# A Theoretical Study of a Generalized Version of Kleinberg's HITS Algorithm

Maristella Agosti (maristella.agosti@unipd.it) and Luca Pretto (luca.pretto@unipd.it)*
*Department of Information Engineering, University of Padua, Via Gradenigo 6/a, 35131 Padua, Italy*

**Abstract.** Kleinberg's HITS algorithm (Kleinberg 1999), which was originally developed in a Web context, tries to infer the authoritativeness of a Web page in relation to a specific query using the structure of a subgraph of the Web graph, which is obtained considering this specific query. Recent applications of this algorithm in contexts far removed from that of Web searching (Bacchin et al. 2002, Ng et al. 2001) inspired us to study the algorithm in the abstract, independently of its particular applications, trying to mathematically illuminate its behaviour. In the present paper we detail this theoretical analysis. The original work starts from the definition of a revised and more general version of the algorithm, which includes the classic one as a particular case. We perform an analysis of the structure of two particular matrices, essential to studying the behaviour of the algorithm, and we prove the convergence of the algorithm in the most general case, finding the analytic expression of the vectors to which it converges. Then we study the symmetry of the algorithm and prove the equivalence between the existence of symmetry and the independence from the order of execution of some basic operations on initial vectors. Finally, we expound some interesting consequences of our theoretical results.

**Keywords:** World Wide Web retrieval, ranking, link analysis, Kleinberg's HITS algorithm

## 1. Introduction and related works

Kleinberg's HITS (Hyperlink-Induced Topic Search) algorithm (Kleinberg 1999) is a link analysis algorithm which was originally developed in a Web context; HITS is thought to be useful for inferring Web pages that would be considered authorities for a particular query made to a search engine. This deduction uses only the information from the structure of a directed graph built on the basis of the particular query, and which is a subgraph of the directed graph representing the Web. The algorithm selects some nodes of this directed graph through the exploitation of a mutual reinforcing relationship. Recent applications of the HITS algorithm in fields far removed from that of Web searching, such as that of word stemming (Bacchin et al. 2002), and the discov-

* Supported in part by a grant from the Italian National Research Council (CNR) research project "Technologies and Services for Enhanced Content Delivery".

ery of a connection between the technique of HITS and the Latent Semantic Indexing technique (Ng et al. 2001), have led us towards a theoretical study of this algorithm. In fact we think that an in-depth mathematical analysis is necessary to better understand the behaviour of this algorithm in various contexts. The purpose of this analysis is that of deriving properties to decide if the algorithm could reasonably be applied in other fields.

In this paper we perform this mathematical analysis, considering a general formulation of the algorithm from which the original algorithm can be deduced as a particular case; we focus on the convergence of the algorithm and on the study of its symmetry. We have tried to carry out this analysis examining the algorithm independently of its Web application, so that we address some aspects of the algorithm that are crucial to having a wider knowledge of its general behaviour.

The analysis of retrieval algorithms in general is not yet typical in Information Retrieval. A technical survey on Information Retrieval on the Web can be found in Agosti and Melucci (2001); (Henzinger 2001) is a brief survey focused on hyperlink analysis for the Web. Kleinberg's original work is described in Kleinberg (1999). The cited article introduces the HITS algorithm in a Web context; HITS is used to rank the Web pages retrieved by a text-based search engine in response to a specific query. Since Kleinberg's algorithm works on a subgraph built on the basis of the query, it is said to be an algorithm which gives a query-dependent ranking, as opposed to the algorithms giving a query-independent ranking, such as PageRank (Page et al. 1998), used by the Google search engine (Brin and Page 1998). SALSA, another famous link analysis algorithm for performing query-dependent ranking, is described in Lempel and Moran (2001). These and other link analysis algorithms are studied in depth in Borodin et al. (2001), where some useful criteria to compare link analysis algorithms are also defined. The stability of link analysis algorithms, i.e. how much these algorithms are perturbed by the changing of a portion of the set of Web pages on which they are working, is studied in Ng et al. (2001). We have been led towards the mathematical analysis of the HITS algorithm described here by the results presented in Bacchin et al. (2002). This latter work uses the core of Kleinberg's algorithm in a context far removed from that of Web searching, that is the context of word stemming. In Ng et al. (2001) there is another interesting application of the algorithm in a different context, the one of traditional Information Retrieval for "flat" documents; in the cited paper there is an attempt to see the connections between HITS and the well-known Latent Semantic Indexing technique. A brief theoretical analysis of the HITS algorithm to study its limitations in a Web context is performed in Miller et al. (2001);

this analysis is used to suggest new algorithms that try to overcome these limitations.

This paper is structured as follows. In Section 2 we give some mathematical results that should be remembered in order to understand the progression of the paper. Section 3 sets out Kleinberg's algorithm and the generalization of the algorithm we are considering here. Sections 4 and 5 study the convergence of the algorithm. Section 6 studies the symmetry of the algorithm and proves the equivalence between symmetry and the order of the basic update operations. Since this work has a theoretical aim, we do not give a list of all its possible applications. In Section 7, however, we expound some interesting consequences of our theoretical results.

## 2. Some preliminary mathematical results

In this section we present the main mathematical results that will be used throughout this paper. Our aim is to remind the reader of these results and to fix terminology and notation. The proofs of the theorems presented here can be found in the cited literature. In the rest of this paper we will explicitly give proofs of our original results.

We consider a *graph* as consisting of a vertex set $V$ and an edge set $E$, where an edge is an unordered pair of vertices. A *directed graph*, or *digraph*, consists of a node set $N$ and an arc set $A$, where an arc is an ordered pair of nodes. In the following, we will consider the vertex set $V$ of a graph as given by $V = \{1, 2, \ldots, n\}$. A generic graph $G = (V, E)$ with $|V| = n$ can be represented by an $n \times n$ adjacency matrix $\mathbf{A}$, whose generic entry $(i, j)$, $(\mathbf{A})_{ij} \triangleq a_{ij}$, is given by

$$a_{ij} = \begin{cases} 1 & \text{if an edge between vertices } i \text{ and } j \text{ exists,} \\ 0 & \text{otherwise.} \end{cases}$$

A weighted graph $G_w = (V_w, E_w)$, where $|V_w| = n$, is a graph whose edges have an associated positive weight, so that for example the edge between vertices $i$ and $j$ has a weight $e_{ij}$. The $n \times n$ matrix $\mathbf{W}$ whose generic entry $(i, j)$ is

$$w_{ij} = \begin{cases} e_{ij} & \text{if an edge between vertices } i \text{ and } j \text{ with weight } e_{ij} \text{ exists,} \\ 0 & \text{otherwise,} \end{cases}$$

is referred to as the matrix whose underlying graph is $G_w$. On the other hand, we say that $G_w$ is the graph described by the matrix $\mathbf{W}$. Similar definitions can be given for digraphs.

Let us denote by $\mathbb{R}$ the set of real numbers and by $\mathbb{C}$ the set of complex numbers. Let $L$ be a vector space[1] over $\mathbb{C}$. Let $\mathbf{A}$ be an $n \times n$ complex matrix: $\mathbf{A} \in \mathbb{C}^{n \times n}$. If for a scalar $\lambda \in \mathbb{C}$ and a vector $\mathbf{v} \in L$, $\mathbf{v} \neq \mathbf{0}$, it is $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ we say that $\lambda$ is an *eigenvalue* of $\mathbf{A}$ belonging to the eigenvector $\mathbf{v}$, and that $\mathbf{v}$ is an *eigenvector* of $\mathbf{A}$ with the eigenvalue $\lambda$. A matrix $\mathbf{A}$ is said to be *symmetric* if $\mathbf{A} = \mathbf{A}^T$, where $\mathbf{A}^T$ denotes the transpose of matrix $\mathbf{A}$. For real symmetric matrices the following fundamental facts hold (Godsil and Royle 2001):

THEOREM 2.1. *The eigenvalues of a real symmetric matrix* $\mathbf{A}$ *are real numbers.*

THEOREM 2.2. *Let* $\mathbf{A}$ *be a real symmetric* $n \times n$ *matrix. Then* $\mathbb{R}^n$ *has an orthonormal basis consisting of eigenvectors of* $\mathbf{A}$*.*

COROLLARY 2.3. *If* $\mathbf{A}$ *is an* $n \times n$ *real symmetric matrix, then there are real matrices* $\mathbf{L}$ *and* $\mathbf{D}$ *such that* $\mathbf{L}^T\mathbf{L} = \mathbf{L}\mathbf{L}^T = \mathbf{I}$ *and* $\mathbf{L}\mathbf{A}\mathbf{L}^T = \mathbf{D}$*, where* $\mathbf{I}$ *is the identity matrix and* $\mathbf{D}$ *is the diagonal matrix of eigenvalues of* $\mathbf{A}$*.*

An $n \times n$ real symmetric matrix $\mathbf{A}$ is *positive semidefinite* if $\mathbf{u}^T\mathbf{A}\mathbf{u} \geq 0$ for all vectors $\mathbf{u} \in \mathbb{R}^n$; it is *positive definite* if it is positive semidefinite and $\mathbf{u}^T\mathbf{A}\mathbf{u} = 0$ if and only if $\mathbf{u} = \mathbf{0}$. These terms are used only for symmetric matrices.

THEOREM 2.4. *A real symmetric matrix is positive semidefinite if and only if its eigenvalues are nonnegative. A real symmetric matrix is positive definite if and only if its eigenvalues are positive.*

THEOREM 2.5. *A real symmetric matrix* $\mathbf{A}$ *is positive semidefinite if and only if there is a real matrix* $\mathbf{B}$ *such that* $\mathbf{A} = \mathbf{B}^T\mathbf{B}$*.*

Let $\mathbf{A}$ be a square matrix. The *spectral radius* $\rho(\mathbf{A})$ is the maximum of the moduli of the eigenvalues of $\mathbf{A}$. If $\lambda$ is an eigenvalue of $\mathbf{A}$ with $|\lambda| = \rho(\mathbf{A})$, we say that $\lambda$ is a *dominant eigenvalue*; in this case, if $\mathbf{v}$ is an eigenvector of $\mathbf{A}$ with $\lambda$ as eigenvalue, we say that $\mathbf{v}$ is a *dominant eigenvector*. An $n \times m$ real matrix $\mathbf{A}$ is said to be *positive* if all its entries are positive numbers: $(\mathbf{A})_{ij} \triangleq a_{ij} > 0$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. An $n \times m$ real matrix $\mathbf{A}$ is said to be *nonnegative* if all its entries are nonnegative numbers: $(\mathbf{A})_{ij} \triangleq a_{ij} \geq 0$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

[1] Typically, in Linear Algebra literature, the symbol used to denote a vector space is $V$. Here we have preferred to use $L$, as we have used $V$ to denote the vertex set of a graph. $L$ refers to the less commonly used term "linear space" in place of "vector space".

A nonnegative square matrix $\mathbf{A}$ is said to be *irreducible* if its underlying digraph is strongly connected. Note that if $\mathbf{A}$ is also symmetric, $\mathbf{A}$ is irreducible if and only if its underlying graph is connected. In the rest of this paper we will use these results (Godsil and Royle 2001, Salce 1993):

THEOREM 2.6 (Frobenius). *Let* $\mathbf{A}$ *be a real nonnegative and irreducible square matrix. Then:*

1. $\rho(\mathbf{A}) > 0$*;*
2. $\rho(\mathbf{A})$ *is an eigenvalue of* $\mathbf{A}$ *belonging to a real positive eigenvector;*
3. $\rho(\mathbf{A})$ *has* 1 *as algebraic multiplicity.*

A real nonnegative and irreducible square matrix $\mathbf{A}$ is said to be *primitive* if its spectral radius $\rho(\mathbf{A})$ is a strictly dominant eigenvalue, i.e. no other eigenvalue of $\mathbf{A}$ has the modulus that equals $\rho(\mathbf{A})$.

Using the results presented in Theorems 2.4 and 2.6 we can get what is stated in the following corollary:

COROLLARY 2.7. *Let* $\mathbf{A}$ *be a positive semidefinite real symmetric matrix. If* $\mathbf{A}$ *is nonnegative and irreducible, then* $\mathbf{A}$ *is primitive.*

If $\lambda$ is a strictly dominant eigenvalue belonging to the eigenvector $\mathbf{v}$, then $\mathbf{v}$ is said to be a strictly dominant eigenvector.

## 3. Kleinberg's algorithm revised

In this section we describe Kleinberg's HITS algorithm (Kleinberg 1999) and we extend the algorithm to make it more general. Even though we are describing the algorithm using its original Web application, we want to stress that the core of the algorithm can be applied in contexts far removed from that of Web searching.

It is useful to our analysis to underline that the HITS algorithm is composed of two different parts. The first, related to the *construction of the directed graph*, depends on the particular application of the algorithm. The second, *bringing hubs and authorities to the surface*, is the core of the algorithm and works on the digraph built in the first part, trying to deduce which nodes of the digraph can be considered particularly "important" (authorities) and which nodes give a particular importance to other nodes (hubs); this work is done considering only the structure of the directed graph built in the first part.

## Construction of the directed graph

This part is strongly dependent on the particular application of the algorithm. In fact, the directed graph may represent different entities and relationships between entities, depending on the application of the algorithm. For instance, in a Web search application, the nodes may represent Web pages and the arcs may represent links between Web pages; in a stemming application, the nodes may represent prefixes and suffixes of words and the arcs may represent the existence of words with some prefixes and suffixes and so on. Here we give a brief description of this part of the algorithm with regard to its Web application, as proposed in the original paper by Kleinberg. In its classical Web application, the adoption of Kleinberg's algorithm has the target of improving the performance of Web search engines. To be more precise, the main goal of its use is that of arranging for the search engine to retrieve a list of important, or authoritative, Web pages as an answer to a query $\sigma$.

The beginning of the algorithm is dedicated to the construction of a root set $R_\sigma$ of Web pages retrieved by a text-based search engine as an answer to the query $\sigma$. This root set is made of the first $m$ pages in the list of pages retrieved by the search engine; typically $m \cong 200$. Then a base set $S_\sigma$ of Web pages is built, considering the pages in the root set, plus all the Web pages that are pointed to by at least one page of the root set and the Web pages pointing to at least one of the pages of the root set. Since the number of Web pages pointing to a certain Web page can be very large, it is necessary to limit the number that has to be taken into consideration by the algorithm. Let $k$ be the maximum number actually considered of Web pages pointing to each page in $R_\sigma$. Typically $k \cong 50$. If we consider the base set $S_\sigma$ and all the existing links between pages in $S_\sigma$, we get a set of interlinked Web pages which can be represented by a directed graph $G_\sigma$. In this directed graph each node represents a Web page and each arc from node $i$ to node $j$ represents a link from page $i$ to page $j$. The second part of the algorithm works on this directed graph $G_\sigma = (N_\sigma, A_\sigma)$ . With regard to the query $\sigma$, the algorithm tries to infer the importance of each Web page $i$ from the structure of the directed graph $G_\sigma$. Since a link from page $i$ to page $j$ is considered as conferring authority to page $j$, before applying the core of the algorithm to the directed graph, a slight change is made on it: all the arcs representing a link between two pages with the same domain name are removed, as they logically have the purpose of helping navigation inside a Web site, more than that of conferring authority to the pointed Web page.

## Bringing hubs and authorities to the surface

This part is the core of the algorithm, working on the digraph $G_\sigma = (N_\sigma, A_\sigma)$. In the following we suppose $|N_\sigma| = n$. The basic idea here is that if a Web page in the base set $S_\sigma$ has many in-links, it can be either an authority on the considered subject or simply a popular page; but if these in-links come from pages that are for the most part linking to the same other pages, then it is reasonable to take this page to be an authority (see Fig. 1). So the goal of the algorithm is that of bringing authorities to the surface and separating them from mere popular pages. This can be done by exploiting a kind of mutual reinforcing approach. To make this mutual reinforcement appear, a hub weight $h_i$ and an authority weight $a_i$ are assigned to each node $i$ of the digraph $G_\sigma$. These two weights are both initially set to 1, and then authority weights and hub weights are updated according to the formulas:

$$a_i^{(k)} = \sum_{j:j\to i} h_j^{(k-1)} \qquad h_i^{(k)} = \sum_{j:i\to j} a_j^{(k)} \tag{1}$$

where $k$ is the step of the iterative procedure we are considering and the symbol "$\to$" means that the page on its left points to the page on its right. The first sum is extended to all the Web pages $j$ that point to page $i$, and the second to all the pages $j$ that are pointed to by page $i$. If we consider the adjacency matrix $\mathbf{A} = [a_{ij}]$ of the directed graph $G_\sigma$, and the vectors of authorities and hubs:

$$\mathbf{a}^{(k)} = \begin{bmatrix} a_1^{(k)} \\ a_2^{(k)} \\ \vdots \\ a_n^{(k)} \end{bmatrix} \qquad \mathbf{h}^{(k)} = \begin{bmatrix} h_1^{(k)} \\ h_2^{(k)} \\ \vdots \\ h_n^{(k)} \end{bmatrix}$$

we can express the formulas (1) through:

$$\begin{aligned} \mathbf{a}^{(k)} &= \mathbf{A}^T\mathbf{h}^{(k-1)} \\ \mathbf{h}^{(k)} &= \mathbf{A}\mathbf{a}^{(k)}. \end{aligned}$$

These formulas, giving the basic operations of the algorithm, are to be considered for $k = 1, 2, \ldots$. Moreover, since the value of some authorities and hubs could become too large, a normalization is imposed at each step.

Let us formalize this part of the algorithm. Let

$$\mathbf{u} = \left.\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}\right\} n \text{ entries.}$$

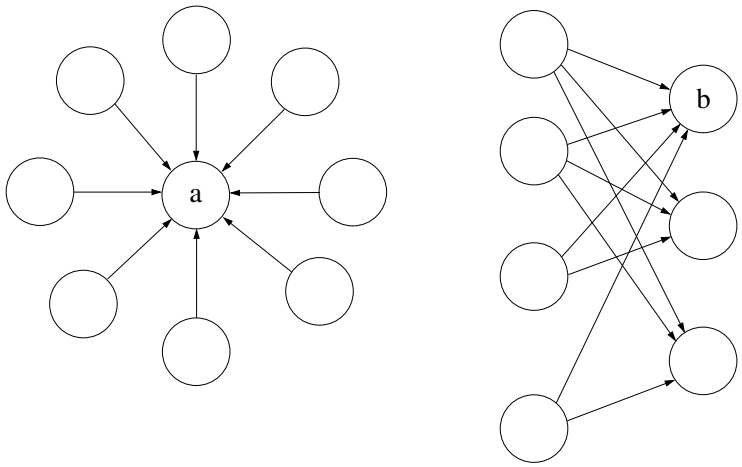


*Figure 1.* The difference between a page that is merely popular (page a) and an authority (page b) according to the idea underlying Kleinberg's algorithm.

Let $M > 1$ be an integer number that gives the number of iterations when applying the algorithm. In a theoretical setting the number of iterations to be considered should be infinite, but in the Web application of the algorithm it has been noted in experiments that a number of iterations equal to 20 is sufficient. Using a matrix notation, we can formalize this part of the algorithm as follows:

$\mathbf{a}^{(0)} := \mathbf{u}$;
$\mathbf{h}^{(0)} := \mathbf{u}$;
for $k := 1$ to $M$ do
begin
$\mathbf{a}^{(k)} := \mathbf{A}^T\mathbf{h}^{(k-1)}$;
$\mathbf{h}^{(k)} := \mathbf{A}\mathbf{a}^{(k)}$;
normalize $\mathbf{a}^{(k)}$ so that $\|\mathbf{a}^{(k)}\| = 1$;
normalize $\mathbf{h}^{(k)}$ so that $\|\mathbf{h}^{(k)}\| = 1$
end
$\mathbf{a} := \mathbf{a}^{(M)}$;
$\mathbf{h} := \mathbf{h}^{(M)}$.

At the end of the iterations, $\mathbf{a}$ gives the authority vector and $\mathbf{h}$ the hub vector. Since the choice of the norm to normalize vectors does not influence the behaviour of this algorithm, we have not specified it in the written version of the algorithm; it can be noted that Kleinberg

uses $\|\cdot\|_2$. After $k$ steps, $k > 0$, we have, normalization apart:

$$\begin{aligned} \mathbf{a}^{(k)} &= (\mathbf{A}^T\mathbf{A})^{k-1}\mathbf{A}^T\mathbf{u} \\ \mathbf{h}^{(k)} &= (\mathbf{A}\mathbf{A}^T)^k\mathbf{u}. \end{aligned}$$

In this paper we are interested in introducing an extension to Kleinberg's algorithm. This extended algorithm can be obtained from the one previously described, by considering the possibility of giving a positive weight to each arc of the digraph. This weight may have different meanings, depending on the applications of the algorithm: in a Web context the weight can be applied with the meaning of the probability that a particular arc indeed gives a measure of authoritativeness. Therefore we can consider an $n \times n$ matrix $\mathbf{W} = [w_{ij}]$ whose generic entry $(i, j)$ is:

$$w_{ij} = \begin{cases} e_{ij} & \text{if the arc from node } i \text{ to } j \text{ has the positive weight } e_{ij}, \\ 0 & \text{if no arc exists from node } i \text{ to node } j. \end{cases}$$

Obviously, if $e_{ij} = 1$ for every arc from node $i$ to node $j$, so that

$$w_{ij} = \begin{cases} 1 & \text{if an arc from node } i \text{ to node } j \text{ exists} \\ 0 & \text{otherwise,} \end{cases}$$

we get $\mathbf{W} = \mathbf{A}$, and so the classic version of the algorithm is a particular case of the revised one. Let $G_\sigma = (N_\sigma, A_\sigma)$, with $|N_\sigma| = n$, be the weighted digraph we are considering, described with matrix $\mathbf{W}$. In this revised formulation of the algorithm we get, normalization apart:

$$\begin{aligned} \mathbf{a}^{(k)} &= (\mathbf{W}^T\mathbf{W})^{k-1}\mathbf{W}^T\mathbf{u} \\ \mathbf{h}^{(k)} &= (\mathbf{W}\mathbf{W}^T)^k\mathbf{u}, \end{aligned}$$

where $k = 1, 2, \ldots$.

It is clear that matrices $\mathbf{W}^T\mathbf{W}$ and $\mathbf{W}\mathbf{W}^T$ play a fundamental role in the study of the convergence of this algorithm, so that it is worthwhile studying their properties.

## 4. Properties of the matrices $\mathbf{W}^T\mathbf{W}$ and $\mathbf{W}\mathbf{W}^T$

Let us consider some properties of the matrix $\mathbf{W}^T\mathbf{W}$ that will be useful to see deeper inside the behaviour of Kleinberg's algorithm and its revised version.

Let us say $(\mathbf{W})_{ij} \triangleq w_{ij}$ the entry $(i, j)$ of the matrix $\mathbf{W}$. Moreover, let $\mathbf{B} \triangleq \mathbf{W}^T\mathbf{W}$ and $(\mathbf{B})_{ij} \triangleq b_{ij}$. From the definition of matrix product we get: $b_{ij} = \sum_{k=1}^{n} w_{ki}w_{kj}$, since the first matrix $\mathbf{W}$ is transposed, i.e.

$(\mathbf{W}^T)_{ik} = w_{ki}$; from this formula we see that the generic entry $b_{ij}$ is a nonzero entry if and only if there is at least one node $k$ of the digraph $G_\sigma$ with outgoing arcs towards nodes $i$ and $j$ at the same time. Matrix $\mathbf{W}^T\mathbf{W}$ is an $n \times n$ real and nonnegative symmetric matrix with a particular structure, so that all its $n$ eigenvalues are real and nonnegative, as we can get from Theorems 2.1, 2.5 and 2.4. It is useful to consider matrix $\mathbf{W}^T\mathbf{W}$ as having an underlying weighted graph $G_w$, where an edge between vertex $i$ and vertex $j$ exists if and only if in the original digraph $G_\sigma$ there is at least one node with outgoing arcs towards nodes $i$ and $j$. So we can consider the graph $G_w$ as to be composed of a certain number of connected components, with no connections between each other. With no loss of generality we can group together the vertices of each of these components, so that all the vertices belonging to the same connected component are identified by consecutive numbers. In other words, if we get $m$ connected components, so that they have $k_1$, $k_2$, ..., $k_m$ vertices, we may suppose that the first component has the vertices $1, 2, \ldots, k_1$, the second has the vertices $k_1 + 1, k_1 + 2, \ldots, k_1 + k_2$ and so on. After this preliminary work, matrix $\mathbf{B}$ is a matrix with diagonal blocks, with this structure:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \ldots & \mathbf{0} \\ \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{B}_m \end{bmatrix} \tag{2}$$

where $\mathbf{B}_i$ is the matrix having as underlying graph the weighted component $i$. From (2) we get that the eigenvalues of $\mathbf{B}$ are given by the eigenvalues of $\mathbf{B}_1$, plus the eigenvalues of $\mathbf{B}_2$, plus the eigenvalues of $\mathbf{B}_3$, and so on till the eigenvalues of $\mathbf{B}_m$. Each of these matrices $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_m$ is a real, irreducible and nonnegative symmetric matrix, with real nonnegative eigenvalues, since the eigenvalues of $\mathbf{W}^T\mathbf{W}$ are real and nonnegative. Note that some of these matrices can be $\mathbf{0}$, i.e. a $1 \times 1$ matrix with a single zero entry. Since we are not considering the trivial case, that is the case in which $\mathbf{W}^T\mathbf{W} = \mathbf{0}$, in the following we will suppose that at least one of the matrices $\mathbf{B}_i$, $i = 1, 2, \ldots, m$ has some nonzero entries. From Corollary 2.7 and Theorem 2.6 we get that each of the matrices $\mathbf{B}_i$, $i = 1, 2, \ldots, m$ which is different from $\mathbf{0}$ has a strictly dominant eigenvalue, i.e. an eigenvalue which is strictly greater than all the other eigenvalues of the matrix; moreover, all the entries of its associated eigenvector are greater than 0. We can see that the eigenvectors of matrix $\mathbf{B}$ can be obtained from the eigenvectors of the matrices $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_m$ by just considering 0 the entries corresponding to the other matrices. For example, starting from the eigenvector $\mathbf{u}_{kj}$,

i.e. the eigenvector $k$ of matrix $\mathbf{B}_j$, for matrix $\mathbf{B}$ we get the eigenvector:

$$\begin{array}{r} 1\left\{\right. \\ 2\left\{\right. \\ \\ j\left\{\right. \\ \\ m\left\{\right. \end{array} \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \\ \hline \vdots \\ \hline \mathbf{u}_{kj} \\ \hline \vdots \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] \tag{3}$$

Starting from each eigenvector of $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_m$ we get $n$ eigenvectors of $\mathbf{B}$ with this structure, which form a basis of $\mathbb{R}^n$. Theorem 2.2 assures that each matrix $\mathbf{B}_i$, $i = 1, 2, \ldots, m$ has an orthonormal basis consisting of its eigenvectors, and this is the basis we will consider in the following, so that even $\mathbf{B}$ will have an orthonormal basis. Finally, since each matrix $\mathbf{B}_i$ is real and symmetric, from Corollary 2.3 we get that $\mathbf{B}_i$ is orthogonally diagonalizable, meaning that an orthogonal matrix $\mathbf{Q}_i$ such that

$$\mathbf{Q}_i^{-1}\mathbf{B}_i\mathbf{Q}_i = \begin{bmatrix} \lambda_{1i} & 0 & \ldots & 0 \\ 0 & \lambda_{2i} & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \ldots & \lambda_{k_i i} \end{bmatrix}$$

does exist, where $\lambda_{1i}, \lambda_{2i}, \ldots, \lambda_{k_i i}$ are the $k_i$ real and nonnegative eigenvalues of $\mathbf{B}_i$; note that these eigenvalues are not necessarily all different from each other. From the considerations developed before, we have:

THEOREM 4.1. *Matrix $\mathbf{W}^T\mathbf{W}$ has a strictly dominant eigenvalue if and only if one of the matrices $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_m$ has a strictly dominant eigenvalue that is greater than the strictly dominant eigenvalue of every other matrix of this multiset of matrices.*

COROLLARY 4.2. *If the matrix $\mathbf{W}^T\mathbf{W}$ represents a connected graph $G_w$, then it has a strictly dominant eigenvalue.*

All the results obtained for the matrix $\mathbf{W}^T\mathbf{W}$ can also be obtained for the matrix $\mathbf{W}\mathbf{W}^T$. Besides, these two matrices have the same

characteristic polynomial and so the same eigenvalues, with the same algebraic multiplicity (Salce 1993).

Note that if $\mathbf{W} = \mathbf{A}$, so that

$$w_{ij} = a_{ij} = \begin{cases} 1 & \text{if an arc from node } i \text{ to node } j \text{ exists,} \\ 0 & \text{otherwise,} \end{cases}$$

we get the matrix $\mathbf{A}^T\mathbf{A}$ which has an interesting interpretation in bibliometrics. In fact, interpreting each arc from node $i$ to node $j$ as a citation of the document represented by node $j$ from the document represented by node $i$, $(\mathbf{A}^T\mathbf{A})_{ij}$ gives the number of documents that cite the documents $i$ and $j$ at the same time; this quantity is called *co-citation*. On the other hand, matrix $\mathbf{A}\mathbf{A}^T$ has a generic $(i, j)$ entry which represents the number of documents cited by both nodes $i$ and $j$; this quantity is called *bibliographic coupling*.

## 5. Convergence of the revised algorithm

Now we are ready to investigate the convergence of Kleinberg's HITS algorithm in its revised formulation. In his seminal paper (Kleinberg 1999) Kleinberg asserts that the classic HITS algorithm makes the authority vector sequence $\mathbf{a}^{(k)}$, $k = 1, 2, \ldots$, converge to the strictly dominant eigenvector of matrix $\mathbf{W}^T\mathbf{W}$ if this strictly dominant eigenvector exists, and the hub vector sequence $\mathbf{h}^{(k)}$, $k = 1, 2, \ldots$, converge to the strictly dominant eigenvector of matrix $\mathbf{W}\mathbf{W}^T$ if this strictly dominant eigenvector exists. If these matrices do not have a strictly dominant eigenvector, i.e. two or more eigenvalues have the maximum value, in Kleinberg (1999) it is merely asserted that the algorithm converges anyway. As we have seen in Section 4, in general these dominant eigenvectors are not unique, so that an in-depth mathematical treatment of this subject has to deal with this problem. Moreover, we are dealing with a more general formulation of Kleinberg's algorithm, and we want to see if the conclusions drawn for the classic algorithm are also valid for the general one.

So, let us consider the formula:

$$\mathbf{a}^{(k)} = (\mathbf{W}^T\mathbf{W})^{k-1}\mathbf{W}^T\mathbf{u} \qquad k = 1, 2, \ldots \tag{4}$$

which gives us the value of the authority vector at the step $k$ of the revised HITS algorithm, before normalization. The linearity of the system we are considering allows us to work with this formula, and to consider its normalization only when we need it, since it makes no difference whether we consider a normalization at each step or only

the final normalization. Moreover, in the following we will consider the structure (2) for matrix $\mathbf{W}^T\mathbf{W} = \mathbf{B}$, since, as we have seen before, this fact does not limit the generality of our treatment. So, let us consider the formula (4), and suppose that matrix $\mathbf{W}^T\mathbf{W}$ has the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ belonging to the eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, respectively. As we have seen in Section 4 each of these eigenvectors is related to a particular diagonal block of matrix (2), so that it has entries 0 corresponding to the other matrices, as illustrated in formula (3). If more than one matrix $\mathbf{B}_i$ has a strictly dominant eigenvalue equal to $\lambda_1$, i.e. the dominant eigenvalue of $\mathbf{B}$, then we have an example of the more general case, that is the one we discuss in this present section. So, let us suppose $\lambda_1 = \lambda_2 = \cdots = \lambda_r > \lambda_{r+1} \geq \lambda_{r+2} \geq \cdots \geq \lambda_n$, i.e. we have $r$ eigenvalues with maximum value. Their eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ are related to the $r$ different matrices $\mathbf{B}_{h_1}, \mathbf{B}_{h_2}, \ldots, \mathbf{B}_{h_r}$, so that $\mathbf{v}_j,\ 1 \leq j \leq r$, has positive entries corresponding to the diagonal block $\mathbf{B}_{h_j}$, and 0 otherwise. In the following we will consider $\mathcal{B} \triangleq \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ as an orthonormal basis for $\mathbb{R}^n$. We can express vector $\mathbf{W}^T\mathbf{u}$ in terms of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$:

$$\mathbf{W}^T\mathbf{u} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_r\mathbf{v}_r + \alpha_{r+1}\mathbf{v}_{r+1} + \cdots + \alpha_n\mathbf{v}_n.$$

Since $\mathcal{B}$ is an orthonormal basis,

$$\alpha_j = \langle \mathbf{W}^T\mathbf{u}, \mathbf{v}_j \rangle \qquad j = 1, 2, \ldots, n$$

where $\langle \mathbf{W}^T\mathbf{u}, \mathbf{v}_j \rangle$ denotes the scalar product between $\mathbf{W}^T\mathbf{u}$ and $\mathbf{v}_j$. Moreover we can note that $\alpha_j > 0$, $j = 1, 2, \ldots, r$, since at least one entry of $\mathbf{W}^T$ in each row $i$ corresponding to a row of $\mathbf{B}_{h_j}$ in formula (2) must be greater than 0, because $\mathbf{B}_{h_j}$ represents a connected graph, so in every row it must have at least one entry that is greater than 0. In this last reasoning we have also implicitly used the fact that $\mathbf{W}^T$ is a nonnegative matrix, that $\mathbf{u} = [1\ 1 \ldots 1]^T$ and that all the entries of $\mathbf{v}_j$ in the rows corresponding to $\mathbf{B}_{h_j}$ are strictly positive. So we can write, from (4):

$$\begin{aligned}
\mathbf{a}^{(k)} &= (\mathbf{W}^T\mathbf{W})^{k-1}\mathbf{W}^T\mathbf{u} = \\
&= (\mathbf{W}^T\mathbf{W})^{k-1}(\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_n\mathbf{v}_n) = \\
&= \alpha_1\lambda_1^{k-1}\mathbf{v}_1 + \alpha_2\lambda_2^{k-1}\mathbf{v}_2 + \cdots + \alpha_n\lambda_n^{k-1}\mathbf{v}_n \qquad k = 1, 2, \ldots
\end{aligned}$$

Setting $\lambda \triangleq \lambda_1 = \lambda_2 = \cdots = \lambda_r$ we have

$$\begin{aligned}
\mathbf{a}^{(k)} &= \lambda^{k-1}(\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_r\mathbf{v}_r + \sum_{i=r+1}^{n} \frac{\alpha_i\lambda_i^{k-1}\mathbf{v}_i}{\lambda^{k-1}}) = \\
&= \lambda^{k-1}(\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_r\mathbf{v}_r + \mathbf{v}(k)) \qquad k = 1, 2, \ldots
\end{aligned}$$

where

$$\mathbf{v}(k) \triangleq \sum_{i=r+1}^{n} \frac{\alpha_i \lambda_i^{k-1} \mathbf{v}_i}{\lambda^{k-1}} \qquad k = 1, 2, \ldots$$

is a sequence of vectors so that $\lim_{k \to +\infty} \mathbf{v}(k) = \mathbf{0}$; in fact all the entries vanish as $k \to +\infty$, since $\lambda_i / \lambda < 1$ if $r + 1 \leq i \leq n$. The algorithm makes us compute

$$\lim_{k \to +\infty} \frac{\mathbf{a}^{(k)}}{\|\mathbf{a}^{(k)}\|}$$

where $\|\mathbf{a}^{(k)}\| = \|\lambda^{k-1}(\alpha_1 \mathbf{v}_1 + \cdots + \alpha_r \mathbf{v}_r + \mathbf{v}(k))\| = \lambda^{k-1} \|\alpha_1 \mathbf{v}_1 + \cdots + \alpha_r \mathbf{v}_r + \mathbf{v}(k)\|$ and, since every vector norm $\|\mathbf{x}\|$, with $\mathbf{x} = [x_1 \; x_2 \ldots x_n]^T$, is a continuous function of the variables $x_1, x_2, \ldots, x_n$ (Comincioli 1995):

$$\lim_{k \to +\infty} \|\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_r \mathbf{v}_r + \mathbf{v}(k)\| = \|\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_r \mathbf{v}_r\|$$

so that

$$\lim_{k \to +\infty} \frac{\mathbf{a}^{(k)}}{\|\mathbf{a}^{(k)}\|} = \frac{\alpha_1 \mathbf{v}_1 + \cdots + \alpha_r \mathbf{v}_r}{\|\alpha_1 \mathbf{v}_1 + \cdots + \alpha_r \mathbf{v}_r\|} \; . \tag{5}$$

From (5) we get that *under whatever norm, the revised HITS algorithm —and so, as a particular case, Kleinberg's HITS algorithm—converges to a unit authority vector which is a linear combination of all the dominant eigenvectors of* $\mathbf{W}^T \mathbf{W}$.

Following the same line of reasoning we can obtain for the hub vector:

$$\lim_{k \to +\infty} \frac{\mathbf{h}^{(k)}}{\|\mathbf{h}^{(k)}\|} = \frac{\beta_1 \mathbf{w}_1 + \beta_2 \mathbf{w}_2 + \cdots + \beta_r \mathbf{w}_r}{\|\beta_1 \mathbf{w}_1 + \beta_2 \mathbf{w}_2 + \cdots + \beta_r \mathbf{w}_r\|} \tag{6}$$

where we have denoted with $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_r$ the $r$ dominant eigenvectors of $\mathbf{W}\mathbf{W}^T$ and with $\beta_i$, $i = 1, 2, \ldots, r$, the scalar product:

$$\beta_i \triangleq \langle \mathbf{u}, \mathbf{w}_i \rangle \qquad i = 1, 2, \ldots, r$$

where, from the structure of $\mathbf{u}$ and $\mathbf{w}_i$, we get $\beta_i > 0$, $i = 1, 2, \ldots, r$. Note that the number of dominant eigenvectors for the two matrices $\mathbf{W}^T \mathbf{W}$ and $\mathbf{W}\mathbf{W}^T$ is the same, since these two matrices have the same characteristic polynomial and Theorem 2.2 holds. From (6) we get that *under whatever norm, the revised HITS algorithm—and so, as a particular case, Kleinberg's HITS algorithm—converges to a unit hub vector which is a linear combination of all the dominant eigenvectors of* $\mathbf{W}\mathbf{W}^T$.

## 6. Order of operations and symmetry

In the classic HITS algorithm the order of the basic operations on the hub and authority vectors is fixed, since the algorithm starts from the definition of the hub vector

$$\mathbf{h}^{(0)} = \mathbf{u}$$

and it carries out the following steps:

$$\begin{aligned} \mathbf{a}^{(1)} &= \mathbf{A}^T\mathbf{h}^{(0)} \\ \mathbf{h}^{(1)} &= \mathbf{A}\mathbf{a}^{(1)} \\ \mathbf{a}^{(2)} &= \mathbf{A}^T\mathbf{h}^{(1)} \\ &\vdots \end{aligned}$$

as we have seen in Section 3. Since this seems an arbitrary choice, we might wonder if it is the same to start with the update of hub values, that is

$$\begin{aligned} \mathbf{a}^{(0)} &= \mathbf{u} \\ \mathbf{h}^{(1)} &= \mathbf{A}\mathbf{a}^{(0)} \\ \mathbf{a}^{(1)} &= \mathbf{A}^T\mathbf{h}^{(1)} \\ &\vdots \end{aligned}$$

In other words, calling, as usual, $\mathbf{a}^{(k)}$, $\mathbf{h}^{(k)}$, $k = 1, 2, \ldots$ the authority and hub vectors we have at step $k$ following the classic algorithm, and $\mathbf{a}_{[i]}^{(k)}$, $\mathbf{h}_{[i]}^{(k)}$, $k = 1, 2, \ldots$ the authority and hub vectors we have at step $k$ following the algorithm with the basic operations inverted, we would like to know if

$$\lim_{k\to+\infty} \mathbf{a}^{(k)} \qquad \text{equals} \qquad \lim_{k\to+\infty} \mathbf{a}_{[i]}^{(k)}$$

and if

$$\lim_{k\to+\infty} \mathbf{h}^{(k)} \qquad \text{equals} \qquad \lim_{k\to+\infty} \mathbf{h}_{[i]}^{(k)} \ .$$

This study concerns not only the problem of the order of the basic operations but also that of the symmetry of the algorithm. According to Borodin et al. (2001), a link analysis algorithm is said to be *symmetric* if inverting all the arcs in the digraph constructed by the algorithm simply interchanges the hub and authority values. The problem of symmetry is not a mere curiosity about the behaviour of the algorithm. For example, if we consider a stemming application of the algorithm, it can seem more "natural" to consider a link from a prefix to a suffix

for each word but, at the same time, what is especially interesting is the appropriateness, or authoritativeness, of the prefix as a stem: this seems to be given by the authority weight of the prefix provided by the algorithm which uses inverted links, that is links from suffixes to prefixes. So, is it the same to consider the hub weight of the prefixes in the original formulation as to consider their authority weights using the inverted links?

The first result we have obtained on this point regards the equivalence between the problem of the order of the basic operations and that of the symmetry of the algorithm. Even if until here we have presented the question by referring to the classic HITS algorithm, from now on we can work again on the revised algorithm; studying this more general form does not affect the complexity of the treatment.

THEOREM 6.1. *The revised HITS algorithm—and so, as a particular case, the HITS algorithm—are symmetric if and only if the changing of the order of the basic operations does not affect the results provided by the application of the algorithm.*

*Proof.* If $\mathbf{a}^{(k)}$, $\mathbf{h}^{(k)}$, $\mathbf{a}_{[i]}^{(k)}$, $\mathbf{h}_{[i]}^{(k)}$, $k = 1, 2, \ldots$ have the meaning previously defined in this section, it is

$$\begin{aligned}
\mathbf{a}^{(k)} &= (\mathbf{W}^T\mathbf{W})^{k-1}\mathbf{W}^T\mathbf{u} \\
\mathbf{a}_{[i]}^{(k)} &= (\mathbf{W}^T\mathbf{W})^{k}\mathbf{u} \\
\mathbf{h}^{(k)} &= (\mathbf{W}\mathbf{W}^T)^{k}\mathbf{u} \\
\mathbf{h}_{[i]}^{(k)} &= (\mathbf{W}\mathbf{W}^T)^{k-1}\mathbf{W}\mathbf{u}
\end{aligned}$$

where again $k = 1, 2, \ldots$. On the other hand, the digraph with inverted arcs is described by the matrix $\mathbf{W}^T$, since its generic entry $(i, j)$ is equal to the entry $(j, i)$ of the matrix describing the original digraph. Therefore, the authority and hub vectors for the digraph with inverted arcs are given by:

$$\begin{aligned}
\mathbf{a}_{[s]}^{(k)} &= (\mathbf{W}\mathbf{W}^T)^{k-1}\mathbf{W}\mathbf{u} \\
\mathbf{h}_{[s]}^{(k)} &= (\mathbf{W}^T\mathbf{W})^{k}\mathbf{u}
\end{aligned}$$

where $k = 1, 2, \ldots$, so that

$$\mathbf{a}_{[s]}^{(k)} = \mathbf{h}_{[i]}^{(k)} \qquad \mathbf{h}_{[s]}^{(k)} = \mathbf{a}_{[i]}^{(k)} \qquad k = 1, 2, \ldots \tag{7}$$

From (7) the result immediately follows.

We can now prove that the HITS algorithm, and so its revised version, are not symmetric, and so, from Theorem 6.1, they are not

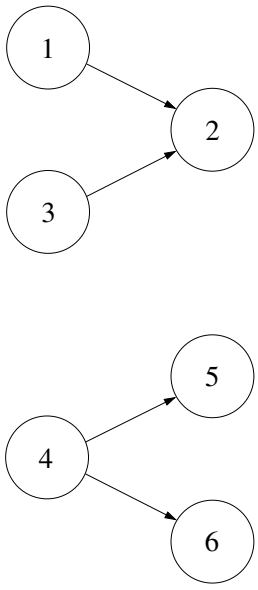


*Figure 2.* A counter example to prove that the HITS algorithm is not symmetric.

independent of the order of update operations, either. Let us consider the digraph in Fig. 2 and suppose that each arc has weight 1; the related adjacency matrix $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

so that

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The eigenvalues of this matrix are given by: $\lambda \triangleq \lambda_1 = \lambda_2 = 2$, $\lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 0$; it has two dominant eigenvectors, with eigenvalue

2, given by:

$$\mathbf{v}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Now we have

$$\mathbf{A}^T\mathbf{u} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\langle \mathbf{A}^T\mathbf{u}, \mathbf{v}_1 \rangle = \begin{bmatrix} 0 & 2 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 2$$

$$\langle \mathbf{A}^T\mathbf{u}, \mathbf{v}_2 \rangle = \begin{bmatrix} 0 & 2 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{2}{\sqrt{2}}$$

Thus, using the theory developed before, we can say that the sequence of authority vectors converges to the following vector, before normalization:

$$2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{2}{\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

For example, using $\|\cdot\|_2$, the limit vector is given by

$$\frac{1}{\sqrt{6}}\begin{bmatrix}0\\2\\0\\0\\1\\1\end{bmatrix} \tag{8}$$

Now, if we consider the digraph obtained by inverting the arcs of the digraph depicted in Fig. 2, we get the digraph depicted in Fig. 3. Its adjacency matrix is given by:

$$\mathbf{A}_{[s]} = \begin{bmatrix}0&0&0&0&0&0\\1&0&1&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&1&0&0\\0&0&0&1&0&0\end{bmatrix} = \mathbf{A}^T$$

so that

$$\mathbf{A}_{[s]}\mathbf{A}_{[s]}^T = \begin{bmatrix}0&0&0&0&0&0\\0&2&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&1&1\\0&0&0&0&1&1\end{bmatrix} = \mathbf{A}^T\mathbf{A}$$

and the eigenvalues of $\mathbf{A}_{[s]}\mathbf{A}_{[s]}^T$ are the eigenvalues of $\mathbf{A}^T\mathbf{A}$: $\lambda \triangleq \lambda_1 = \lambda_2 = 2$, $\lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 0$. Its dominant eigenvectors are again:

$$\mathbf{v}_1 = \begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix} \qquad \mathbf{v}_2 = \begin{bmatrix}0\\0\\0\\0\\\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{bmatrix}$$

and we have

$$\langle \mathbf{u}, \mathbf{v}_1 \rangle = 1 \qquad \langle \mathbf{u}, \mathbf{v}_2 \rangle = \frac{2}{\sqrt{2}}$$

so that the sequence of the *hub* vectors has a limit of:

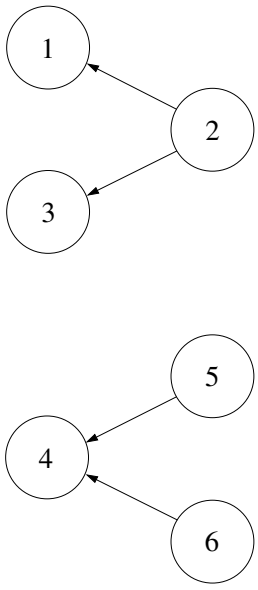


*Figure 3.* The digraph of Fig. 2 with the arcs inverted.

$$1 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{2}{\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

that is, using $\|\cdot\|_2$:

$$\frac{1}{\sqrt{3}} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

which differs from (8). This proves that the algorithm is not symmetric.

From this counter example, and using the equivalence stated in Theorem 6.1, we can see that in general the application of the HITS algorithm, and so of its revised version, gives a limit vector that depends on the starting vector.

## 7. Some interesting consequences

In this paper we carried out an analysis of Kleinberg's HITS algorithm, considered in general, so independently of its specific Web application. Our aim was to try to illuminate the behaviour of this algorithm, since some recent works have convinced us that interesting applications could be found for it in fields far removed from that of Web searching. Moreover, we were interested in studying a generalized version of the algorithm, which we called the revised HITS algorithm, so that the classic version of the algorithm is a particularization of the revised one. Following these starting ideas we found that the key to mathematically understanding the behaviour of this algorithm is in the structure of two matrices, $\mathbf{W}^T\mathbf{W}$ and $\mathbf{W}\mathbf{W}^T$. We found that these matrices can always be transformed into block diagonal matrices, where each block can be seen as representing a connected component of a weighted graph. This analysis formed the basis to proving the convergence of the algorithm to a limit vector, even in the cases in which these matrices have more than one dominant eigenvector; moreover, we found the analytic expression of this limit vector. Finally, we proved the equivalence between the problem of the symmetry of the algorithm and that of the order of the basic operations performed over the hub and authority vectors, and we found that the algorithm is not symmetric.

The theoretical analysis performed here has the purpose of throwing light on the behaviour of the algorithm, and so its range should be broad. Pursuing this aim, we have generalized the HITS algorithm, and studied it in the abstract. The utility of this kind of study should go beyond the mathematical explanation of some experimental results. Moreover, the possible applications of its results could go beyond the ideas the authors of the analysis now have. This is a common situation in theoretical analyses. In this section, however, we would like to qualitatively explain the correlation between the dominant eigenvalues of a nonnegative real symmetric matrix and the structure of the weighted graph underlying it. Afterwards we would like to indicate a couple of situations in which our analysis clarifies the actual behaviour of the algorithm, or its right application.

### 7.1. Graphs and dominant eigenvalues

In the analysis we have developed in Sections 4 and 5 we have proved that the revised HITS algorithm converges to a unit authority vector which is a linear combination of all the dominant eigenvectors of $\mathbf{W}^T\mathbf{W}$. Moreover, we have proved a similar result for hubs; nevertheless, to make our treatment more specific, in the rest of this subsection

we will focus on the authority vector and so on the matrix $\mathbf{W}^T\mathbf{W}$. As usual, all the facts which will be shown are however also valid for the hub vector and the matrix $\mathbf{W}\mathbf{W}^T$.

Using the results of our analysis, now we know that the limit vector has a block structure, where the generic block $j$ is a nonzero block if and only if $\mathbf{B}_j$ has a dominant eigenvalue which equals the dominant eigenvalue of $\mathbf{B}$. Since $\mathbf{B}_j$ represents a weighted connected graph that is the $j$-th connected component of the weighted graph represented by $\mathbf{B}$, it is a central point to grasp which are the relations between a weighted graph and the dominant eigenvalues of the matrix representing it. So, what does it mean if a weighted connected graph is represented by a matrix whose dominant eigenvalues are large? Which are the qualitative differences between weighted connected graphs that are represented by matrices with different dominant eigenvalues?

To try to answer these questions, we need to remind the reader of the following theorem (Godsil and Royle 2001):

THEOREM 7.1. *Suppose* $\mathbf{A}$ *is a real nonnegative* $n \times n$ *matrix whose underlying directed graph* $G$ *is strongly connected. Suppose* $\mathbf{A}_1$ *is a real nonnegative* $n \times n$ *matrix such that* $\mathbf{A} - \mathbf{A}_1$ *is nonnegative. Then* $\rho(\mathbf{A}_1) \leq \rho(\mathbf{A})$*, with equality if and only if* $\mathbf{A}_1 = \mathbf{A}$*.*

Applying this theorem to our case, it is easy to see that if we have an initial weighted connected graph $G_1$ and we move to another graph $G$ obtained from $G_1$ just adding one or more edges with positive weights, or increasing the weights of one or more pre-existent edges, the new weighted graph is represented by a matrix with larger dominant eigenvalues. In Fig. 4 three simple weighted graphs, $G_1$, $G_2$, and $G_3$, are depicted. Starting from the initial graph $G_1$, graph $G_2$ is obtained from $G_1$ by adding an edge with weight 0.1 between vertices 2 and 4. Graph $G_3$ is obtained from graph $G_2$ by increasing the weight of the edge between vertices 1 and 2. Calling $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ the matrices describing $G_1$, $G_2$ and $G_3$, respectively, it is $\rho(\mathbf{A}_1) < \rho(\mathbf{A}_2) < \rho(\mathbf{A}_3)$. Applying these remarks to Kleinberg's algorithm, we can say that each block $\mathbf{B}_j$ has a dominant eigenvalue which gets bigger as each one of block $\mathbf{B}_j$'s entries enlarges. So, considering the digraph $G_\sigma$ and the generic entry $(i, j)$ of the block $B_j$, when the number of nodes $k$ pointing to $i$ and $j$ simultaneously becomes bigger, or the weights of arcs $(k, i)$ and $(k, j)$ get bigger, the dominant eigenvalue of $B_j$ gets larger.

Moreover, keeping in mind the following theorem (Salce 1993):

THEOREM 7.2. *If* $\mathbf{A}$ *is a real nonnegative* $n \times n$ *matrix, whose generic entry* $(i, j)$ *is denoted with* $a_{ij}$*, we have:*

$$\min_i \sum_j a_{ij} \leq \rho(\mathbf{A}) \leq \max_i \sum_j a_{ij}$$

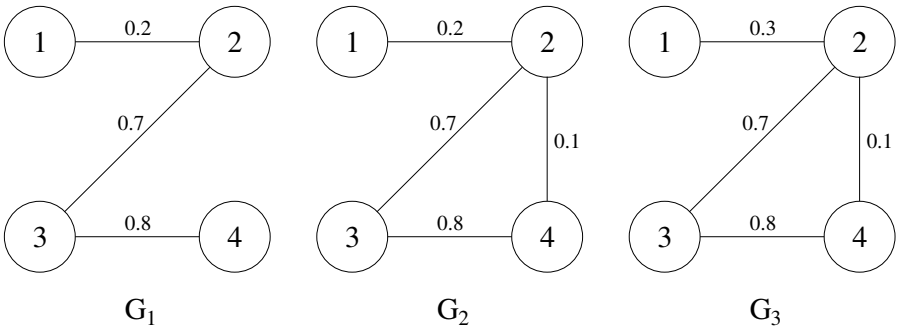


*Figure 4.* Three weighted graphs represented by matrices with increasingly larger dominant eigenvalues.

$$\min_j \sum_i a_{ij} \leq \rho(\mathbf{A}) \leq \max_j \sum_i a_{ij} \ ,$$

we can see that there is a kind of relation between the size of the dominant eigenvalue of a block $\mathbf{B}_j$ and the size of the block (number of rows, and so number of columns). Moreover, a qualitative rule can be: between two blocks with different sizes but with entries with comparable values, the largest one has probably a larger dominant eigenvalue. This means, of course, that between two weighted connected graphs with edges with comparable weights, the one with a larger number of vertices is probably represented by a matrix with a larger dominant eigenvalue.

### 7.2. How the HITS algorithm works on the Web

Now we are ready to try to explain the behaviour of the classic algorithm in its original Web application. To make our explanation concrete, let us suppose there are a couple of Web communities in the digraph $G_\sigma = (N_\sigma, A_\sigma)$ on which the algorithm works; this situation is depicted in Fig. 5. This figure represents a simple and neat topology; the actual digraphs on which the algorithm works can, of course, have a much more complex topology.

At a quick glance at the figure, one could imagine that the algorithm will give, as authorities, a combination of the pages in the left community and in the right one. In other words, we might assume that some pages with high authority weight could be in one of the two communities and some other pages with high authority weight in the other. Our results show that this is generally false: unless the first eigenvalue of the two blocks of $\mathbf{W}^T\mathbf{W}$ representing the two authorities communities is the same, a rather improbable event as we can infer from the facts

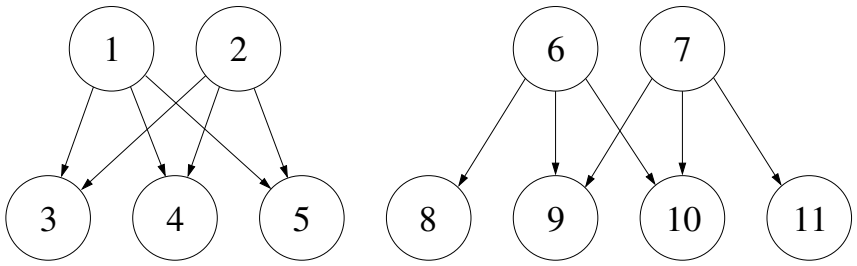


*Figure 5.* A digraph $G_\sigma$ with two separate communities.

shown in the previous subsection, one community wins, and the weights of the authorities of the other community vanish. On the other hand, all the authority weights of the authorities of the winning community will be positive. For example, in the simple case of Fig. 5, the left community wins, and the algorithm gives a positive authority weight only to the authorities in this community, that is only to the nodes 3, 4, and 5, while all the other nodes' authority weight equals zero. This means that the algorithm is intrinsically community-oriented, that is it tends to emphasize one authorities community. So our analysis can, at least partially, explain the so-called tightly knit community (TKC) effect (Lempel and Moran 2001): it was experimentally seen that, especially with some queries, Kleinberg's algorithm tends to emphasize only one of the communities of Web pages in the answer, missing the topics in the other communities.

### 7.3. An application to stemming

In this subsection we would like to outline an application of the HITS algorithm to stemming. Here we have the purpose of giving an idea of an application of the algorithm in a field far removed from that of Web searching. Moreover, we would like to show how in this context the problem of symmetry, which was addressed in Section 6, naturally arises. The reader interested in the first article written on this subject should read Bacchin et al. (2002), in which the idea of applying the core of Kleinberg's algorithm to stemming is enriched by statistical interpretations; in Bacchin et al. (2002) and in Di Nunzio et al. (2003) many experimental results can be found.

The stemming process (Baeza-Yates and Ribeiro-Neto 1999) consists of reducing each word of a document or of a query to its grammatical root. This process can improve both the effectiveness and efficiency of information retrieval systems. Among stemming strategies, the most commonly used is the suffix removal strategy, whose purpose is to divide

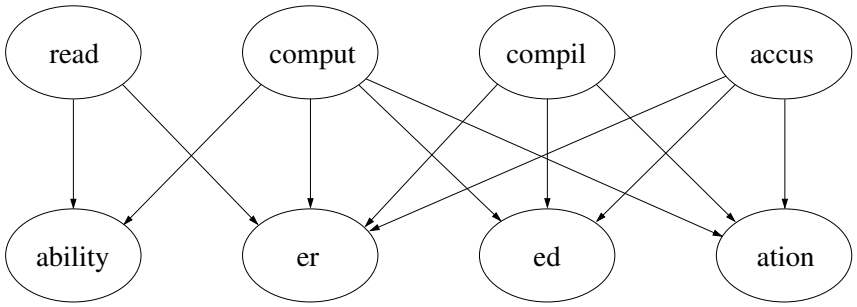


*Figure 6.* Mutual reinforcement between stems and derivations.

each word into two parts, a stem and a derivation, appropriately removing the last part of the word, i.e. its suffix. For instance, in the words `connected`, `connecting`, `connection` and `connections`, `connect` is the stem, and `ed`, `ing`, `ion` and `ions` are the derivations, respectively.

The first part of the algorithm proposed in Bacchin et al. (2002) divides each word into different pairs (prefix, suffix), considering all the possibilities; then a digraph is built, in which each node represents a prefix or a suffix, and an arc between nodes $i$ and $j$ exists if, and only if, there is a word with a prefix represented by $i$ and a suffix represented by $j$. Now, how can we distinguish, among all the possible pairs (prefix, suffix), the pair (stem, derivation) for each word? The idea underlying the algorithm is that even between prefixes and suffixes there is a kind of mutual reinforcement, that is a good prefix points to many good suffixes, while a good suffix is pointed to by many good prefixes. To better understand the idea underlying this application, let us see Fig. 6, taken from Di Nunzio et al. (2003). In this figure, the mutual reinforcement between the stems `read`, `comput`, `compil`, `accus` and the derivations `ability`, `er`, `ed`, `ation` is shown. Note that if we considered generic pairs (prefix, suffix), the digraph would be more sparse, without the above illustrated mutual reinforcement.

The considerations made above suggest that we apply the core of Kleinberg's algorithm to the prefixes-suffixes digraph, to find the stems of words. In this case we have a typical situation in which the symmetry problem arises: in fact, what we are looking for are good prefixes, that is prefixes which are candidates to be stems, but a natural application of the algorithm would indicate good suffixes, since the algorithm was born to find good authorities. Now the question is: do the best hubs for the natural application of the algorithm exactly correspond to the best authorities for the application of the algorithm to the digraph with reversed arcs? The theory developed in Section 6 allows us to reply in the negative to this question. Therefore, the theory developed

in Section 6 could suggest the application of the core of Kleinberg's algorithm to a new digraph, with inverted arcs, to tackle the stemming problem in an innovative way.

## 8. Future work

Our future work will focus on two specific topics:

1. First of all, we would like to study the possibility of finding an a priori defined number of steps, on the basis of the structure of the starting digraph, so that the iterative calculation of the authority and hub vectors can stop after this number of steps, since *the order* of the values of the authority and hub entries does not change any more after it. This seems to be an interesting problem, since what is really important in using the authority, and in some cases the hub vectors, is the ranking supplied by these vectors, more than the actual value of their entries. Moreover, the problem is original since it regards the ranking of the entries, while Numerical Analysis literature (Golub and Van Loan 1996) considers the distance between the limit vector and the vector we get after $k$ iterations.
2. An in-depth analysis of eigenvector-based techniques should be undertaken. In particular, in his experiments (Kleinberg 1999), Kleinberg found that the second eigenvector, in relation to certain queries, brought the logical separation between some communities of Web pages to the surface. This is a facet of the algorithm's behaviour that has not yet been analytically understood, and we think that this present work, and especially what we have shown in Section 4, could be a useful starting point to deal with this problem.

## Acknowledgements

During the preparation of this paper we greatly benefited from the stimulating working environment provided by Michela Bacchin, Nicola Ferro and Massimo Melucci. We would also like to thank the anonymous reviewers of this paper for the suggestions they gave us to improve its quality.

## References


Agosti M and Melucci M (2001) Information Retrieval on the Web. In: Agosti M, Crestani F and Pasi G, eds. Lectures on Information Retrieval. Number 1980 in LNCS. Springer, Berlin, 2001. pp. 242–285.

Bacchin M, Ferro N and Melucci M (2002) University of Padua at CLEF 2002: experiments to evaluate a statistical stemming algorithm. In: Peters C, ed. Working Notes for the CLEF 2002 Workshop, 2002. pp. 161–168.

Baeza-Yates R and Ribeiro-Neto B (1999) Modern Information Retrieval. ACM Press, New York.

Borodin A, Roberts GO, Rosenthal JS and Tsaparas P (2001) Finding authorities and hubs from link structures on the World Wide Web. In: Proceedings of the World Wide Web Conference, 2001. http://www.www10.org/cdrom/papers/314/index.html.

Brin S and Page L (1998) The anatomy of a large scale hypertextual Web search engine. In: Proceedings of the World Wide Web Conference, 1998. http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm.

Comincioli V (1995) Analisi Numerica: Metodi, Modelli, Applicazioni. McGraw-Hill, Milano.

Di Nunzio GM, Ferro N, Melucci M and Orio N (2003) The University of Padua at CLEF 2003: experiments to evaluate probabilistic models for automatic stemmer generation and query word translation. In: Peters C, ed. Working Notes for the CLEF 2003 Workshop, 2003. pp. 211–223. http://clef.iei.pi.cnr.it:2002/2003/WN_web/27.pdf.

Godsil C and Royle G (2001) Algebraic Graph Theory. Number 207 in Graduate Texts in Mathematics. Springer, New York.

Golub GH and Van Loan CF (1996) Matrix Computations. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, third edition, Baltimore.

Henzinger MR (2001) Hyperlink analysis for the Web. IEEE Internet Computing, 5(1):45–50.

Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5):604–632.

Lempel R and Moran S (2001) SALSA: The stochastic approach for link-structure analysis. ACM Transactions on Information Systems, 19(2):131–160.

Miller JC, Rae G, Schaefer F, Ward LH, LoFaro T and Farahat A (2001) Modifications of Kleinberg's HITS algorithm using matrix exponentiation and Web log records. In: Croft WB, Harper DJ, Kraft DH and Zobel J, eds. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, 2001. pp. 444–445.

Ng AY, Zeng AX and Jordan MI (2001) Stable algorithms for link analysis. In: Croft WB, Harper DJ, Kraft DH and Zobel J, eds. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, 2001. pp. 258–266.

Page L, Brin S, Motwani R and Winograd T (1998) The PageRank citation ranking: bringing order to the Web. Unpublished manuscript. http://google.stanford.edu/~backrub/pageranksub.ps (downloaded January 2002).

Salce L (1993) Lezioni sulle Matrici. Decibel-Zanichelli, Bologna.