $l = 50$. For each value of l we computed the corresponding eigenvalue ratio. When the brush width is held at $b = 1$ we obtain a family of brooms for which the eigenvalue ratio increases monotonically from 0.7796 at $l = 5$ to 0.9277 at $l = 50$. Fixing the brush width at $b = 2$ and letting l increase produces a dramatically faster convergence of the gap ratio from 0.9524 at $l = 5$ to 1.000 at $l = 50$ (all calculations rounded to 4 significant digits). Figure 5.2 shows the graph of handle length versus eigenvalue ratio for $b = 1, 2, 3$.

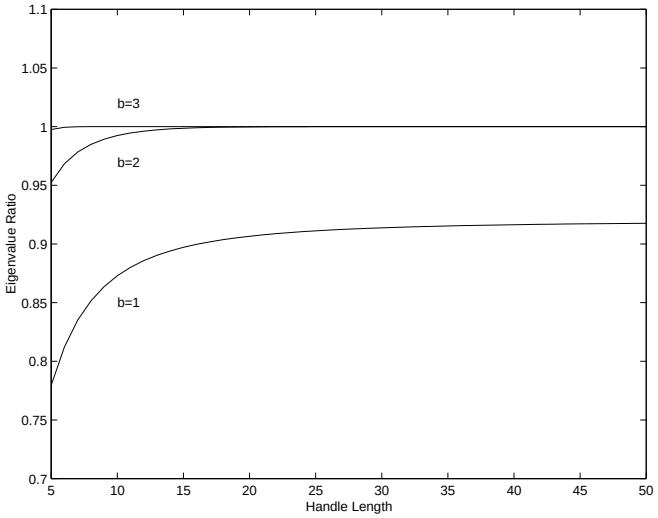

FIG. 5.2. *Ratio between second and dominant eigenvalues appears to approach one as* $l \to \infty$.

One can consider many different modifications to the adjacency matrix A input to HITS, beyond our Exponentiated Input $e^A - I$. In the example above we fixed the modification of A, and varied the graph, in order to get an eigenvalue ratio close to one. We now give an example showing that we can vary the modification of A for a fixed graph, and thus obtain an eigenvalue ratio close to one.

Example 5.2.2. Consider the two-level binary tree (delete vertex 5 from the graph shown in Section 4.1, Figure 4.3). Here the standard HITS algorithm is badly behaved because the graph has a repeated dominant eigenvalue. Consider a modification of A where paths of length 2 are counted with value ε. Let $\widetilde{A} = A + \varepsilon A^2$. With this $\widetilde{A}$, we can prove that the dominant eigenvalue will be simple for any weakly connected graph. Then the input to the HITS algorithm corresponds to finding eigenvectors of

$$\widetilde{A}^T\widetilde{A} = \begin{bmatrix} 2+4\varepsilon^2 & 2\varepsilon & 2\varepsilon & \cdots & 0 \\ 2\varepsilon & 2 & 0 & & \\ 2\varepsilon & 0 & 2 & & \\ \vdots & & & \ddots & \\ 0 & & & & 0 \end{bmatrix}, \tag{5.1}$$

where all omitted entries are 0. As $\varepsilon \to 0$, Gershgorin's disk theorem [12, p.344] shows that the three largest eigenvalues of $\widetilde{A}^T\widetilde{A}$ become arbitrarily close.

To sum up, the examples in this subsection indicate that the top few eigenvalues, either of $(e^A - I)^T(e^A - I)$ or of $\widetilde{A}^T\widetilde{A}$, may be very close to each other, even if they are not equal in modulus.

5.3. Usage Weighted Input to HITS. A different type of modification to the adjacency matrix is possible when webserver logs are available for a given network. In [20], we proposed *Usage Weighted Input to HITS*, in which we initialize the matrix to zero, and then increment the (i, j)-entry every time a user travels from node i to node j. (The resulting input matrix need not mirror the network, since users may ignore some hyperlinks and may navigate directly between pages that are not hyperlinked.) In each iteration the most frequently followed links play the largest role in determining new authority weights. The effect is similar to that of lifting by gradient ascent [7], but does not require direct querying of users. Similar approaches have recently been advocated by [26] and [27].

6. Related Work. Kleinberg's HITS algorithm is presented in [14], and discussed in more detail in [9]. Several authors have extended the HITS algorithm using textual analysis. In their paper [1] on topic distillation in hyperlinked environments, Bharat and Henzinger combine the connectivity analysis of the HITS algorithm with content analysis. They identify three limitations of the HITS algorithm: (i) mutually reinforcing relationships between hosts; (ii) automatically generated links that do not reflect a human assessment of the value of pages; and (iii) non-relevant documents leading to topic drift. They propose modifications to the HITS algorithm, using content analysis, that address these limitations. For instance, they address the common problem of topic drift by introducing the *relevance weight* of a node, which measures the similarity of the node to a query document. Relevance weights are then used to prune non-relevant nodes, and to regulate the influence of a node according to its relevance, specifically by multiplying the vectors of hub and authority weights componentwise by the vector of relevance weights. Both the limitations and the solutions are different from those considered in the present paper; in particular we do not use content analysis.

The ARC or *automatic resource compiler* system [5], for compiling a list of authoritative web documents on a given topic, also augments the HITS algorithm with content analysis. Specifically, the ARC system multiplies the weight assigned to an *href* link to a given node by a positive factor that increases with the amount of topic-related text occurring around the *href* link. These ideas are developed further in the paper [4] on spectral filtering. Here nodes may be webpages or even smaller units, allowing a finer analysis of the link topology, and in particular inhibiting topic drift. In [4], the HITS algorithm is also modified to address several other idiosyncratic features of the web. The adjacency matrix is replaced by an *affinity matrix* in which the entry a_{ij} represents the strength of the affinity between nodes i and j. The value of a_{ij} is a sum of three components, one of which quantifies the occurrence of query terms in text near the hyperlink to node j.

The paper [7] on customized authority lists modifies the HITS algorithm by *lifting* of authority weights. The idea is to incorporate user feedback to adjust the authority weights to reflect an individual user's evaluation of authoritativeness.

Borodin et al [2] discuss other related theoretical aspects of authority and hub algorithms including both HITS and SALSA. They provide two mechanisms for determining when different algorithms provide similar rankings and then use this framework to compare several ranking algorithms. The stability of ranking algorithms to graph

modifications is also discussed in detail. These ideas are similar in spirit to those in [21, 22].

Our work differs from [4, 5] in that we do not use textual analysis, and from [7] in that we do not require direct querying of users. Our Exponentiated Input work differs from the other work discussed here in that the adjacency matrix is modified to take into account paths of length greater than one. Our work differs from [2, 21, 22] in that we are not concerned with the stability of the result to changes in the network structure. We are interested in networks for which the results of the ranking algorithm are sensitive to changes in the initialization.

The use of links in ranking documents and finding authoritative sources is similar to the techniques of citation analysis in bibliometrics. See also [4, 7] for fuller surveys of the literature.

7. Conclusions and Further Work. We have given a unified treatment of the existence and uniqueness of ranking vectors generated by four link analysis algorithms, and of convergence of the algorithms. We have shown that the HITS and SALSA algorithms behave unexpectedly on some graphs. In particular, their output can depend on the initial seed vector; the output of HITS can depend on whether the implementation uses the initial authority vector or the initial hub vector; HITS can inappropriately assign zero weights to parts of a graph; and the authority and hub vectors produced by SALSA can be inconsistent. The PageRank algorithm does not display this behavior. Nor does Exponentiated Input to HITS, if the graph is weakly connected.

It is important to understand when this unexpected behavior occurs. We have characterized the graphs for which HITS and SALSA are badly behaved; for this we presented a general method that extends ideas in [2, 16]. This method can be easily applied to other algorithms and can help motivate modifications that eliminate the bad behavior. In this light, we have introduced a new modification of the HITS algorithm, namely Exponentiated Input to HITS, which eliminates bad behavior so long as the network is weakly connected. We have also proposed a change to the initialization of SALSA which eliminates inconsistency in hub and authority weights.

The idea of Exponentiated Input can be extended to SALSA by changing the input matrix A with similar results. The following natural extension leads to a similar algorithm. Consider a web surfer surfing for n time steps. At each step, she either remains where she is, with probability $(n-1)/n$, or she follows a link at random, with probability $1/n$. After n time steps the expected number of links followed is 1. The surfer then follows links backward in a similar manner for n time steps. Taking the limit as $n \to \infty$, the algorithm reduces to finding the dominant eigenvector of $e^{W_r} e^{W_c^T}$.

The choice of scaling factors (for paths of each length) used in constructing the Exponentiated Input matrix would be interesting to study. This is especially true because calculating powers of a large matrix is computationally expensive. At present we have no principled way to decide which scaling factors are most appropriate for a given application.

While the results of this paper are concerned with existence and uniqueness, and with the dependence of rankings on initialization, we have discussed how the methods applied here may relate to other important ideas of Link Analysis. In particular, the HITS authority graph G' can be used to explore both stability and community issues addressed in other works.

REFERENCES

[1] K. Bharat and M. R. Henzinger, *Improved algorithms for topic distillation in a hyperlinked environment*, in Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998.

[2] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, *Finding authorities and hubs from link structures on the World Wide Web*, in Proceedings of the Tenth International Conference on the World Wide Web, December 2001, pp. 415–429.

[3] S. Brin and L. Page, *The anatomy of a large-scale hypertextual (web) search engine*, in Proceedings of the Seventh International Conference on the World Wide Web, 1998.

[4] S. Chakrabarti, B. Dom, D. Gibson, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, *Spectral filtering for resource discovery*, in ACM SIGIR workshop on Hypertext Information Retrieval on the Web, 1998.

[5] S. Chakrabarti, B. Dom, S. Rajagopalan, D. Gibson, and J. Kleinberg, *Automatic resource compilation by analyzing hyperlink structure and associated text*, in Proceedings of the Seventh International Conference on the World Wide Web, 1998.

[6] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg, *Mining the link structure of the World Wide Web*, IEEE Computer, 32 (1999), pp. 60–67.

[7] H. Chang, D. Cohn, and A. McCallum, *Creating customized authority lists*, in Proceedings of the Seventeenth International Conference of Machine Learning, 2000.

[8] F. R. Gantmacher, *Matrix Theory*, vol. 2, Chelsea, 1974.

[9] D. Gibson, J. Kleinberg, and P. Raghavan, *Inferring web communities from link topology*, in Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, 1998.

[10] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, second ed., 1989.

[11] M. Henzinger, *Link analysis in web information retrieval*, IEEE Data Engineering Bulletin, 23 (2000), pp. 3–8.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[13] IBM, *The Clever Project*. http://www.almaden.ibm.com/cs/k53/clever.html.

[14] J. Kleinberg, *Authoritative sources in a hyperlinked environment*, Journal of the ACM, 46 (1999), pp. 604–632.

[15] P. D. Lax, *Linear Algebra*, John Wiley and Sons, 1997.

[16] R. Lempel and S. Moran, *The stochastic approach for link-structure analysis (SALSA) and the TKC effect*, in Proceedings of the Ninth International Conference on the World Wide Web, May 2000.

[17] L. Li, Y. Shang, and W. Zhang, *Improvement of HITS-based algorithms on web documents*, in Proceedings of the Eleventh International Conference on the World Wide Web, May 2002.

[18] C. R. MacCluer, *The many proofs and applications of Perron's Theorem*, SIAM Review, 42 (2000), pp. 487–498.

[19] The Mathworks Inc., *Matlab version 5.3*, 1999.

[20] J. C. Miller, G. Rae, F. Schaefer, L. A. Ward, A. Farahat, and T. LoFaro, *Modifications of Kleinberg's HITS algorithm using matrix exponentiation and web log records*, in Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 2001, pp. 444–445.

[21] A. Y. Ng, A. X. Zheng, and M. I. Jordan, *Link analysis, eigenvectors and stability*, in Proceedings of the Eighteenth International Conference on Machine Learning, 2001.

[22] ———, *Stable algorithms for link analysis*, in Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 2001.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank citation ranking: Bringing order to the Web*, http://newdbpubs.stanford.edu:8090/pub/1999-66, 1998.

[24] S. Robinson, *The ongoing search for efficient Web search algorithms*, SIAM News, 37 No. 9 (2004), pp. 4,11.

[25] N. Sundaresan, J. Yi, and A. Huang, *Using metadata to enhance a web information gathering system*, in WebDB (Informal Proceedings), 2000, pp. 11–16.

[26] M. Wang, *A significant improvement to Clever algorithm in hyperlinked environment*, in Proceedings of the Eleventh International Conference on the World Wide Web, 2002.

[27] G.-R. Xue, H.-J. Zeng, Z. Chen, W.-Y. Ma, H.-J. Zhang, and C.-J. Lu, *User access pattern enhanced small web search*, in Proceedings of the Twelfth International Conference on the World Wide Web, 2003.