

Mining Topological Importance From The Eigenvectors Of Directed Graphs

M. Burgess
Oslo University College

G. Canright and K. Engø-Monsen
Telenor Research

February 7, 2007

Abstract

We study methods for extracting importance scores for nodes on a directed graph, using only the topology of the graph as a guide. Such ranking has many applications in directed networks. Our strategy is to find an eigenvector, which is positive everywhere on the graph and can be interpreted as a global importance function for its topological content, avoiding significant modification of the graph topology. We review some alternative adjacency operators and show that, for a general reducible graph, there is no globally obtainable importance function without graph augmentation. Two new methods of augmenting graphs are then proposed to allow such a global ranking to be computed. Preliminary tests are shown to perform favourably alongside other well-known approaches that do make large modifications. Our work has applications to all fields where directed graphs are used. We discuss briefly some example applications, in particular World Wide Web page ranking.

(Keywords: Graph mining, web searches, node ranking, HITS, link analysis)

1 Introduction

Graphs are ubiquitous in science. They are used whenever one can define entities (nodes) and relations between them (links). In computer science, graphs are used to represent both logical and physical connections. Outstanding examples include computer networks, security systems [1], software graphs [2, 3], and the World Wide Web[4].

The analysis of a graph can be both illuminating and practical, as illustrated by the rich potential for applications alluded to in section 8 of the paper. We are interested in the basic and general problem of defining the *importance* function of the nodes in the graph, using a definition based solely on the topology of the

graph. Such a definition of importance arises naturally in applications where a human social process is present, such as contact networks and the World Wide Web. Specifically, we wish to explore possibilities that arise when one generalizes the notion of *eigenvector centrality*[5, 6] to directed graphs. This involves the finding of an eigenvector of the adjacency matrix which represents the graph, and then defining the ‘centrality’ of node i as the i th component of the (suitably normalized) eigenvector.

Importance ranking using the centrality concept typically uses the *principal eigenvector*, i.e. the eigenvector corresponding to the largest eigenvalue in a graph’s spectrum. Justification for this choice of vector may be found in ref. [6], for the case of undirected links. For directed links, similar reasoning is given in ref. [7]. In each case, the fundamental idea is that the centrality of a node is proportional to the sum of the centralities of the node’s neighbours. For undirected graphs, this means all of the node’s neighbours; for directed graphs, in- or out-neighbours may be distinguished. In colloquial terms, the ranking uses the principle that it’s not just how many connections you have, but how important those connections are, that matters. In each case, however, a precise statement of this reasoning gives an eigenvalue equation for the centrality.

New results in such topological ranking come rarely, leading to a number of alternative ways of ranking graphs[8, 9, 10, 11], however we offer such a contribution here. The principal eigenvector, for positive semi-definite matrices such as adjacency matrices, is itself positive definite in the undirected case, or positive semi-definite in the directed case. In order to define a graph-wide (global) measure of node importance from an eigenvector, it is desirable that that eigenvector be positive everywhere. Our main result in this paper is to find conditions giving a *positive definite eigenvector* (PDEV), and hence a global ranking, for a directed graph.

We note that the mathematics, and the results, differ qualitatively between directed and undirected graphs. In the directed case, one has a choice of matrices from which a principal eigenvector can be derived. These include the adjacency matrix A , its transpose, normalized versions of the above, and even products of them. Normalized adjacency matrices have been used in the famous PageRank algorithm [7]. Products AA^T and $A^T A$ were first explored in the HITS algorithm of Kleinberg [12]. Each of these latter combinations yields a symmetric matrix; but they are not the same unless the graph is undirected. Normalized (weight preserving) versions of these two HITS operators were studied in the SALSA method [13]. Finally, we mention that the backward analogue of the PageRank matrix has been discussed in ref. [14]. Much has been written on these, see for example refs. [15, 16, 17, 18].

The six matrix operators mentioned above collapse to only two choices when one removes the property of direction from the links. That is, all the normalized cases (the PageRank operator and its backward analogue, as well as the compound but normalized SALSA operators) become the same for the undirected case. However, it is easy to show [19] that the dominant eigenvector of a symmetric, positive semidefinite, weight preserving matrix is trivial, in the sense that a component i , of its eigenvector, is simply a constant times the node

degree k_i . In short: using the principal eigenvector of the normalized adjacency matrix for an undirected graph sets a node's importance to be proportional to its degree. Thus, if one uses this approach, it really is 'how many you know', and not 'who you know', that matters to the ranking.

In contrast, the two HITS operators both become simply A^2 for the case of undirected links (which give a symmetric adjacency matrix, $A = A^T$). The dominant eigenvector of A^2 is simply the dominant eigenvector of A . The centrality measure which is obtained from A is also known in social science, where it is termed 'eigenvector centrality' or EVC [5, 20]. The eigenvector giving EVC is not 'trivial', in the sense used above: that is, it is truly an instance where the centrality of your neighbours (and not just their number) determines your own centrality.

Each of these methods for defining centrality or importance gives different results. And yet these methods (as we will see below) do not exhaust the possibilities for choice of matrix for this approach to computing node importance.

2 Aims

In this paper, we wish to answer the question: what combination of matrix operator and structural properties will give a *positive definite eigenvector* (PDEV) for an entire directed graph?

In answering this question, we will draw heavily on a basic structural element of a directed graph, namely a subgraph known as a *strongly connected component* (SCC), defined below. We will also develop and use the notion of *dominance*, in which an SCC can impose its influence on other SCCs. Using these ideas, we will elucidate in some detail why, for the general directed graph, virtually any graph of interest will not give a satisfactory PDEV. In doing so, we will examine a pair of simple operators: the non-normalized adjacency matrix A , and its transpose, which have not, to our knowledge, been studied with regard to our main question before. Finally, we will use our understanding to propose two new methods (which we call '*sink remedies*') for curing the problem of sinks.

Each of our methods relies on an analysis of the SCC structure of a graph, which yields what we call a *metagraph*. This is closely related to the 'collapsed graph'[21] or 'component graph' known from mathematics). However, the two methods are rather different in nature, and give rather different results.

Much of our thinking is grounded in the very specific and practical problem of scoring and ranking Web pages in searches, using an importance function obtained from link analysis, and more specifically, from that class of link analysis techniques which uses eigenvectors. The above-cited PageRank [7], HITS [12], and SALSA [13] methods, along with the work of Ding et al [14], have been developed with this purpose in mind. Also, the Web graph provides a concrete application to have in mind, which is helpful for rendering our discussion less abstract. We wish to emphasize however that ranking of Web pages is only one possible application of these ideas, out of many. Our ideas and results are in principle applicable to any problem for which (i) a directed graph gives a useful

representation, and (ii) it is desired to evaluate the importance of the nodes of the graph, using the topology of the graph to determine the answer.

3 Notations and conventions

We use the symbol $G = \langle \mathcal{N}, \mathcal{E} \rangle$ for a directed graph, with a set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E}$. The adjacency matrix of the full graph is written A , with indices suppressed; its transpose is written A^T . Latin indices r, c are used for labelling the rows and columns of the adjacency matrix. These run from $1 \dots N$, where N is the number of elements in $\mathcal{N}$.

We assume that the matrix element A_{rc} of the adjacency matrix A may be thought of as $A_{r \rightarrow c}$. That is, $A_{rc} = 1$ if r points to c . The symbol λ is used as an unspecified eigenvalue of a matrix. ρ is reserved for the principal eigenvalue, or spectral radius of a strongly connected graph. A graph which has repeated eigenvalues in its spectrum is said to be *degenerate*.

The in-degrees of the nodes, i.e. the numbers of links that point to each node, is denoted by the vector $\vec{k}^{\text{in}}$; the out-degree vector, or number of outgoing links per node, is denoted $\vec{k}^{\text{out}}$. Recalling that A_{rc} implies $r = \text{row}$ and $c = \text{column}$, then in a directed graph, the row-sum of the adjacency matrix is the out-degree of the node whose row is summed:

$$\sum_c A_{rc} = k_r^{\text{out}}. \quad (1)$$

Similarly, the column sum is the in-degree of the node:

$$\sum_r A_{rc} = k_c^{\text{in}}. \quad (2)$$

We shall be most interested in A^T and matrices derived from this. A right principal eigenvector of this matrix will be denoted $\vec{\Psi}$, and a sub-vector of this will be denoted $\vec{\xi}$.

A square, leading-diagonal sub-matrix of A is written a_{rc} , with r, c now running over a limited set of values. A non-square, off-diagonal sub-matrix of A is written ℓ_{rc} .

We shall make frequent recourse to the concept of an ‘SCC’:

Definition 1 (SCC - Strongly Connected Component) *Let G be a directed graph. A strongly connected component of G is a maximal subgraph, g , in which there exists a directed path from every node in g to every other node in g , by some route.*

We use the latin indices i, j to label SCCs. It is implicit in this definition that a path follows the direction of the arrows (links). The set of SCCs is uniquely determined by the graph, and every node belongs to one and only one SCC. The number of SCCs for a graph with N nodes may be as large as N for a directed

acyclic graph or DAG, or as small as one. In the latter case we simply say that the graph is strongly connected (SC).

The weaker property of being *connected* simply requires that every node in a connected subgraph has some path to every other, *disregarding* the direction of the links (ie, treating them as undirected). We will discuss only connected graphs in this paper.

The Perron-Frobenius theorem addresses the property of *reducibility* in a graph[22, 23]. Each irreducible region can be associated with an SCC, for the following reason. A real $N \times N$ matrix M , with non-negative entries (such as an adjacency matrix), is said to be irreducible if every element, labelled as a row-column pair (r, c) , is greater than zero in some finite power of the matrix, i.e., for every pair (r, c) , there is a p such that $(M^p)_{rc} > 0$. The adjacency matrix A of a strongly connected graph is irreducible, since $(A^p)_{rc}$ is just the number of paths from r to c of length p . Conversely, if a graph is not strongly connected, then it is reducible.

If M is irreducible, then there is a maximum eigenvalue $\rho(M)$, known as its spectral radius, and the corresponding eigenvector is positive everywhere (PDEV). This result is directly relevant to our work: it says that the adjacency matrix A of a strongly connected graph gives a PDEV, which may then be used as an importance function for the nodes. Hence, only non-strongly-connected (reducible) graphs need further consideration to yield a PDEV.

We shall often find it convenient to speak of the flow of ‘weight’ under the operation of adjacency operations.

Definition 2 (Weight) *We use the terminology ‘non-zero weight’ on a strongly connected region C_i , to mean that the components of the right principal eigenvector $\vec{\Psi}$ corresponding to the nodes in C_i have a non-zero value. A ‘weight vector’ is any vector of non-negative values that is used to analyze the eigen-properties of a graph.*

With the above conventions, we find that multiplication of a weight vector $\vec{w}$, by the adjacency matrix A , has the effect of transferring the weight vector’s component values to neighbouring nodes, *against* the direction of the arrows. A^T sends the weights in the direction of the arrows.

Definition 3 (Forward and backward operators) *We call $A \equiv \hat{B}$ the ‘backward operator’. Conversely, we call the ‘forward operator’ $A^T \equiv \hat{F}$.*

Although $A = \hat{B}$ and $A^T = \hat{F}$ are distinct, with distinct dominant eigenvectors, there is a duality to the matrices; switching between them simply reverses all links, and maps sources to sinks and sinks to sources. It suffices to phrase our discussion in terms of one of them. We shall therefore often concentrate our discussion on $\hat{F}$, as this is most easily visualized.

4 Elementary structure of directed graphs

4.1 Metagraph reduction

A simple picture of the coarse-grained structure of a general directed graph is afforded by Perron-Frobenius theory. A graph can be discussed in terms of its strongly connected regions and their links. For the purposes of spectral analysis, each SCC region behaves like a node in a metagraph.

We assume that the whole graph G is *connected*, so that all SCCs are connected by at least one unidirectional link (if one SCC were joined to another by a bi-directional link, they would be the same SCC). We can thus define a coarse-grained *metagraph* for any connected, directed graph, as follows.

Lemma 1 (Metagraph reduction of a directed graph) *Let G be a directed graph. Take each SCC and replace it by a single meta-node; links between SCCs are carried over unchanged from the original graph. The metagraph Γ has one meta-node per SCC, and is a directed acyclic graph (DAG).*

A proof follows easily: assume a cycle of SCCs in the metagraph, then this becomes an single SCC and thus one has a contradiction.

The resulting metagraph emphasizes links between SCCs, while de-emphasizing links within SCCs. Note that there are no directed loops in the metagraph, since these must be internal to SCCs.

We note here that our metagraph differs only slightly from the common notion of 'acyclic component graph' [24], which does not include any information on the number of links between pairs of connected SCCs, only on their direction. We retain all inter-SCC links in the metagraph, as this information will be useful for the analysis techniques we develop below.

4.2 Eigenvalues

Perron-Frobenius theory tells us that the set of eigenvalues of a reducible matrix is simply the multiset union of the eigenvalues of the irreducible subgraphs. This result will be quite useful for our further analysis, and is easily seen. Consider the reducible matrix:

$$A = \begin{pmatrix} \boxed{a_{11}} & \ell_{12} & \dots & \ell_{1n} \\ 0 & \boxed{a_{22}} & \dots & \ell_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \boxed{a_{nn}} \end{pmatrix} \quad (3)$$

Here, a_{ii} is the adjacency sub-matrix of the strongly connected component i . The off-diagonal elements, represent inter-SCC links, if ℓ_{ij} is nonzero for $i \neq j$. The non-zero link blocks do not have to be all on the same side of the diagonal, but non-zero elements on one side on the diagonal will imply zeroes for the reflected counterpart. This translates to the simple statement in the previous section, that all links connecting i to j must have the same direction.

The eigenvalues of this matrix have a very simple structure: the positioning of the zeroes implies the property:

$$\det(A - \lambda I) = \det((a_{11} - \lambda I)(a_{22} - \lambda I) \cdots (a_{nn} - \lambda I)). \quad (4)$$

In other words, the secular equation factorizes, and the eigenvalues of the reducible matrix are simply the multiset-union of the eigenvalues belonging to each of the irreducible components. The connections between the SCCs do not therefore alter the eigenvalue spectrum, however they do perturb the eigenvector components.

4.3 Flows

A directed graph depicts a flow, with its arrows. It is natural to use flow terminology to refer to the relative positions of nodes. The terms upstream and downstream are partial orderings of the set of nodes.

Definition 4 (Upstream and Downstream) *SCC C_2 is said to be downstream of SCC C_1 if there is a path leading from C_1 to C_2 in the metagraph which follows the direction of the links. Upstream is the dual or complement of this case: i.e. if C_2 is downstream of C_1 , then C_1 is upstream of C_2 .*

Upstream and downstream are mutually exclusive relations between SCCs. There exist some SCC pairs that are neither upstream nor downstream of one another; see, for example, the pair C_2 and C_3 in fig. 3b, and the pair C_1 and C_2 in fig. 3c.

A general directed graph can also have isolated nodes, by which we mean, nodes that are not a part of any larger SCC (see fig. 1). Such isolated nodes may be sources, sinks, or neither. We find that both the language used, and the resulting picture, are simplified if we regard such isolated nodes as special cases of an SCC. That is, a subgraph consisting of a single node, and no links (except for possible self-loops) has the property that every node in the subgraph can be reached from every other, within p hops, for $p = 0$. If one ignores the possibility of self-loops, then the 1×1 adjacency matrix for such a subgraph is 0 (which is also its only eigenvalue, and its spectral radius). Unless we explicitly state otherwise, we assume that a graph has no self-loops.

Definition 5 (Zero and non-zero SCCs) *An isolated node that is not part of any larger irreducible region is called a zero-SCC, since it has only a single eigenvalue and that eigenvalue is equal to zero. SCCs with more than one node have a spectral radius which is greater than zero and are thus nonzero SCCs.*

Our inclusion of isolated nodes in the set of SCCs is not a universal convention. Doing so means that every SCC in a connected directed graph has a set of neighbouring SCCs at a distance of one hop. If we do not count the zero SCCs, then there can be in general rather complicated acyclic arrangements of isolated nodes connecting “neighbouring” nonzero SCCs. These connecting

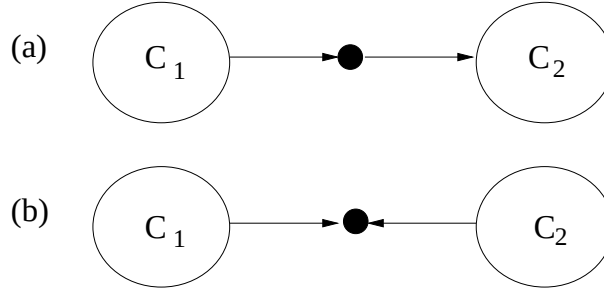


Figure 1: Strongly connected regions can be connected by various combinations of arrows, possibly with other SCCs (black dots) in between. Some connections are isolating, as in (b), whereas others, such as in (a), allow a flow of weight and so are non-isolating. Since SCCs act as single meta-nodes, with a 'gain' (defined in the text) given by their dominant eigenvalue, it is both convenient and unifying to regard individual nodes as being SCCs with zero gain, i.e. as zero-SCCs. In matrix form they form trivially irreducible blocks. Note that the notion of isolating or non-isolating connection applies to pairs of SCCs which are separated by more than one hop.

acyclic subgraphs must still be included in the metagraph, as they affect the nature of the possible weight flows between the SCCs they connect.

At times we will find it convenient to distinguish isolating and non-isolating connections.

Definition 6 (Isolating and non-isolating paths) *Consider two nonzero SCCs C_1 and C_2 , in a connected, directed graph G . If there is a path for weight to flow from C_1 to C_2 (or in the opposite direction), then we call the connection non-isolating. If such a flow is not possible, then the connection is isolating.*

An example of a non-isolating connection is an unbroken unidirectional chain (see fig. 1(a)) from one nonzero SCC to the other. An example of an isolating connection (fig. 1(b)), is out-links from both C_1 and C_2 which terminate at an isolated sink node.

If a pair of SCCs has a non-isolating connection, then they are either upstream or downstream of one another. Conversely, if they are neither upstream nor downstream of one another, then they have an isolating connection.

4.4 Sources and sinks

A crucial difference between directed and undirected graphs is that directed graphs can have *sinks*, or *absorbing nodes*, while undirected graphs cannot.

Definition 7 (Sources and sinks) *A sink node r is any node with out-degree zero ($k_r^{\text{out}} = 0$). The dual of a sink is a source node: a source node has in-degree zero ($k_r^{\text{in}} = 0$).*

The notion of sources and sinks also applies to SCCs in the metagraph. A sink SCC, C_i , has out-degree zero in the meta-graph ($k_i^{\text{out}} = 0$), and a source SCC, C_j , has zero in-degree ($k_j^{\text{in}} = 0$).

Neither sources nor sinks exist in undirected graphs. We observe that a graph with at least one sink must have at least one source. Sinks are a problem for an eigenvector approach to importance ranking. The problem was identified in ref. [7].

The computation of the dominant eigenvector may be likened to finding an asymptotic distribution of weights, distributed over the nodes, under repeated action by the chosen matrix operator. For instance, multiplication of a weight vector $\vec{w}$ by $\hat{F} = A^T$ sends the weight w_r to each of r 's out-going neighbours; i.e, it sends the weight forward.

Repeating this operation many times results in the principal eigenvector of $\hat{F}$, i.e. $\vec{w} \xrightarrow{\hat{F}^\infty} \vec{\Psi}$. Since $\hat{F}$ sends weight with the arrows, one might guess that repeated forward weight propagation sends all the weight into sinks, leaving zeroes everywhere else in $\vec{\Psi}$. Similarly, one might assume that repeated operation of $\hat{B}$ (giving finally its principal eigenvector) places all weight at the sources of the graph.

As we will show in the rest of this paper, these guesses are not strictly correct; but they capture the 'flavor' of the correct answer, in that weight does accumulate in the 'downstream' direction (ie, the sending direction—towards sinks for $\hat{F}$, and towards sources for $\hat{B}$), so that 'upstream' regions (in particular, sources for $\hat{F}$ and sinks for $\hat{B}$) are typically left with zero weight.

Hence, any attempt to assign importance to nodes in a directed graph by finding a dominant eigenvector must deal with the problem of sinks. The PageRank approach uses a 'random surfer' operator, which amounts to adding an all-to-all symmetric graph (with a weight ϵ on the added links) to the given directed graph. Clearly this operator eliminates all sinks and hence avoids the problem[7]. Such a solution is also simple to apply: even though it appears to destroy the property of graph sparseness for the adjacency matrix (which makes practical computations on large graphs tractable), the action of this operator may be computed offline, to the point where it simply requires adding a vector at each iteration [7]. It is also unbiased with respect to the structure of the given graph.

5 Eigenvectors for non-normalized graph operators

To understand the problems surrounding directed graphs, it is helpful to strip the problem down to bare bones and then reconstruct the complexity of general graphs inductively. Let us examine the elementary cases of two- and three-SCC metagraphs.

5.1 Two SCCs

Consider simple case of a graph with two SCCs, of unknown numbers of nodes. We denote these SCCs as C_1 (with adjacency a_{11}) and C_2 (with adjacency matrix a_{22}), and represent the whole graph with a 2×2 super-matrix structure A . That is, we let

$$\hat{B} = A = \begin{pmatrix} a_{11} & \ell_{12} \\ 0 & a_{22} \end{pmatrix}, \quad (5)$$

and

$$\hat{F} = A^T = \begin{pmatrix} a_{11}^T & 0 \\ \ell_{12}^T & a_{22}^T \end{pmatrix}. \quad (6)$$

It is clear from eqn. (5) that C_1 points to C_2 , but not vice versa: that is, C_1 acts as a source in this connected pair of SCCs and C_2 has the role of a sink.

It is intuitive to examine the transpose of the adjacency matrix $\hat{F} = A^T$, using our conventions, since this operator generates flows in the direction of the graph's arrows. We denote a right eigenvector of A^T as

$$\vec{\Psi} = \begin{pmatrix} \vec{\xi}_1 \\ \vec{\xi}_2 \end{pmatrix}, \quad (7)$$

We begin by making a number of observations about this simple construction, framed in the following result:

Theorem 1 (SCC spectral domination) *Let Γ be a reducible, directed meta-graph, consisting only of a source C_1 , with spectral radius ρ_1 , and sink C_2 with spectral radius ρ_2 . If $\vec{\Psi}$ is the right principal eigenvector of A^T , then $\vec{\Psi} > 0$, iff $\rho_1 > \rho_2$.*

We shall say that an SCC C_1 *dominates* an SCC C_2 if $\rho_1 > \rho_2$. Thus, theorem 1 says that $\vec{\Psi}$ is a positive definite eigenvector (PDEV) if and only if the source dominates the downstream remainder of the graph.

Proof 1 *To prove this result, we write:*

$$\begin{pmatrix} a_{11}^T & 0 \\ \ell_{12}^T & a_{22}^T \end{pmatrix} \begin{pmatrix} \vec{\xi}_1 \\ \vec{\xi}_2 \end{pmatrix} = \lambda \begin{pmatrix} \vec{\xi}_1 \\ \vec{\xi}_2 \end{pmatrix}, \quad (8)$$

which can be written as a pair of linear equations:

$$\begin{aligned} a_{11}^T \vec{\xi}_1 &= \lambda \vec{\xi}_1 \\ \ell_{12}^T \vec{\xi}_1 + a_{22}^T \vec{\xi}_2 &= \lambda \vec{\xi}_2. \end{aligned} \quad (9)$$

There are two possibilities: either (i) λ is an eigenvalue of a_{11}^T (and hence of a_{11}), so that $\vec{\xi}_1 = \vec{\psi}_1$ is an eigenvector of the irreducible block C_1 ; or (ii) λ is

not an eigenvalue of a_{11}^T , in which case $\vec{\xi}_1 = 0$, in order to satisfy the first of the equations.

Consider first (ii): in this case, it is impossible to achieve $\vec{\Psi} > 0$ (PDEV). The vanishing of $\vec{\xi}_1$ simplifies the second equation of (9), making it an eigenvalue equation for the isolated region C_2 . Hence, if we are to get $\vec{\xi}_2 \neq 0$, then λ must be an eigenvalue of the region C_2 . Hence, let $\lambda = \lambda_2$. For this case, we see there is a non-negative eigenvector for the coupled pair of regions, by choosing λ_2 to be equal to ρ_2 , the principal eigenvalue of C_2 .

Now consider case (i): $\lambda = \lambda_1$, and $\vec{\xi}_1 > 0$ iff $\lambda_1 = \rho_1$. Now, if ρ_1 is not in the spectrum of C_2 , the second equation of (9) is easily solved:

$$\vec{\xi}_2 = (\rho_1 - a_{22}^T)^{-1} \ell_{21}^T \vec{\xi}_1. \quad (10)$$

Now we must ask, can we also make $\vec{\xi}_2 > 0$, as given in eqn. (10). In the right-hand side of (10), everything is non-negative except, possibly, the first factor, which is now $(\rho_1 - a_{22}^T)^{-1}$. This quantity is strictly positive if and only if $\rho_1 > \rho_2$. Therefore we require $\rho_1 > \rho_2$ for a PDEV.

Finally we cannot discount the special case that $\rho_1 = \rho_2 \equiv \rho$. Then (9) becomes

$$\begin{aligned} a_{11}^T \vec{\xi}_1 &= \rho \vec{\xi}_1 \\ \ell_{21}^T \vec{\xi}_1 + a_{22}^T \vec{\xi}_2 &= \rho \vec{\xi}_2. \end{aligned} \quad (11)$$

We see that the case $\vec{\xi}_1 = 0$ is still possible, with the unperturbed dominant eigenvector on C_2 . That is, the case-(ii) solution is unaffected by this special condition, and a case-(ii) solution does not give a PDEV. If, however, we try to impose a case-(i) solution, we get a contradiction:

$$(\rho - a_{22}^T) \vec{\xi}_2 = \ell_{21}^T \vec{\xi}_1. \quad (12)$$

The problem here is that the left-hand side has zero determinant, while the right-hand side is strictly positive. We can look for some way out of this problem by considering a more general form for a case-(i) solution. Suppose we write:

$$\begin{pmatrix} a_{11}^T & 0 \\ \ell_{12}^T & a_{22}^T \end{pmatrix} \begin{pmatrix} \alpha \vec{\xi}_1 \\ \vec{\xi}_2 \end{pmatrix} = \rho \begin{pmatrix} \alpha \vec{\xi}_1 \\ \vec{\xi}_2 \end{pmatrix}, \quad (13)$$

with α and $\vec{\xi}_2$ to be determined. The upper equation is satisfied for any α . The lower equation is

$$(\rho - a_{22}^T) \vec{\xi}_2 = \alpha \ell_{21}^T \vec{\xi}_1. \quad (14)$$

Now we see that the only way to avoid the contradiction found in (12) is to set $\alpha = 0$ —thus forcing a case-(ii) solution. Here we fail to find a case-(i) solution (with nonzero weight on C_1) for this case—even though we have chosen an eigenvalue belonging to the spectrum of C_1 .

Hence we find that, for the special case $\rho_1 = \rho_2$, there is only one independent eigenvector corresponding to the two degenerate eigenvalues; this eigenvector is a case-(ii) eigenvector, with zero weight in the source SCC, and hence it is not a PDEV. This completes the proof.

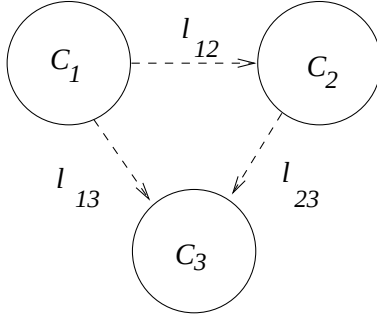


Figure 2: Three strongly connected regions joined by potential directed links in the direction of the arrows.

For importance ranking, we seek stable flows that give non-negative eigenvector distributions for the whole graph. We use the term ‘gain’ as a synonym for the spectral radius. In this language, theorem 1 says that the source must have a greater gain than the sink (i.e. $\rho_1 > \rho_2$), if it is to drive the flow of weight to the sink in such a way that both regions have positive weight everywhere. In this case, the principal eigenvalue for the whole graph is ρ_1 . Even more succinctly: if the source dominates the sink, we get a PDEV.

If, on the other hand, the source has a smaller gain than the sink, we have two choices: it receives zero weight (case (i) above), and the sink C_2 has positive weight from its unperturbed principal eigenvector (here the eigenvalue is ρ_2); or, the coupled system also has a stable flow (eigenvector) for eigenvalue ρ_1 —but it will not be non-negative in C_2 .

5.2 Three or more SCCs

The essence of the directed interaction between strongly connected regions is captured in the pairwise discussion above. Let us now consider what happens when we combine more than two strongly connected regions in a directed graph, focusing only on the action of the forward operator $\hat{F}$. A dual picture can readily be derived for the backward operator $\hat{B}$.

Consider a collection of strongly connected regions $\{C_1, C_2, C_3\}$ represented by diagonal blocks a_{11} , a_{22} and a_{33} , with directed link structures between them ℓ_{12} , ℓ_{13} and ℓ_{23} (see fig. 2). The adjacency matrix has the block structure:

$$A = \begin{pmatrix} a_{11} & \ell_{12} & \ell_{13} \\ 0 & a_{22} & \ell_{23} \\ 0 & 0 & a_{33} \end{pmatrix}, \quad (15)$$

and the forward operator is now:

$$\hat{F} = A^T = \begin{pmatrix} a_{11}^T & 0 & 0 \\ \ell_{12}^T & a_{22}^T & 0 \\ \ell_{13}^T & \ell_{23}^T & a_{33}^T \end{pmatrix}. \quad (16)$$

This case has a number of possible outcomes, depending on the relative sizes of the dominant eigenvalues in the strongly connected regions.

We restrict attention to the dominant eigenvalue and eigenvector for the entire graph: this eigenvector will be non-negative. Hence we restrict attention to the largest eigenvalue, which is the spectral radius of the whole graph: $\rho_{\max} = \max(\rho_1, \rho_2, \rho_3)$. Furthermore, unless otherwise stated, we shall assume that $\rho_{\max}$ is non-degenerate (i.e. that no two SCCs have the same spectral radius, so that $\rho_i \neq \rho_j$ when $i \neq j$). Besides being rare, such cases do not give any special insight in general.

We begin by noting that source nodes do not influence the graph spectrum unless they dominate downstream parts of the graph.

Lemma 2 (Weak source) *Let G be a reducible, directed graph, labelled such that C_1 is a source, and let the spectral radius $\rho_{\max}$ of the graph be non-degenerate; then C_1 has principal eigenvector component $\vec{\xi}_1 = 0$ iff $\rho_{\max}$ is not in the eigenvalue spectrum of a_{11} for C_1 .*

Proof 2 *We consider first the case of three SCCs. Let $\vec{\Psi}^T = (\vec{\xi}_1, \vec{\xi}_2, \vec{\xi}_3)$, and let $\hat{F}$ multiply this. We obtain:*

$$\begin{aligned} a_{11}^T \vec{\xi}_1 &= \rho_{\max} \vec{\xi}_1 \\ \ell_{12}^T \vec{\xi}_1 + a_{22}^T \vec{\xi}_2 &= \rho_{\max} \vec{\xi}_2 \\ \ell_{13}^T \vec{\xi}_1 + \ell_{23}^T \vec{\xi}_2 + a_{33}^T \vec{\xi}_3 &= \rho_{\max} \vec{\xi}_3. \end{aligned} \quad (17)$$

If we assume $\rho_{\max}$ is not in the spectrum of C_1 , then the only solution to the first equation is $\vec{\xi}_1 = 0$.

Now we prove the converse. Assume $\vec{\xi}_1 = 0$. It follows from the second of eqn. (17) that, either $\rho_{\max}$ is in the spectrum of C_2 , or that $\vec{\xi}_2 = 0$. In the former case, by non-degeneracy, $\rho_{\max}$ is not in the spectrum of C_1 . In the latter case, we have the same choice for C_3 : either $\rho_{\max}$ is in its spectrum, or $\vec{\xi}_3 = 0$. We reject $\vec{\xi}_3 = 0$ because otherwise $\vec{\Psi} = 0$. Hence $\rho_{\max}$ is in the spectrum of either C_2 or C_3 , and we have proven the converse.

The generalization to n SCCs follows by induction.

Lemma 3 (Redundant sources) *Let G be a reducible, directed graph, with a strongly connected source component C_i , such that $\vec{\xi}_i = 0$, i.e. the components of the principal eigenvector of the graph $\vec{\Psi}$ are all zero for the source. It follows that all results about the eigenvector components for the remainder of the graph are independent of C_i , and C_i may be deleted from the graph for the purpose of further analysis.*

Proof 3 Referring, once again to eqn. (17), we note that the components of $\vec{\xi}_i = 0$ multiply all of the link structures ℓ_{ij} , rendering them impotent. The rule

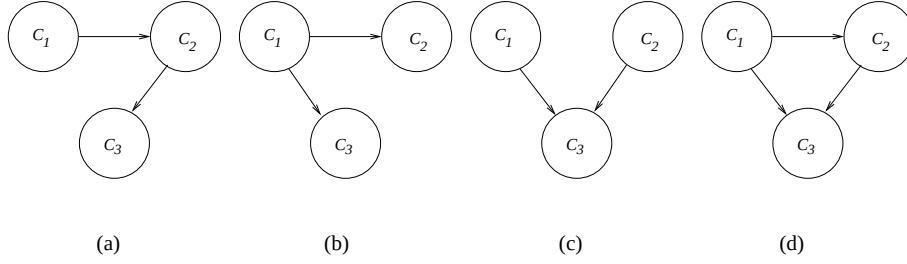


Figure 3: Various special cases of a three-region metagraph. The possibilities for non-negative eigenvectors on the metagraph can be deduced by inspection.

may be extended by induction for greater numbers. Clearly, in the n th iteration, one has simply:

$$\sum_{j=1}^{n-1} \ell_{jn}^T \vec{\xi}_j + a_{nn}^T \vec{\xi}_n = \rho_{\max} \vec{\xi}_n, \quad (18)$$

and the same argument applies.

As usual, the equation for the source SCC C_1 , i.e. the first equation of (17), is the simplest: either C_1 has $\rho_{\max}$ in its spectrum, in which case $\vec{\xi}_1 > 0$, or it does not, and $\vec{\xi}_1 = 0$. In the first case we know, from the previous section, that C_1 can ‘drive’ the other two SCCs, yielding positive weight in the corresponding $\vec{\xi}_j$, as long as $\rho_{\max}$ is not also in one of those regions’ spectra, i.e. as long as $\rho_1 > \rho_j$, where $j \neq 1$.

In the second case (zero weight on C_1), we can ignore C_1 : any analysis involving the remaining two SCCs may be carried out as if the SCC C_1 did not exist.

We see already that the topology of the metagraph plays a pivotal role in determining the structure of the principal eigenvector. Consider next the four specific metagraphs for three SCCs, as shown in fig. 3. We may assume, without loss of generality, that C_1 is a source and C_3 is a sink. The graphs must also be connected. Case (a) is the only possibility that has no link from C_1 to C_3 ; (b) to (d) give the only possibilities when there is such a link present. Hence these four graphs exhaust the possibilities (to within permutations of the SCC labels).

For case (a), Lemmas 2 and 3, along with Theorem 1 (possibly extended to three SCCs), give us the following result: one has a non-negative $\vec{\Psi}$, with eigenvalue $\rho_{\max}$, with $\vec{\Psi}$ having positive weight in the region having $\rho_{\max}$ in its spectrum, and in all regions downstream of this region, and zero weight in all upstream regions. Thus there is positive weight everywhere (giving a PDEV) only if $\rho_{\max}$ is in the spectrum of C_1 .

For case (b), we find a similar conclusion as for case (a): positive weight in the region having $\rho_{\max}$ in its spectrum, and in all regions downstream of

this region, and zero weight in all other regions. Here we must say 'all other regions' rather than 'all downstream regions', since (as already noted) the upstream/downstream relation is not well defined for the pair C_3 and C_2 . For example, suppose C_3 has the eigenvalue $\rho_{\max}$. Then we easily find from (17) that there can be no weight on C_1 —consistent with the fact that C_1 is upstream of C_3 . C_2 is however neither upstream nor downstream from C_3 . However, setting $\vec{\xi}_1 = 0$ in (17) decouples the remaining two equations (an instance of the Redundant Source Lemma), and it is then obvious from the non-degeneracy of $\rho_{\max}$ that $\vec{\xi}_2$ is also zero.

Thus in this case, weight vanishes in a sink SCC (as well as in the source). However, once we apply the Redundant Source Lemma to C_1 , C_2 can equally well be viewed as a special case of a source (since it is isolated, hence has no in-links). Thus we identify a more general condition for weight to vanish on an SCC:

Lemma 4 (Vanishing SCCs) *An arbitrarily chosen SCC C_n has $\vec{\xi}_n = 0$ iff $\rho_{\max}$ is not in the eigenvalue spectrum of C_n and all upstream SCCs C_j satisfy $\vec{\xi}_j = 0$.*

Proof 4 *Assume $\vec{\xi}_n = 0$ and consider eqn. (18); we immediately obtain that, if $\ell_{jn}^T \neq 0$ then $\vec{\xi}_j = 0$. In other words, all SCCs C_j that lie one hop upstream of C_n have zero weight. Now consider the the same argument, starting at each such SCC C_j ; iterating the same reasoning implies that their one hop upstream neighbours also have zero weight. We continue this reasoning to show that there is zero weight in all SCCs upstream of C_n .*

Now we use Lemma 3 and eliminate all the SCCs that are upstream of C_n from the graph. Then C_n becomes a source SCC and it has zero weight, by assumption. Lemma 2 now implies that $\rho_{\max}$ is not in its spectrum.

Now we prove the converse. We assume that all upstream SCCs C_j have $\vec{\xi}_j = 0$, and that $\rho_{\max}$ is not in the spectrum of C_n . Lemma 3 then allows us to eliminate all upstream SCCs, making C_n a source. Lemma 2 then implies that $\vec{\xi}_n = 0$. This completes the proof.

Case (c) exhibits competition between the sources to dominate the sink part of the graph. It has the same general characteristics as the other solutions: for instance, weight only on C_3 when $\rho_{\max}$ arises from C_3 . If the dominant eigenvalue $\rho_{\max}$ comes from one of the source regions, i.e. is an eigenvalue of (say) C_1 , then we get positive weight on C_1 and C_3 . C_2 has zero weight, by Lemma 2. Colloquially, we can say that C_2 cannot support the large gain $\rho_{\max}$ without help in the form of injected weight; hence it must have zero weight. Thus we see a phenomenon which may be called *competition* between source SCCs.

Corollary 1 (Source competition) *Let Γ be a metagraph with multiple source-SCCs, each of which has a distinct and unique spectral radius. We denote this subset of SCCs by $C_s = \{C_\rho\}$, thus indexing the source SCCs with their spectral*

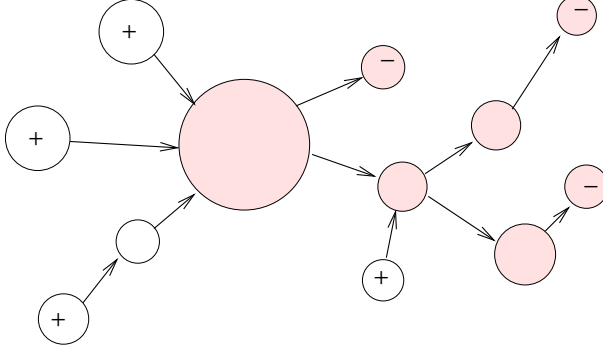


Figure 4: A schematic depiction of our results for many SCCs. The size of each meta-node represents the size of its spectral radius. Only the shaded SCCs have non-zero weight in the right principal eigenvector of $\hat{F}$; unshaded SCCs have zero weight and hence no effect. If the sources (marked with a + symbol) could be ‘pumped up’ to be largest, then the flow could encompass the whole graph.

radii ρ . Then, at most one source SCC can have a principal sub-eigenvector $\vec{\xi}_\rho > 0$ in Ψ , and all others $\vec{\xi}_{\rho' \neq \rho} = 0$ are zero.

The proof follows from Lemma 4.

Suppose we violate, in case (c), the assumption of distinct spectral radii, setting $\rho_{\max}(C_1) = \rho_{\max}(C_2)$, with both of these being greater than the largest eigenvalue of C_3 . From all the above reasoning, it seems that it should be possible to have positive definite weight everywhere in the graph for the dominant eigenvector. This case is of course rare. However we will examine this case (and its generalization to more than two sources) in more detail later, when we examine new ‘sink remedies’.

Finally we come to case (d). Here, we note that the relation upstream/downstream is well defined for all pairs of SCCs in this metagraph. Hence we can use exactly the same words as used for case (a).

Thus, focusing only on non-negative distributions and on the largest eigenvalue $\rho_{\max}$, we see that simply inspecting the metagraph is sufficient to determine what is possible. Furthermore, we have developed an intuition which may be readily generalized to the case of many SCCs. We may summarize this intuition using a more informal language, as follows. In the principal eigenvector of a graph, nonzero weight is found only in the SCC having the largest eigenvalue $\rho_{\max}$ in its spectrum, and all SCCs which are downstream of this SCC (see fig. 4). Upstream SCCs, with lesser spectral radii must have zero weight in this eigenvector. There can be SCCs which are not downstream of this biggest source SCC, and these will all have zero weight in the dominant eigenvector.

We now summarize the results above in a form that suggests applications to importance ranking.

Theorem 2 (Vanishing tributaries for dominant source) *Let G be a graph*

with a non-repeated principal eigenvalue $\rho_{\max}$, and let C_i be the SCC which yields this eigenvalue. The principal eigenvector $\vec{\Psi}$ of G has positive components $\xi_i > 0$ in C_i and in all other SCCs C_j downstream of it; elsewhere the components of $\vec{\Psi}$ are zero.

In other words, only those parts of the graph, which lie downstream of the dominant SCC can be scored greater than zero, using the eigenvector method for a forward operator $\hat{F}$.

Corollary 2 (Single source flow is global) *If the largest-eigenvalue SCC, C_1 , is a source for the metagraph, then the principal eigenvector satisfies $\vec{\Psi} > 0$, iff it is the only source SCC in the graph.*

The proof of this follows readily from the foregoing results.

Finally, if we abandon the assumption that $\rho_{\max}$ is non-degenerate, and ask for the condition for a PDEV, we have suggested (although not proven) the following:

Theorem 3 (Equal-gain sources share dominance) *An eigenvector $\vec{\Psi}$ for graph G is a PDEV, i.e. $\vec{\Psi} > 0$, iff (a) the largest eigenvalue, $\rho_{\max}$, is in the spectrum of a source SCC, and (b) all source SCCs have $\rho_{\max}$ in their spectrum.*

The proof of this result will be provided in section 7.2.2.

What this theorem implies is that, when a graph's principal eigenvector is driven by more than one dominant source, only those sources that have spectral radii equal to the maximum spectral radius of the graph can have non-zero components in the principal eigenvector $\vec{\Psi}$. This suggests a tantalizing possibility: namely that if one could make all sources large and equal, then a PDEV importance score could be achieved without otherwise modifying the topology of the metagraph.

Thus far we have shown the “only if” part here, i.e. that the principal eigenvector is not a PDEV when (b) does not hold. We defer a full proof that (a) and (b) are sufficient until section 7.2.2.

These conclusions give an extremely stringent condition for a PDEV, which is rarely satisfied in a naturally occurring graph: all sources must have the dominant eigenvalue of the entire graph in their spectrum.

As a postscript to this section, we offer a topologically-motivated proof that the set of eigenvalues of a directed graph is the multiset union of the eigenvalue sets for the SCCs (Perron-Frobenius).

Proof 5 ((Alternative) Spectrum of reducible G) *Consider an eigenvalue Λ which is not in the spectrum of any SCC. Assume Λ is in the spectrum of the full graph, with corresponding eigenvector E . Then we know from Lemma 2 that all sources have zero weight in E . We then use Lemma 3 to remove all these source SCCs from the graph. This gives rise to new sources, which also (by our assumption) do not have Λ in their spectrum; hence we can remove them. We repeat this process until the only SCCs remaining are those that are sinks in the*

original graph. These SCCs also have zero weight in E , by Lemma 4. Hence $E = 0$, and so Λ is not in the spectrum of the graph.

Thus we find that the spectrum of the graph cannot include eigenvalues not taken from the union of the sets of eigenvalues of the SCCs. However the graph has N eigenvalues (for N nodes), as does the union of the sets of eigenvalues for the SCCs. Hence the two sets are the same.

Note that the topological argument is compactly stated because we included isolated nodes as a type of SCC. Zero SCCs provide one zero eigenvalue each to the spectrum of the full graph. Hence zero can be a highly degenerate eigenvalue for a directed graph.

6 Eigenvectors for normalized graph operators

Normalization is the adjustment of link-weights in a graph's adjacency matrix representation, such that the rows or columns sum to one. Under some circumstances, this can have the effect that, under multiplication by the matrix, the total amount of node-weight is conserved. When this occurs, we say that the operator is *weight conserving*. Normalized matrices are sometimes called stochastic matrices[23], since they have a probability interpretation.

Our discussion so far has been for general directed graphs. The commonest case assumes a weight of 1 on every link. This case gives a spectral radius which is larger than one, due to the fact that each node, under multiplication by (for example) $\hat{F}$, sends out the weight it has, multiplied by its out-degree (a number typically greater than one). Weight conserving graph operators include the very well known example of PageRank, and are thus also of interest.

The general behaviour of eigensystems for normalized graph operators is rather different from that for the non-normalized case. The main challenge in addressing the normalized cases is a notational one, since there are several distinct ways in which one can apply normalization constraints, and each depends on whether one moves with or against the arrows. In the literature, most authors have focused on cases of particular interest to them, thus avoiding a general notation. Since we have use for all of the combinations in our discussion, we must address this issue first.

Column normalization of an (untransposed) adjacency matrix means dividing each column by the in-degree of the node concerned, and row normalization means dividing the rows by the out-degree. Other possibilities exist [14], such as dividing by various mixtures of the in and out degrees, but this would not yield the same row or column sums in A , for all nodes, and thus the property of a unit spectral radius would be lost. We shall not consider further possibilities here.

Let the normalization operator $\hat{N}_R(M)$ denote the unbiased operation of normalizing the rows of matrix M so that they sum to unity; i.e. by dividing

each element in the row by the sum of all the elements in the row. That is,

$$(\hat{N}_R(A))_{ij} = \frac{A_{ij}}{\sum_j A_{ij}} = \frac{A_{ij}}{k_i^{\text{out}}}, \quad (19)$$

where these definitions hold for $k_i^{\text{out}} \neq 0$. For $k_i^{\text{out}} = 0$, we take the normalization of a row of zeroes to be a row of zeros.

Similarly, let $\hat{N}_C(M)$ denote the operation of column-normalization of the matrix; then

$$(\hat{N}_C(A))_{ij} = \frac{A_{ij}}{\sum_i A_{ij}} = \frac{A_{ij}}{k_j^{\text{in}}}, \quad (20)$$

with the same caveat for $k_j^{\text{in}} = 0$. These normalizations clearly give

$$\sum_i (\hat{N}_C(A))_{ij} = \sum_j (\hat{N}_R(A))_{ij} = 1, \quad (21)$$

whenever there is a nonzero entry in the respective column or row. When A is an adjacency matrix, a column of zeroes represents a source ($k^{\text{in}} = 0$) and a row of zeroes, a sink.

Because of their normalization to 1, the entries of a column-normalized matrix can be interpreted as probabilities for the transition $j \rightarrow i$, ie, against the arrows:

$$P(i \leftarrow j) = P(i|j) = (\hat{N}_C(A))_{ij}. \quad (22)$$

The probability interpretation does not work for a row-normalized matrix, which in general is not weight conserving (recall our definition of a weight conserving operator). However (see below) the transpose of a row-normalized matrix is column-normalized. A column-normalized matrix is weight conserving as long as there are no zero columns.

Lemma 5 (Normalized graph operators have unit spectral radius) *A non-negative matrix that has been either row or column normalized has a spectral radius or principal eigenvalue of 1; i.e., there is a $\vec{\psi}$ and $\vec{\psi}'$ such that*

$$\begin{aligned} \hat{N}_R(M)\vec{\psi} &= \vec{\psi} \\ \hat{N}_C(M)\vec{\psi}' &= \vec{\psi}'. \end{aligned} \quad (23)$$

For the proofs, readers are referred to standard texts[22, 23].

An important corollary is the following:

Corollary 3 (Normalized isolated SCCs have unit radius) *Suppose a graph G consists of a collection of isolated SCCs. The normalization of an adjacency operator for G gives each SCC a spectral radius of one.*

There are two ways of normalizing a given positive matrix. Furthermore, one can apply the same kinds of normalization to the transpose of the given matrix; and the two operations of normalization and transpose do not commute. There are valid reasons for considering each of these operations, depending on how we wish to interpret graphs. Based on these observations, we choose the following notation:

$$\mathcal{A}_{\text{out}} = \hat{N}_R(A) = \hat{f}^T \quad (24)$$

$$\mathcal{A}_{\text{in}} = \hat{N}_C(A) = \hat{b} \quad (25)$$

$$\mathcal{A}_{\text{out}}^T = (\hat{N}_R(A))^T = \hat{N}_C(A^T) = \hat{f} \quad (26)$$

$$\mathcal{A}_{\text{in}}^T = (\hat{N}_C(A))^T = \hat{N}_R(A^T) = \hat{b}^T \quad (27)$$

to represent the distinct cases of interest. The labels ‘in’ and ‘out’ on the left hand side indicate which degree of the nodes has been averaged over.

We use small letters for normalized operators, on the right hand side. Thus the symbol $\hat{b}$ conveys the meaning of a normalized backward operator, whose entries may be interpreted as probabilities because of column normalization. Analogously, $\hat{f}$ signifies a normalized forward operator. The two operators $\hat{f}^T$ and $\hat{b}^T$ are row normalized, hence not weight preserving. We will be most interested in the column-normalized (and therefore weight preserving) matrices, of these four; hence they get simple names.

6.1 Special solutions in terms of node degree

In the undirected case, we saw that the local node degree distribution is the solution of the eigenvector problem for normalized adjacency matrices. Because of the loss of symmetry of the adjacency matrix, this property is not preserved in the directed case, except in two special situations: (i) graphs for which $k_r^{\text{in}} = k_r^{\text{out}}$ for all nodes r (see figs. 5 (a), (b) and (c)); and (ii) graphs for which $k_r^{\text{in}} = k_c^{\text{in}}$ for all r, c or $k_r^{\text{out}} = k_c^{\text{out}}$ for all r, c , i.e. k^{in} -regular graphs, or k^{out} -regular graphs (figs. 5 (b) and (c)). Next we present proofs of these two results.

Theorem 4 (In-out symmetry and $\hat{b}$) *Let $\hat{b} = \mathcal{A}_{\text{in}}$ be the column-normalized adjacency operator of a connected graph G . The (in- or out-)degree vector of the graph is a PDEV solution (with eigenvalue 1) iff $\vec{k}^{\text{in}} = \vec{k}^{\text{out}} = \vec{k}$.*

Proof 6 *We define $\vec{k}^{\text{in}}$ and $\vec{k}^{\text{out}}$ to be the vector of in-degrees and out-degrees, respectively. Now we consider $\mathcal{A}_{\text{in}}\vec{k}^{\text{in}}$:*

$$\begin{aligned} (\mathcal{A}_{\text{in}}\vec{k}^{\text{in}})_r &= \sum_c (\mathcal{A}_{\text{in}})_{rc} k_c^{\text{in}} \\ &= \sum_{c; k_c^{\text{in}} \neq 0} \frac{A_{rc}}{k_c^{\text{in}}} k_c^{\text{in}} \\ &= \sum_{c; k_c^{\text{in}} \neq 0} A_{rc} . \end{aligned} \quad (28)$$

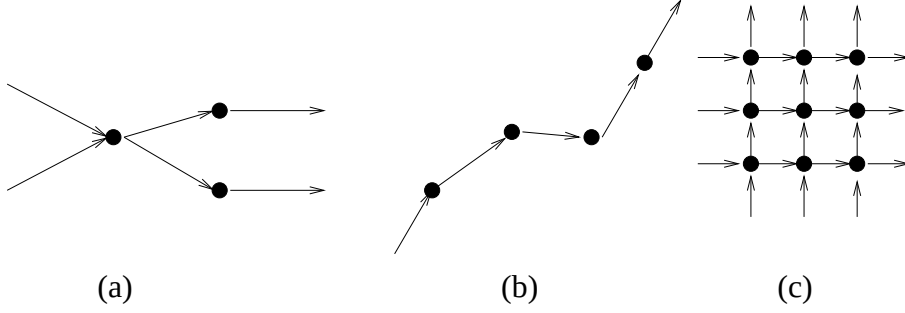


Figure 5: Some of the special solutions for normalized directed flows.

Now we assume that $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$. This implies that any zeroes in one of these vectors shows up for the same node in the other—giving a disconnected node and hence a disconnected graph. Therefore, for $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$, both vectors are positive definite. This lets us drop the condition on the sum in (28), giving

$$\begin{aligned}
 (\mathcal{A}_{\text{in}} \vec{k}^{\text{in}})_r &= \sum_c A_{rc} \\
 &= k_r^{\text{out}} \\
 &= k_r^{\text{in}}.
 \end{aligned} \tag{29}$$

Therefore, for the special case that $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$ (see Fig. 5), we find that the node degree vector is a right eigenvector of $\mathcal{A}_{\text{in}} = \hat{b}$, with eigenvalue one.

Now we assume that $\vec{k}^{\text{in}}$ is a PDEV of $\mathcal{A}_{\text{in}}$ with eigenvalue 1. We can begin with (28) which is identically true. Now we can drop the condition on the sum, due to our assumption that $\vec{k}^{\text{in}}$ is a PDEV. This gives again

$$\mathcal{A}_{\text{in}} \vec{k}^{\text{in}} = \vec{k}^{\text{out}} ; \tag{30}$$

while our assumption gives

$$\mathcal{A}_{\text{in}} \vec{k}^{\text{in}} = \vec{k}^{\text{in}} . \tag{31}$$

Thus we get $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$, which completes the proof in the other direction.

This is an interesting result, for the following reason: we see that the condition $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$, along with connectivity of the graph, means that each of these vectors is positive definite. Hence the principal eigenvector of $\hat{b}$, for any connected graph satisfying $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$, is positive definite. In a later section, we show that this can never occur if the graph is reducible. Hence, the present result suggests that all such graphs (ie, those with $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$) are strongly connected. We shall prove that this is the case in section 6.4.

A second special solution is found for a left eigenvector of $\hat{b}$.

Theorem 5 (Constant in-degree for $\hat{b}$) Let $\hat{b} = \mathcal{A}_{\text{in}}$ be the column-normalized adjacency operator of a connected graph G . The in-degree vector of the graph is a positive definite left eigenvector of $\hat{b}$ iff $\vec{k}^{\text{in}} = k^{\text{in}} \vec{J}$, where $\vec{J}^T = (1, 1, 1, \dots)$.

Proof 7 To prove this we seek a left eigenvector for $\mathcal{A}_{\text{in}} = \hat{b}$. Consider the product $(\vec{k}^{\text{in}})^T \mathcal{A}_{\text{in}}$:

$$\begin{aligned} ((\vec{k}^{\text{in}})^T \mathcal{A}_{\text{in}})_c &= \sum_r k_r^{\text{in}} (\mathcal{A}_{\text{in}})_{rc} \\ &= \sum_r k_r^{\text{in}} \frac{A_{rc}}{k_c^{\text{in}}} \quad (k_c^{\text{in}} \neq 0). \end{aligned} \quad (32)$$

Here we see that if all nodes have the same in-degree, so that $\vec{k}^{\text{in}} = k^{\text{in}}(1, 1, \dots, 1)^T \equiv k^{\text{in}} \vec{J}$ (and thus all $k_c^{\text{in}} \neq 0$), we get the simple result that

$$\begin{aligned} ((\vec{k}^{\text{in}})^T \mathcal{A}_{\text{in}})_c &= \sum_r A_{rc} \\ &= k_c^{\text{in}}, \end{aligned} \quad (33)$$

or

$$(\vec{k}^{\text{in}})^T \mathcal{A}_{\text{in}} = (\vec{k}^{\text{in}})^T. \quad (34)$$

Thus, for the special case that $\vec{k}^{\text{in}} = k^{\text{in}} \vec{J}$ (see figs. 5 (b) and (c)), we find that $(\vec{k}^{\text{in}})^T$ is a left eigenvector of $\mathcal{A}_{\text{in}} = \hat{b}$ with eigenvalue one.

We note that these two theorems coincide in the case of a regular toroidal (periodic) lattice, such as in fig. 5c. Similar conclusions are found for row-normalization in the transposed picture. For instance, the analogue of (29) is

$$\mathcal{A}_{\text{out}}^T \vec{k}^{\text{out}} = \vec{k}^{\text{in}}, \quad (35)$$

implying that $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$ is a right eigenvector of $\hat{f} = \mathcal{A}_{\text{out}}^T$ with eigenvalue 1 for in-out symmetrical graphs. Also, analogously to (34), for the case $\vec{k}^{\text{out}} = k^{\text{out}} \vec{J}$, we get that

$$(\vec{k}^{\text{out}})^T \mathcal{A}_{\text{out}}^T = (\vec{k}^{\text{out}})^T, \quad (36)$$

so that $(\vec{k}^{\text{out}})^T$ is a left eigenvector of $\hat{f} = \mathcal{A}_{\text{out}}^T$ with unit eigenvalue, for out-regular lattices.

Lemma 6 (Duality) A directed graph G and its associated matrices satisfy a dual symmetry, $*$ such that:

$$* \equiv (L \leftrightarrow R), (\text{in} \leftrightarrow \text{out}), (\leftrightarrow \text{T}) \quad (37)$$

and

$$(\mathcal{A}_{\text{in}}) \vec{\psi} = \lambda \vec{\psi} \quad \leftrightarrow \quad (\mathcal{A}_{\text{in}})^* \vec{\psi}^* = \lambda \vec{\psi}^* \quad \leftrightarrow \quad \vec{\psi}^T \mathcal{A}_{\text{out}}^T = \lambda \vec{\psi}^T. \quad (38)$$

Note that, if we remove the normalization constraints from *any* row or column of the matrix, these special solutions cease to apply. This has particular consequences for solving normalized reducible graphs that contain several nonzero strongly connected regions, as we will see below.

6.2 No general solutions in terms of node degree

In general, for normalized flows on directed graphs, there is no solution directly in terms of the local node degrees. To see this, we may seek a solution in terms of linear combinations of the degree vectors $R\vec{k}$ in order to solve the constraints, where R is an undetermined matrix. Specifically, let us try $R\vec{k}$ as a right eigenvector of $\mathcal{A}_{\text{in}}$:

$$\begin{aligned} \mathcal{A}_{\text{in}} R\vec{k} &= \lambda R\vec{k} \\ (\mathcal{A}_{\text{in}} R\vec{k})_r &= \lambda (R\vec{k})_r \\ \sum_c (\mathcal{A}_{\text{in}})_{rc} \sum_d R_{cd} k_d &= \lambda \sum_d R_{rd} k_d \\ \sum_d \left(\sum_c (\mathcal{A}_{\text{in}})_{rc} R_{cd} \right) k_d &= \sum_d (\lambda R_{rd}) k_d. \end{aligned} \quad (39)$$

This is of the form $X\vec{k} = Y\vec{k}$. We have already found special solutions, where R is the unit matrix and $\vec{k}$ takes on special values. If we now ask for a solution for *any* $\vec{k}$, we must have $X = Y$; that is,

$$\mathcal{A}_{\text{in}} R = \lambda R. \quad (40)$$

However we note that, for this equation to be satisfied, the matrix R must have columns that are already (degenerate) eigenvectors of $\mathcal{A}_{\text{in}}$. Hence this approach requires a solution in order to find one, and so postulating the given dependence on $\vec{k}$ offers no new information.

We can however find the dominant left eigenvector for each of $\hat{b} = \mathcal{A}_{\text{in}}$ and $\hat{f} = \mathcal{A}_{\text{out}}^T$, from the results of the previous section. Equation (34), along with the condition for which it is valid, tells us that

$$\vec{J}^T \mathcal{A}_{\text{in}} = \vec{J}^T, \quad (41)$$

which shows that $\vec{J}^T = (1, 1, 1, \dots)$ is a left eigenvector of $\mathcal{A}_{\text{in}}$. Similarly, from eqn. (36) we get

$$\vec{J}^T \mathcal{A}_{\text{out}}^T = \vec{J}^T. \quad (42)$$

Thus we find a left eigenvector for general $\mathcal{A}_{\text{in}} = \hat{b}$ and $\mathcal{A}_{\text{out}}^T = \hat{f}$. However this eigenvector places the same weight at every node. This left eigenvector can therefore play no role in ranking the nodes for their importance. That responsibility must fall to the right eigenvector.

We note here that transposing the above two results gives us dominant right eigenvectors for $\mathcal{A}_{\text{in}}^T$ and $\mathcal{A}_{\text{out}}$. We can view these two matrices as giving a non-weight-conserving diffusion process, where weights are moved by application of the matrices. For example, the process represented by $\mathcal{A}_{\text{out}}$ replaces the weight at each node i with the average of the weights over out-neighbours of i . Similarly, the process $\mathcal{A}_{\text{in}}^T$ replaces the weight at each node with the average of the weights over in-neighbours. We see here that each of these processes converges to a uniform distribution.

6.3 Reducible normalized graph operators

The approach to studying a normalized graph operator is no different, in principle, from that of the non-normalized case, but there is an additional subtlety imposed by the presence of links between irreducible components. As long as strongly connected regions are unconnected, the normalization (by either row or column) causes each SC region to have a spectral radius of 1. Let us now consider a graph composed of two SCCs. Then it is easy to prove the following result:

Theorem 6 ($\hat{f}$ never has a PDEV on a reducible graph) *Let γ be a normalized, directed metagraph and let S be the set of source SCCs for the graph. All SCCs in S have spectral radius less than 1, hence less than the spectral radius of the whole graph. Thus $\xi_i = 0$ for each $C_i \in S$, and hence there is no PDEV under $\hat{f}$.*

Theorem 7 ($\hat{b}^T$ always has a PDEV on a reducible graph) *Let γ be a normalized, directed metagraph and let S be the set of source SCCs for the graph. All SCCs in S have the maximal spectral radius 1, $\xi_i > 0$ for each $C_i \in S$, and hence there is a PDEV under $\hat{b}^T$.*

These two results may be proven together.

Proof 8 *Consider two disconnected SCCs, so that the whole graph's adjacency matrix takes the form:*

$$A = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix}. \quad (43)$$

Then normalization by method x (row or column) gives

$$N_x(A) = \begin{pmatrix} N_1 & 0 \\ 0 & N_2 \end{pmatrix} \quad (44)$$

$$= \begin{pmatrix} N_x(a_{11}) & 0 \\ 0 & N_x(a_{22}) \end{pmatrix}. \quad (45)$$

By Corollary 3, both N_1 and N_2 have spectral radius 1. Note that the x in eqn. (45) indicates that the method of normalization, either by row or column, is

unspecified. If we now add any upper-right non-diagonal element to this form, making the graph into a connected, reducible structure, then the total graph will not be correctly normalized unless we adjust the row or column sums accordingly. Thus, if we add $\ell_{12} \geq 0$ with at least one positive element:

$$A = \begin{pmatrix} a_{11} & \ell_{12} \\ 0 & a_{22} \end{pmatrix} \quad (46)$$

then we get:

$$N_x(A) = \begin{pmatrix} N'_1 & \ell'_{12} \\ 0 & N'_2 \end{pmatrix}. \quad (47)$$

Since either row or column normalization includes elements of ℓ_{12} , normalization will give some elements in the primed diagonal blocks which are smaller than in the unprimed (unconnected) case. This affects the spectral radius ρ of these blocks. For example, in a column-normalized graph operator $\hat{b}$ ($= \hat{N}_C(A)$), $\rho(N'_2) < \rho(N_2) = 1$, even if only a single link is added from region 1 to region 2. The spectral radius of the sink SCC must be reduced, because the presence of ℓ_{12} (along with column normalization) makes N'_2 non-weight-conserving: it loses weight to the source under operation by $\hat{N}_C(A)$. Similarly, in a row-normalized graph operator $\hat{f}^T$ ($= \hat{N}_R(A)$), $\rho(N'_1) < \rho(N_1) = 1$.

There are two cases:

- For $\hat{f} = (\hat{N}_R(A))^T$: $N'_2 = N_2$ is unchanged by the row normalization, while $\rho(N'_1) < \rho(N_1) = 1$. Thus for this operator one has $\rho_1 < \rho_2 = 1$, regardless of the structures of the components prior to normalization. From section 5.1, plus the fact that C_1 is a source in (46), we know that this situation guarantees that there can never exist a positive definite eigenvector (PDEV) for the combined regions.
- For $\hat{b}^T = (\hat{N}_C(A))^T$: column normalization leads to $\rho(N'_2) < \rho(N_2) = 1$ and $\rho(N'_1) = \rho(N_1) = 1$. Thus in this case $1 = \rho_1 > \rho_2$, regardless of the structures of the modules prior to normalization. Also, $\hat{b}^T$ is a forward operator. Hence we know from section 5.1 that the source can “drive” the sink, so that there always exists a PDEV for the combined regions.

This argument generalizes trivially to many SCCs.

In colloquial terms, for $\hat{f}$, region 1 cannot dominate region 2, hence the only stable solution has zero weight in region 1. In the case of $\hat{b}^T$ (also a forward-acting operator), the opposite is true and there is always a PDEV. Note however that this eigenvector is not useful: from the transpose of eqn. (41), we get that $\vec{J}$ (all ones) is the dominant right eigenvector for $\mathcal{A}_{\text{in}}^T = \hat{b}^T$.

Similar considerations (using duality) tell us that the dominant right eigenvector for $\hat{b}$ has weight only in the source SCCs, while the dominant right eigenvector for $\hat{f}^T$ is uniform.

Thus we see that finding a useful PDEV for normalized representations of reducible graphs is even more difficult than for the reducible, non-normalized case. In the latter case, we see that PDEVs occur *almost never*. For normalized, reducible graph operators that are weight conserving ($\hat{f}$ and $\hat{b}$), PDEVs *never* occur. For normalized, reducible operators that are not weight conserving ($\hat{f}^T$ and $\hat{b}^T$), there is *always* a dominant PDEV—but it is *useless*, as it is uniform.

6.4 A special case for general graphs

We conclude this section with a final theorem that is of general interest. This is actually a result of theorem 4. This result is not intuitively obvious, but makes a strong statement about the reducibility of graphs with in-out symmetry.

Theorem 8 (In-out symmetric (IOS) graphs are strongly connected)

Any connected graph, G , with $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$ is strongly connected, i.e. its adjacency matrix is irreducible.

We provide two proofs of this result. The first proof makes use of our earlier result for a normalized graph representation, and it is particularly simple.

Proof 9 (IOS are SC I) *Suppose G is not strongly connected, i.e. assume it is reducible, and suppose further that $\vec{\Psi}$ is the principal right eigenvector of $\hat{b}$; then $\vec{\Psi} \not\propto 0$, from the dual of theorem 6. But $\vec{k}^{\text{in}} = \vec{k}^{\text{out}} = \vec{k} > 0$, and this is a solution for $\vec{\Psi}$ from theorem 4. Hence we have a contradiction, and thus G is irreducible or strongly connected.*

The second proof is interesting because it does not refer to normalization or even to eigenvectors: it is purely topological.

Proof 10 (IOS are SC II) *Again we assume the converse, i.e. that $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$ but G is reducible. This implies that $G = \langle \mathcal{N}, \mathcal{E} \rangle$ has at least one sink SCC C_s . Now, consider any subset of nodes $n \subset \mathcal{N}$. Then, by virtue of $\vec{k}^{\text{in}} = \vec{k}^{\text{out}}$, we have the local condition:*

$$\sum_{r \in n} k_r^{\text{out}} = \sum_{c \in n} k_c^{\text{in}} \quad (48)$$

But now suppose that this subset of nodes is the set of nodes constituting C_s . Then

$$\sum_{r \in n} k_r^{\text{out}} < \sum_{c \in n} k_c^{\text{in}}. \quad (49)$$

The inequality holds by the following reasoning. Each sum in (49) may be decomposed into a sum using only links which are purely internal to the SCC C_s (ie, start and end in C_s), plus a sum over terms coming from other 'non-internal' links. The two internal sums (ie, one for k^{in} and one for k^{out}) are equal, since each sum merely counts the internal links. The sum over non-internal outgoing links is zero for a sink SCC, while the sum over non-internal incoming links is at least one. Hence we get the strict inequality (49), which gives us a contradiction.

7 Other methods for obtaining a PDEV

Our principal aim in this work is to find a natural measure of importance for the nodes in a directed graph, analogous to the very powerful concept of eigenvector centrality for non-directed graphs. We find that there is no direct analogue of the concept of eigenvector centrality for undirected graphs, but rather a much richer structure of concepts and definitions.

Adding to the complexity of the many eigenvectors that one might use is the fact that, in general, for a directed graph (normalized or not), there is no single eigenvector which has positive weight in all SCCs. Hence we find no simple, clear candidate for a global, positive importance function for a directed graph.

In this section, we ask, in the light of the foregoing results, what approaches might be used to build such an importance function. We propose two possible strategies: (i) use vectors derived from higher powers of the adjacency matrices; (ii) modify the graph. We discuss these strategies in turn, and note that each method has different strengths and leads to different scores, suggesting the possibility of combining a mixture of methods to optimize the ranking of nodes.

7.1 Vectors from higher powers of A and A^T

We have looked at many modifications of the basic adjacency matrix, namely $\hat{F}$, $\hat{B}$, and the four normalized versions given in eqns. (24)–(27). We have seen that none of these can give a useful dominant eigenvector in general. Another possibility is to generalize an idea of Kleinberg [12] and consider higher powers of the adjacency matrix. Kleinberg himself looked at the operators $A^{\text{auth}} \equiv A^T A = \hat{F} \hat{B}$, and $A^{\text{hub}} \equiv A A^T = \hat{B} \hat{F}$ (see fig. 6). These quadratic operators do not work like single link-hops, but rather as double hops along connected pairs of arrows. These double hops give an effective single hop (dashed arrows in fig. 6) with a weight obtained from summing over all possible such double hops. For example, the effective link in fig. 6(a) transfers a ‘pointed-to’ or ‘authority’ weight, via one or more intermediate ‘pointing’ or ‘hub’ nodes. The hub-operator effective link (fig. 6(b)) is simply the dual of this. Kleinberg’s method thus respects the intrinsic duality of a directed graph by giving to each node two importance scores (authority and hub), taken from the dominant eigenvector of A^{auth} and A^{hub} , respectively. This method (also known as HITS) has received a lot of attention in the literature, due to its simplicity and elegance of construction [25, 26, 27].

The Kleinberg operators have the beneficial property of being symmetric; hence the effective graphs which they induce are undirected, and have no sources or sinks. Unfortunately, however, the undirected effective graphs do not necessarily give a consistent global picture, since in some cases the effective graph can be disconnected, even when the original directed graph is connected. For example, the simple structure of fig. 1(a) will give rise to effective hub and authority graphs in which the components C_1 and C_2 are disconnected.

The Kleinberg operators measure two-hop correlations. One could consider higher-order correlations, using operators such as $A A^T A$, $A^T A A^T$, etc. Alter-

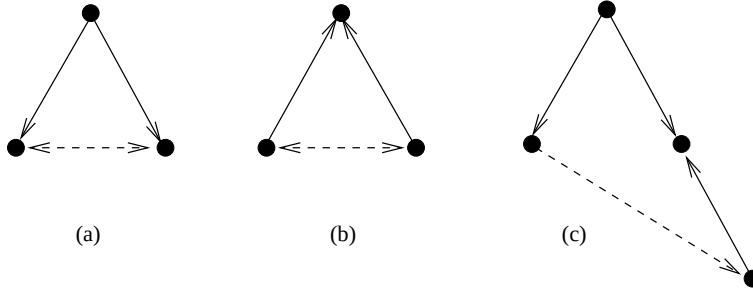


Figure 6: The quadratic Kleinberg operators project structures with adjacent pairs of arrows. (a) shows a symmetric effective link between the two lower nodes, using the authority operator $\hat{F}\hat{B}$, which induces a backwards hop followed by a forward hop. (b) shows a symmetric effective link induced by the dual (hub) operator $\hat{B}\hat{F}$. The induced link is shown by the dotted line and is undirected. Sub-graph (c) shows the action of a third order, asymmetric operator $AA^T A = \hat{B}\hat{F}\hat{B}$, which leads to a directed induced graph.

nating symmetric sequences are simply powers of A^{auth} or A^{hub} ; so they offer nothing new. Other symmetric sequences, such as $A^T A^T A A$, offer a new view of the graph, just as the Kleinberg operators do. Asymmetric sequences [fig. 6(c)] give rise to effective graphs which are still directed; hence these operators may have the same problems as A and A^T in generating global importance functions.

Finally, we note that all higher-order operators have the possibility of generating disconnected effective graphs from an original, connected directed graph. This follows from the simple fact that connectedness is a condition on single hops, while connectedness of higher-order, effective graphs is a more stringent condition on multiple hops.

7.2 Augmenting graphs

As a final strategy for generating global importance functions for general directed graphs, we consider augmenting the given graph, so as to make it more conducive to global ranking by PDEV. The outstanding and well-known example of this approach is the PageRank algorithm[7], used for generating importance weights for the Web. The PageRank algorithm solves the challenge of reducibility by adding bidirectional links between every pair of nodes, thus rendering the entire graph strongly connected. This approach has received a lot of attention in the literature [28, 29].

The Web (see below) consists of many SCCs, and so is not particularly conducive to global ranking. PageRank uses the $\hat{f}$ operator, ie, normalized forward flow. To avoid the above mentioned problems with $\hat{f}$ on reducible graphs, a weighted sum of the $\hat{f}$ operator and a “random surfer” operator $\hat{r}$ is used. The latter operator is a complete (and normalized) graph, ie, it connects all nodes to all nodes with equal probability. The resulting graph is of course

a single, strongly connected component. The links from $\hat{r}$ are given a weight ϵ ; hence for small ϵ , one has, perhaps, the topological properties of a strongly connected graph, without perturbing too strongly the nature of the given graph. The implications of this warrant further study, however: here, we have shown that in general *no* eigenvector of the $\hat{f}$ operator resembles one with positive weight everywhere.

7.2.1 Link reversal

We have focused here on understanding a general, directed graph in terms of its set of SCCs. We reduce these to meta-nodes, connected in a directed, acyclic metagraph. This picture immediately suggests another way of perturbing the given graph so as to make it strongly connected. That is, one can add ϵ times the reverse of every arrow in the metagraph. It is these arcs which join the SCCs; and they are never two-way.

Adding a small weight times the reverse of each of these inter-SCC arcs renders the entire graph SCC. This idea is illustrated in fig. 7(c), where one can simply imagine the nodes in this figure to be meta-nodes, ie, SCCs. Furthermore, it may be that such a modification is a weaker perturbation of the original graph than the random surfer operator: far fewer links are added for large graphs.

There is an additional computational cost for this method however. One must first find the SCCs and map out the metagraph; this latter step includes finding all the inter-SCC links of the metagraph. We will address these computational costs in Section 7.4.

7.2.2 Pumping sources

Finally, we mention another idea inspired by the metagraph picture. One can identify the source SCCs in the metagraph. One then artificially increases the spectral radius, or ‘gain’ of each of these SCCs, until two conditions are satisfied: (i) they are all the same, and (ii) they are dominant with respect to any other SCC. We will call the process of increasing the gain of a source SCC *pumping* the source.

Fig. 8(b) offers an illustration of this idea, with the self-loops representing a mechanism for increasing the gains of 1 and 2. The illustration is only schematic if the graph is truly a metagraph; as we will see, the picture is exact for the case that the both 1 and 2 are zero SCCs.

It remains to specify, in detail, how to pump the sources. A simple way to increase the gain of an SCC, without grossly distorting its topology, is to multiply its diagonal block—all its internal arcs—by a constant $\alpha > 1$. Knowing the spectral radius of all the source SCCs when they have link weight equal to one—and also knowing the largest eigenvalue of the whole graph—it is then a simple matter to find a set of constants that give the desired conditions for the source SCCs. The choice of the value α , for each source SCC, is constrained by conditions (i) and (ii) above. This means that there is one free parameter for all

the source SCCs, since, if there are s source SCCs, (i) gives $(s - 1)$ constraints, while (ii) is only an inequality. One possible choice is to just barely satisfy the inequality; but larger gains may be useful also for some purposes.

The previous paragraph suffices for any source SCC except the special case of a zero SCC. Here there is normally no internal arc to pump. Hence, in the case of a source SCC which is a zero SCC, one must add a self-loop—ie, a diagonal entry for that node in the graph matrix. The weight of this entry is the eigenvalue for that (no longer zero-) SCC; it may be set so as to fulfill conditions (i) and (ii) for pumped sources.

Let us now show that this approach actually can give a PDEV for the resulting, modified graph. In doing so, we provide a proof of theorem 3.

Proof 11 (Equal-gain sources share dominance) *Let $s > 1$ be the number of source SCCs. Now define $\hat{P}_\sigma$ to be the matrix resulting from pumping only source C_σ up to $\rho_{\max}$, where $\sigma = 1, \dots, s$. Let $\hat{T}$ be the totally pumped matrix, i.e. that in which all source SCCs are pumped up to the spectral radius $\rho_{\max}$.*

From section 5.2, one knows that $\hat{P}_\sigma$, for each σ , has a positive semi-definite eigenvector, i.e., one with positive weight on C_σ and on SCCs downstream from C_σ , and with zero weight on the remaining SCCs—in particular, on all other (un-pumped) source SCCs. Let $\vec{\psi}_\sigma$ be the principal eigenvector of $\hat{P}_\sigma$:

$$\hat{P}_\sigma \vec{\psi}_\sigma = \rho_{\max} \vec{\psi}_\sigma . \quad (50)$$

One then has a set of s eigenvectors, $\Sigma = \{\vec{\psi}_\sigma\}$, for the s distinct matrices $\hat{P}_\sigma$, all with same principal eigenvalue $\rho_{\max}$.

Now, for each σ , $\vec{\psi}_\sigma$ is also one of a set of degenerate eigenvectors of $\hat{T}$, with the common eigenvalue:

$$\hat{T} \vec{\psi}_\sigma = \rho_{\max} \vec{\psi}_\sigma . \quad (51)$$

That is, no other sources except C_σ deliver weight to downstream SCCs under action of $\hat{T}$, and hence do not perturb the solution $\vec{\psi}_\sigma$. This is because $\hat{T}$ differs from $\hat{P}_\sigma$ only in the weights of arcs which are internal to other source SCCs $\sigma' \neq \sigma$; but these other source SCCs have zero weight in $\vec{\psi}_\sigma$, hence the difference between $\hat{T}$ and $\hat{P}_\sigma$ has zero effect. (Note a close connection with the logic of Lemma 3.)

One can now obtain a PDEV with the same eigenvalue by taking any linear combination, with positive coefficients $c_\sigma > 0$, of the positive semi-definite eigenvectors:

$$\vec{\Psi} = \sum_{\sigma} c_{\sigma} \vec{\psi}_{\sigma} ; \quad (52)$$

giving

$$\hat{T} \Psi = \sum_{\sigma} c_{\sigma} (\hat{T} \vec{\psi}_{\sigma}) = \rho_{\max} \sum_{\sigma} c_{\sigma} \psi_{\sigma} = \rho_{\max} \Psi . \quad (53)$$

$\vec{\Psi}$ is not zero anywhere, since it is a positive sum of vectors, and every SCC has positive weight in at least one of these vectors. Hence Ψ is a PDEV for the pumped graph $\hat{T}$, with eigenvalue $\rho_{\max}$. This completes the proof.

Note that $\vec{\Psi}$ has s free constants. One simple choice is to set all the c_σ equal; but this is not the only choice. One needs considerations other than pure topology to guide in the choice.

In summary, we see that pumping sources in a reducible graph offers a promising way of generating a PDEV and hence a global importance score. To do this, increase the gain of all sources to a common, dominant value, one source at a time; any linear combination of the resulting dominant whole-graph eigenvectors, with positive coefficients, is a PDEV for the entire, modified graph $\hat{T}$ in which all sources are pumped simultaneously.

Note however that $\hat{T}$ is not strongly connected. Thus a crucial difference between this approach and the others, is that it does not modify the graph by making it strongly connected. It only tunes the importance of certain sources to allow them to play a role in the global flows of the graph. We provide some test results for this approach below.

7.3 Topological cost incurred by sink remedies

Web ranking schemes all contain some arbitrariness in their methodologies for ranking pages. The choices they make can reasonably be called their *ranking policy*. In some cases a policy is based on machine-computable values; in others it is based on payment, or on human selection. In all cases, the choice is a heuristic one, and the success of a scheme in satisfying users can be characterized as a *subjective* measure. There is no absolute scale of quality that truly measures what one is looking for in a search engine.

In our work here, we are only interested in machine-computable criteria for ranking. How then, if the policies are evaluated by subjective criteria, shall one compare or categorize different machine-computable ranking methods, including our own? In this subsection, we give a brief discussion of some graph-theoretical differences among the various methods that we have discussed. In particular, we will characterize the “topological cost” of various types of sink remedies, in terms of the number of links that must be added to a given graph. We will also point out that connectedness of the graph is not preserved by all sink remedies.

First let us consider what properties we insist on in an algorithm. Self-consistency is an essential feature of the network. That is, a method must provide a single, consistent scoring for all nodes in the graph, without exception. The score should be finite and positive. This criterion has played a large role in the considerations of this paper; and we have shown that this criterion is not in general satisfied in a reducible graph. However, all of the various types of sink remedies that we have discussed satisfy this criterion.

Thus, all ranking methods need to find sink remedies. Typically, sink remedies work by modifying the topology of the basic graph somehow. Such topological modifications, while giving a single, consistent global scoring for all nodes,

have a price—that is, there is the danger that topological modifications can distort the information in the original graph. Hence we consider three kinds of risks, or “costs”, associated with sink remedies:

- Large modifications of the topology.
- Modifications that introduce a bias in the graph.
- Modifications that disconnect the graph.

Large modifications of the topology involve two potential problems. First, there may be a large computational cost associated with introducing such large changes. Second, there is always the danger that a large modification may in fact dilute the information in the given topology too much—that is, so much that the resulting eigenvector reflects more the topology modification than it does the original topology. Also, one must take care, with a large change, to avoid introducing any bias.

Modifications which give a bias—favoring some nodes and disfavoring others—are clearly undesirable. The same holds for modifications which disconnect the graph: a disconnected graph gives multiple dominant eigenvectors (one for each connected component); and there is no obviously reasonable way to compare the scores obtained in the different connected components.

Let us now consider the random surfer operator RS as a sink remedy, in the light of these topological costs. It is clear that the RS operator adds order N^2 additional links to the graph, by adding an all-to-all graph to the original graph. Thus, we consider RS to be a “large” modification. However, it does not (as we will show below, and as shown in [7]) incur a correspondingly large computational cost, since the effects of the RS operator may be reduced to operations of $O(N)$. Furthermore, the effect of dilution from the RS operator may be held low, using the weighting parameter ϵ ; the RS operation is clearly not biased in any way; and it does not disconnect the graph. In fact, no method which adds links can disconnect the graph; hence we will not mention this aspect further, until we come to the HITS method.

We mention in this context a related approach, used by the search engine WiseNut [30]. The WiseNut approach makes the graph strongly connected by adding a fictitious Supernode S, to which all other nodes are connected by symmetric links. Thus, by this mechanism, all nodes are connected to all others within at most two hops. The number of links added in this approach is clearly of $O(N)$; hence we will also consider this modification to be “large”. The effects of the added node S and its links are, as with the RS operator, computable with $O(N)$ operations [30]. Also, considerations regarding dilution and bias are the same as for RS.

The number of links added by the link-reversal method LR depends on the structure of the directed graph. We consider two cases: (i) where the number of SCCs is $O(1)$; and (ii) the number of SCCs is of $O(N)$. In case (i) the number of links added by the LR approach is also of $O(1)$. Case (ii) must however also be considered. An obvious example of case (ii) is a DAG of N nodes, which has

N SCCs. However, a more realistic example is a small number of very large SCCs, surrounded by a “cloud” of very many, very small SCCs (often having only one node each). There is some evidence that this latter picture holds for the WWW [4]. Assuming the number of these small SCCs is of $O(N)$, we get case (ii). For such cases, the number of links added by the LR method is also of $O(N)$.

Thus case (ii) for the LR method involves a “large” topology modification. We show in the next subsection that the computational cost of doing so is also of $O(N)$. We note here that *all* links added by the LR method are used solely to connect distinct SCCs—no change is made to the internal structure of any SCC. Furthermore, in case (ii), the vast majority of new links are used to provide two-way connections between the many small SCCs and the others (likely, in the WWW case, the one or two large ones). Finally, we note that every added link connects nodes which are already connected in the original graph. Thus we would argue that there is no obvious bias involved in the LR method.

Finally we come to the question of “dilution” of the original graph by the LR method. One may be tempted to believe that the dilution is as small as it can be, while still giving a modified graph which is strongly connected—or that removing any one of the links added by the LR method will render the graph non-strongly connected. However, neither of these statements is true in general. Furthermore, the two statements are not equivalent.

Let us clarify this. We label these statements: (A) there is no way to render a given graph strongly connected (SC) with fewer links than those used by the LR method; and (B) removing any one of the links added by the LR method will render the graph non-SC. Clearly, (A) implies (B); but (B) does not imply (A), because the two statements are not equivalent (as we will show).

First, recall that we have defined the LR method such that it adds one reverse link for every link in the metagraph—ie, for every inter-SCC link. This is ‘greedy’ in the sense that it suffices to add only *one* such reversed inter-SCC link for every pair of linked SCCs. This ‘greedy’ choice is motivated by the desire to avoid introducing any bias. However, since we here are focusing on topological cost, let us consider now the ‘minimal’ LR strategy (termed LRm), which always adds exactly one reversed link for every pair of linked SCCs—that is, one link for each link in the component graph [24].

Statement (A) does not hold, even for the LRm method. Proof by counterexample: consider a directed graph whose component graph has s SCCs, connected in a linear chain, running from source to sink. The LRm method requires $s - 1$ added links. We can however render the entire graph SC by simply adding one link, pointing from the sink SCC to the source SCC. (This rather strongly alters the flow properties of the graph; but we are only counting added links here.) Hence, for $s > 2$, we can render the graph SC with fewer links than those used in LRm.

Statement (B) (with $LR \rightarrow LRm$) holds for this counterexample to (A); but it does not hold in general. A counterexample to (B) may be constructed as follows: we take our linear chain of SCCs (again with $s > 2$), and add one more link ℓ , pointing from the source SCC directly to the sink SCC. Given this input

topology, the LRm method will add one reverse link for each pair of adjacent SCCs in the chain of SCCs, and will also add the reverse ℓ_r of the link ℓ . The result is a ring of the original SCCs, now two-way connected around the ring. If any one of the new links is removed from this ring, the result is still strongly connected. Hence (B) does not hold for this family of graphs.

Finally we show that (A) and (B) are not equivalent. Since neither is true in general, equivalence must be taken to mean that, for any graph for which (A) holds, (B) also holds; and vice versa. To show that (A) and (B) are not equivalent, we again take the set of graphs whose component graph is a linear chain with $s > 2$ SCCs. For this set of graphs, (B) holds, but (A) does not. Therefore, (A) implies (B), but (B) does not imply (A).

In short: neither the LR nor the LRm method uses, in general, a minimal number of links to render the graph SC. What we can say is that the LR method uses a number of links which is typically proportional to the number of SCCs, and which is of at most $O(N)$. Finally, we note that the new links can be given a weight ϵ , in order to minimize dilution effects.

Next we consider the pumping method for remedying sinks. The pumping method modifies weights on links in all source SCCs. Furthermore, it must add self-loops to all source SCCs which are zero SCCs (ie, those with only one node), since there is no link in such SCCs which one can “pump”. Thus the number of added links is in fact equal to the number of source SCCs which only have one node. We note that there is evidence from [4] that such SCCs exist in significant numbers in the WWW; hence we cannot ignore them. However, these self-links (lying on the diagonal of the adjacency matrix) are qualitatively different from the added links discussed above. That is, they do not alter the *pattern* of flow of weights in the graph—only the *magnitudes*. Hence we consider these added links to be a distinct *type* of perturbation from the added links of RS, WiseNut, or LR. That is, there is a real sense in which they do not change the topology, but simply have an effect just like that due to the adjustment of weights on existing links: namely, both modifications affect the magnitudes of weight flows over the existing topology. Furthermore, the number of added self-links is of $O(1)$ (or even zero) in case (i), and of $O(N)$ (or less) in case (ii). For these reasons, we do not consider these added links to be a “large” modification.

Furthermore, we claim that the pumping method is also free of bias. Of course, in some sense it “favors” nodes lying upstream of the dominant SCC of the graph. But it may be tuned so as to only favor those nodes enough to give them nonzero scores. Furthermore, it does so in a way (as noted just above) which ensures that the flow of weights always follows the topology of the given graph. Thus we would not attribute any biasing effects to the pumping method.

Finally, we come to the HITS method. Here one adds *no* links to the given directed graph. Instead, one finds dominant eigenvectors for two quadratic operators built from the given graph’s adjacency matrix. With no added links, there can be no cost, and no bias. Nevertheless the HITS approach is not risk-free: as noted earlier, the HITS operators have effective, symmetric graphs, which may however be disconnected. We refer the reader again to Figure 1(a). Any graph which may be represented as in Figure 1(a), with the black dot

Sink remedy	L	C
Random surfer RS	$O(N^2)$	Yes
WiseNut	$O(N)$	Yes
Link reversal LR	$O(N)$ (max)	Yes
Pumping	Self-links; $O(N)$ (max)	Yes
HITS	0	No

Table 1: Topological cost (in number L of new links) for various sink remedies. Also we show whether or not the given sink remedy is guaranteed to preserve the property of being connected.

representing a single node, will be disconnected under operation of either the Hub operator $A^{\text{hub}} = \hat{B}\hat{F}$ or the Authority operator $A^{\text{auth}} = \hat{F}\hat{B}$. This may be seen by simply applying either operator to the graph as depicted. One can also replace the black dot in Figure 1(a) with more complex SCCs, such that A^{auth} and/or A^{hub} still represent disconnected effective graphs.

We summarize these observations in Table 1. Here L is the number of links added by the approach, and C is the property that, if the original graph is connected, then so is the effective or modified graph.

We include the remark on ‘self-links’ in the pumping row of the table as a reminder that we view the added links from the pumping method as a distinct (and ‘weaker’) type of modification than that from the other methods.

7.4 Computational costs of sink remedies

In this subsection we consider the computational costs associated with the various sink remedies under consideration. We will restrict our discussion to the case that the given graph has a *sparse* adjacency matrix. For very large document sets such as the World Wide Web, sparsity of the adjacency matrix is both *necessary* (because page scoring by eigenvector computation would otherwise be practically infeasible), and *guaranteed*—because the average degree of Web pages cannot grow linearly with the number N of pages, when N is very large.

For sparse matrices, the simple operation (matrix) $\times$ (vector) involves $O(N)$ operations. The power method for finding the dominant eigenvector may be practically implemented using a number of iterations which is independent of N (see for example the original PageRank paper [7]). Thus, for sparse matrices, and in the absence of sink remedies, we find a (not very useful) dominant eigenvector in a time which is of order N .

This scaling behavior thus gives us a baseline to which we can compare the computational costs of various sink remedies.

We begin with the RS operator. As shown in [7], the effect of the $N(N-1)$ links added by RS may be mostly computed offline. What remains are two operations: computing the sum of the current set of weights, and adding a

vector (whose entries are proportional to this sum). For normalized methods such as PageRank, the sum is known beforehand and need not be computed. For non-normalized methods, this sum may be computed, using operations of $O(N)$ (since the vector is of length N). Or, it may be forced to a fixed value (by normalizing at each iteration); this requires computing it [again, $O(N)$] and then dividing [also $O(N)$]. Hence, in any case, obtaining the sum is not costly. Also, adding two vectors of length N costs N operations. Thus we see that the cost of the RS operation does not exceed (and typically comes far under) the cost of the power method itself.

Similar considerations, and the same conclusion, hold for the WiseNut approach [30].

Next we consider link reversal (LR). LR requires the finding of the set of SCCs of the graph. This is a well-known problem, which has been shown [31, 32] to cost $O(N + L)$, where L is the number of links in the given graph. For sparse matrices, L is of $O(N)$, and hence so is the cost of finding the SCCs. The LR method then needs to find all inter-SCC links. Known algorithms [31, 32] are not guaranteed to do this. However, even the most naive and inefficient approach of searching through all the links will cost $O(L)$ operations, and hence $O(N)$ for sparse matrices.

Now we consider pumping. Pumping, relying on the SCC structure, incurs the same cost as does LR—again, an acceptable cost. However, the pumping method, in addition, must find (i) the dominant eigenvalue for every source SCC, and (ii) the dominant eigenvalue for the entire graph. It is sufficient for these two purposes simply to find the dominant eigenvalue for *every* SCC in the graph; so we will consider the cost of doing this.

We suppose that the given directed graph has N_s 'small' SCCs (with node number of $O(1)$), and N_b 'big' SCCs, each having node number of $O(N)$. Suppose further that N_s is of $O(N)$, while N_b is of $O(1)$. This is a worst case: neither N_s nor N_b can have a scaling behavior which grows faster with N . Now we consider the computational cost of this (worst) case. Each 'small' SCC yields its dominant eigenvalue (DEV) after $O(1)$ operations; hence finding the DEV for all of these costs (in this worst case) $O(N)$ operations. Each 'big' SCC is assumed also to be sparse—since otherwise we cannot maintain the assumption that the entire graph is sparse. Hence finding the DEV for each 'big' SCC costs $O(N)$ operations; and since their number N_b is of $O(1)$, finding the DEV for all 'big' SCCs costs of $O(N)$ operations. Thus we find that, even in the worst case, finding the DEV for *every* SCC costs $O(N)$ operations.

Combining the conclusions obtained in the previous two paragraphs, we find that the total computational cost of the pumping method is of $O(N)$. Hence this cost also scales in a way which is not worse than that for the power method.

Finally we come to the HITS approach. Here the reasoning is simple. The compound operators $\hat{H}$ and $\hat{A}$ are not guaranteed to be sparse. However—as shown by Kleinberg [12]—iterations of the sparse, nonquadratic operators A and A^T may be used to find the DEVs for both $\hat{H}$ and $\hat{A}$. Hence the HITS approach has the same computational cost as does the power method itself.

In conclusion, we find that all of the considered sink remedies have a com-

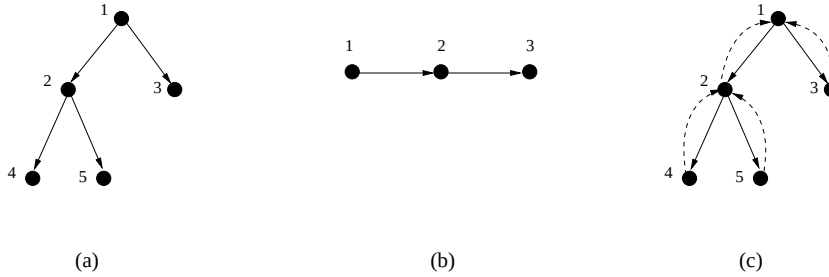


Figure 7: Directed acyclic graphs are pathological cases with highly degenerate eigenvalues. The distributions that result from the figures tend to dump all weight at the sinks. (a) and (b) show two illustrative forms of DAG. In (c) we can interpret the black dots as strongly connected components, and the DAG as a metagraph. Then (c) illustrates an economical way of ensuring strong connectivity for the entire graph, by simply applying a weighted edge in the reverse direction for each edge in the metagraph.

putational cost which is of $O(N)$, and hence not worse than that for the power method itself.

8 Examples, applications and preliminary evaluation

In this section we offer some examples which help to illustrate the ideas and results presented so far.

8.1 Directed acyclic graphs

A directed acyclic graph (DAG) is the extreme limit of a directed graph, in the sense that there are no nonzero strongly connected components. DAGs are also of interest because the metagraph for a general directed graph is a DAG. Hence it is possible to view both kinds of graphs in a single picture, with qualifications below.

DAGs are, in fact, pathological, in the sense that they have only zero eigenvalues, and dump all of their weight onto the sources or sinks. Consider the examples in fig. 7. The forward operator $\hat{F}$ has at least one nonzero eigenvector for each of these graphs. Example eigenvectors are: $\vec{\psi}^T = (0, 0, 1, 1, 1)$ for the tree in (a), and $\vec{\psi}^T = (0, 0, 1)$ for the chain in (b). Note that only sinks get nonzero weight. For the $\hat{B}$ operator, only sources will have nonzero weight.

All of the eigenvalues of a DAG are zero. Furthermore, there are not as many independent eigenvectors as there are nodes for a DAG. Instead, for $\hat{F}$, there is one independent nonzero eigenvector for each sink; and for $\hat{B}$, one for each source. For example, the eigenvector $\vec{\psi}^T = (0, 0, 1, 1, 1)$ for the tree

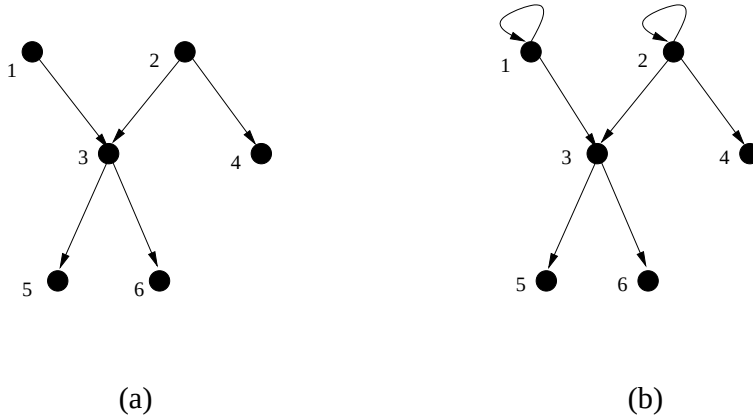


Figure 8: By adding loops at the sources of a DAG, we can ‘pump’ weight into the source node, and hence into the entire graph, on every iteration. This gives the sources a positive eigenvalue (making them dominant), and allows the graph to sustain a global distribution.

in (a) may be viewed as a linear combination of the three independent ‘sink’ eigenvectors $(0, 0, 1, 0, 0)$, $(0, 0, 0, 1, 0)$, and $(0, 0, 0, 0, 1)$. These results also hold for normalized operators such as $\hat{f}$ and $\hat{b}$. That is, dumping all weight at the ‘ends’ (sources or sinks) of a DAG is not sensitive to normalization, which affects upstream nodes only.

These results do not seem useful; however it is instructive to see them as a special case of all our results for a general directed graph, represented as a metagraph which is itself a DAG. To do this, we recall that each node in a primitive (ie, not meta) DAG, such as those in fig. 7, may be viewed as a 1×1 SCC with a gain of zero. (If there were a self-loop on any such node, it would have a gain equal to the weight on the self-loop.) Then we immediately see that no node can be dominant. Furthermore, we have an extreme version of the special case, mentioned in Section 5.1, where two SCCs with equal spectral radius have a non-isolating connection. There we found that the upstream SCC could not ‘drive’ the one downstream, so that the upstream SCC gets only zero weight. The results here for a DAG (again viewing each node as an SCC with zero spectral radius) are simply the many-SCC extension of this result: all weight ends up at the only SCCs which are not upstream of any other.

The reason that DAGs fail to distribute weight evenly amongst nodes is clear: the sources can never be dominant—nor can any other SCC—since all nodes are zero SCCs, with equal (zero) eigenvalues. We can rectify this, however, by borrowing the ‘pumping’ idea from the previous section. That is, we can enhance the gain of any node (but especially the sources—see fig. 8) with ‘pumps’ or self-loops. A self-loop acts as a sustainable source of weight flow that counterbalances the sinks, and can lead to a steady, nonzero flow of weight throughout the structure.

For example, the modification from fig. 8(a) to fig. 8(b) may be represented in terms of the corresponding adjacency matrices as follows:

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} g_1 & 0 & 1 & 0 & 0 & 0 \\ 0 & g_2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (54)$$

Here g_1 and g_2 are the gains on the self-loops at nodes 1 and 2, respectively. When $g_1 \neq g_2$, there are two distinct eigenvectors: one in which node 2 has zero weight, and one in which node 1 has zero weight. If we wish to find a single eigenvector covering the entire graph, then we can set $g_1 = g_2$. The two eigenvalues then become degenerate, and we can take any linear combination of the two eigenvectors, of the form c_1 (driven by node 1) + c_2 (driven by node 2). The nature of the eigenvectors themselves will be determined by the gain(s) chosen. For example, the node-1 eigenvector, with gain (eigenvalue) g_1 , is $(1, 0, 1/g_1, 0, 1/g_1^2, 1/g_1^2)^T$. From this we see that large gains at the sources will concentrate the weight there. Similarly, for small gains < 1 , weight will concentrate at the sinks. (Our conditions for source pumping simply require that $g_1 = g_2 > 0$ for this DAG.) We expect these conclusions to hold for the metagraph case as well.

These simple examples show the nature of the eigenvectors for a DAG, and at the same time illustrate many of our ideas and results using the metagraph picture for a general directed graph.

8.2 The World Wide Web

Of the possible applications of this work, we mention particularly the problem of page ranking in World Wide Web search engines. A general characterization of the WWW was given in ref. [33]. In this work, Broder et al describe a very large strongly connected component with about 56 million nodes (in 1999). We refer to this as the ‘giant component’, or GC. A subgraph called ‘IN’ (44 million nodes) has the property that every node in IN has a directed, non-isolating path to the giant component, and hence to every node in the giant component. Nodes in ‘OUT’ (44 million nodes) can all be reached by a directed path from the giant component; equivalently, a backward operator can reach the giant component from any node in OUT. There are also ‘Tendrils’ flowing from IN to sinks, and into OUT from sources; and ‘Tubes’ flowing from IN to OUT without visiting the giant component. (Flows *from* sources *into* IN are part of IN by definition, as flows from OUT to sinks are part of OUT.) The Tendrils and Tubes consisted (in 1999) of another 44 million nodes. Finally, there are disconnected components. More recent work has been carried out in ref. [4].

The PageRank method has been used to obtain an importance score for nodes on the Web, with great success. As noted previously, PageRank uses the

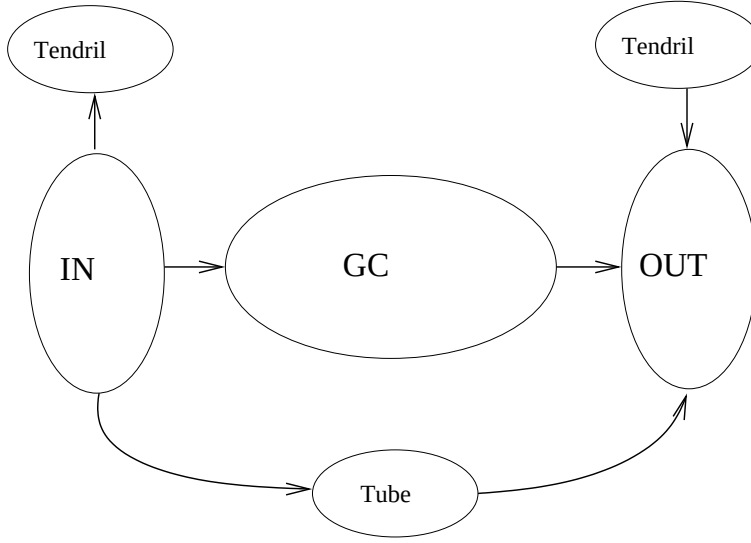


Figure 9: The giant Web components as described by Broder et al.

normalized forward operator $\hat{f}$, supplemented with a 'random surfer' operator which adds, typically, a complete graph, with a given, adjustable, weight. The resulting modified Web graph is strongly connected, and so gives a PDEV—a dominant eigenvector with positive weight at all nodes.

What is the dominant eigenvector for a non-normalized operator—not supplemented by any augmenting links—on this Web graph? Before answering this question, we first note that the description of Broder et al [33] takes account of the existence of zero SCCs only implicitly. That is, such nodes are implicitly included in the description of the connections between the main components. Hence we need to clarify the nature of the connections (isolating, or not) in this Web graph. We begin with the strongly connected components in IN. By the definition of IN, every one of them has a non-isolating connection—directly, or via one or more other strongly connected components—to the giant component. They may have others: to other strongly connected components in IN, to a strongly connected component in a Tendril, to a strongly connected component in a Tube, or even directly to a strongly connected component in OUT.

Similarly, the giant component has non-isolating connections, both to IN and to OUT; and OUT has as many possibilities for non-isolating connections as does IN. In fact, all of the major components as described in [33] have non-isolating connections to one another, with the exception of the parts termed 'disconnected components'. From the discussion in [33], it seems that these parts may have some *isolating* connections to the rest of the Web, ie, that 'disconnected' in Ref. [33] means 'lacking an unbroken *directed* connecting path, in either direction'. In other words, these 'disconnected' components include both isolated components (by our definition) and fully disconnected components. By the definitions given

in Ref. [33], then, there are no isolated strongly connected components in any of the following: Tendrils, IN, the giant component, Tubes, or OUT.

Now we discuss the dominant eigenvector of the Web graph, using the forward $\hat{F}$ operator. We consider each of the six major non-isolated components in turn. (i) The Tendril connected to IN is a pure sink with respect to the rest of the graph. Thus, if the dominant eigenvalue arises from a strongly connected component in the IN-Tendril, there will be no weight in the dominant eigenvector in any of the other major components—although the weight may be distributed among more than one strongly connected component in the IN-tendril. (ii) If the dominant eigenvalue stems from one of the strongly connected components in IN, then the dominant eigenvector will have nonzero weight in one or more strongly connected components in IN; in the IN-Tendril; in the giant component; in all Tubes; and in one or more strongly connected components in OUT. Thus, its weight will be rather widely distributed. However there will be no weight in the OUT-Tendril. (iii) Now suppose the dominant eigenvalue arises from the giant component. This is almost certainly the likely scenario [4]. The corresponding eigenvector will have positive weight in both the giant component and in OUT. The other components will have zero weight. (iv) If the dominant eigenvalue arises from a Tube, it will also place weight in OUT, and in no other component. (v) OUT is a pure sink for the rest of the graph. Hence, if the dominant eigenvalue arises from an strongly connected component in OUT, there will be no weight in any other component. (vi) If the dominant eigenvalue arises from a strongly connected component in the OUT-Tendril then there will be weight also in OUT but nowhere else.

Hence we find that using the $\hat{F}$ operator's dominant eigenvector for Web ranking alone is not likely to give useful results, as it will very probably have zero weight on large parts of the graph. The same holds, by duality, for the $\hat{B}$ operator, since (schematically) the Web graph is self-dual.

Finally, we know that a normalized $\hat{f}$ operator (without sink remedy) will only put weight in sinks: in OUT, or the IN-Tendril.

The Web graph is an interesting structure, but not so special that it can give a useful importance function from simple adjacency matrices. Instead, one must consider the various operators discussed here, in combination with the various strategies described in section 7.2.

8.3 Some tests of interest on Kleinberg slices of the WWW

We have performed some limited tests of our proposed sink remedies, using them in combination with various forward and backward operators discussed in this paper. The aim of these tests is simply to give some feeling for what kinds of results the new methods can give. The tests reported here are very limited; but we believe that they give some (extremely preliminary) indication of the performance of the various approaches in terms of picking out 'good' pages in a document set. A comprehensive test is beyond the scope of the present paper; we leave this for future work.

The World Wide Web is a natural database for testing the ranking algorithms on directed graphs, but it is far too large for a testing and development purposes. To compare our algorithms with other scoring algorithms, we have used a simple modification of the test reported by Kleinberg in [12].

Our modification is to use a machine-implementable test of 'goodness', since tests with real users are time consuming. Our logic is as follows: millions of users (and our own experience) report that, statistically, the combined set of algorithms used by Google gives 'good' results. Therefore we use the top hits of Google, for a given search, as a machine-testable definition of goodness. We acknowledge that there may be 'good' pages (as chosen by real users) in the topic-focused subgraph of the Kleinberg test, which our 'Google-is-best' measure of goodness will not find. This is then another objection to our tests (besides their small number). However, we believe that Table 2, and the discussion following it, give valuable insight in terms of comparing a wide variety of methods.

The Kleinberg test begins by using a commercial search engine (in our case, Google) to give us a short hit-list of the top 150 hits, for a given set of keywords. The resulting set of nodes, or start-set, is then augmented by adding all pages which are out-links and in-links with respect to pages on the original hit list. Finally, of course, all links between pages in the resulting set are included.

We call the resulting Web subgraph a 'Kleinberg slice' of the whole WWW. It is small, which is advantageous for testing, and it differs from many datasets (such as the entire Web) in that it is focused around a topic.

Each Kleinberg slice is characterized by a key phrase. We have tested five sets of phrases: 'tequila', 'numerical integration', 'magic', 'mathematics', and 'turtle'. Here we report some limited results for two of the resulting Kleinberg slices. 'Tequila' was rejected as being too heavily skewed by advertisements; 'numerical integration' was far too small with only 396 nodes in the final slice, from a start set of 150 nodes, and 'mathematics' proved too large for our present computational resources. Two slices remained: 'turtle' (2503 nodes) and 'magic' (3842 nodes).

Once a slice has been identified, we perform a pure link analysis on the slice, with no recourse to any kind of text relevance. The original hit list obtained from Google is based on PageRank plus other analyses, including text analysis, so even applying PageRank alone to the slice is not guaranteed to produce good results.

Using various scoring algorithms, including our own sink-remedies, we look at the top p pages, ranked purely according to their importance score from the link analysis, i.e. their weight in the resulting PDEV. We then rate a page in the top p pages of the test method as 'good' and count one point if this page is also found in Google's original top 150 reference pages; otherwise, it is 'uninteresting', in that the method gets zero points. We give results for $\epsilon = 0.1$ and for $p = 10$ in table 2.

A hazard with taking Kleinberg slices is that there is no guarantee of a well-developed link structure. That is, a pessimist can imagine that a Kleinberg slice may be dominated by many, poorly interlinked pieces, with each piece consisting

of structures of the form: (pages from In-set) $\implies$ (one page from start set) $\implies$ (pages from Out-set).

This pessimistic picture is confirmed by our analysis. We find graphs with many sources and sinks. This may be viewed with a certain ambivalence: on the one hand such a graph is the ideal test for a sink remedy; on the other hand, it may not be very representative of a real Web graph. It is known that both IN and OUT are highly fragmented and contain many isolated sources and sinks [4]. In this respect, our Kleinberg slices are perhaps not atypical. However, in our Kleinberg slices the start-set is not as well-connected as is the giant component of the Web graph.

For example, the turtle-slice was composed of 37 disconnected components. The largest connected component (LCC) was found to have 1684 nodes, distributed over 1580 SCCs. It is clear from this that most of the SCCs in the turtle-LCC have only one node. In fact, we find 1556 one-node SCCs in the turtle LCC; at the other end of the spectrum, the largest SCC had 28 nodes. Interestingly enough, this SCC is a *source* SCC. More generally, we found 907 one-node source SCCs in the turtle SCC, 515 one-node sinks, and 140 zero SCCs that were neither source nor sink. The largest source SCC had (as mentioned) 28 nodes, the largest sink SCC had 23 nodes, and the largest remaining SCC had 27 nodes.

We found similar numbers for the magic-slice. The whole slice had 3842 nodes, breaking down into 22 connected components, with the magic LCC having 3074 nodes distributed among 2865 SCCs. Again, there were very many one-node sources and sinks. The largest SCC had 39 nodes, and was in the 'centre' of the slice (ie, it was neither a source nor a sink).

Hence we find a rather challenging SCC structure for our tests. Nevertheless, we believe that the tests are fair and unbiased. The results are as follows.

We have examined the following link analysis methods: PageRank (PR); both HITS operators; the *non*-normalized forward operator, combined with a non-normalized random surfer operator, termed F+RS; the corresponding B+RS operator; the forward operator $\hat{F}$, with link reversal, termed F+LR; $\hat{B}$ with link reversal; and the same two operators with pumping.

In the link reversal tests, the reversed links were given the weight ϵ ; and in the pumping tests, the sources were pumped by so that their final gain was $1 + \epsilon$ times the largest eigenvalue in the LCC of the slice.

The results of the tests are most interesting. Very crudely, from the results in table 2 and other similar tests, we would rank the methods as follows: pumping (either Forward or Backward) comes out on top, then F+RS/B+RS, then PageRank (PR). Of course these rankings are preliminary and only suggestive. We are currently working on more extensive tests with Wikipedia samples.

One striking point is the poor showing of the HITS algorithm itself. It is apparent, from table 2, that it is almost unable to pluck out 'good' pages from the start set. The HITS approach is the only one lacking the tuning parameter ϵ , so we were unable to vary any tuning parameter for better performance. More work is needed to understand this result. However, we have mentioned one possible explanation earlier in this paper, namely, that compound operators

ALGORITHM	Tentative Performance Rank	Points “Turtle”	Points “Magic”	Rank “Turtle”	Rank “Magic”
Pumping F+P $\hat{F}$	1	10	2	13.8	12.5
Pumping B+P $\hat{B}$	1	1	5	59	34.4
F+RS $(1 - \epsilon)\hat{F} + \epsilon\hat{R}$	2	4	4	19.5	31.3
B+RS $(1 - \epsilon)\hat{B} + \epsilon\hat{R}$	2	2	4	83	66.3
PageRank $(1 - \epsilon)\hat{f} + \epsilon\hat{r}$	3	3	1	14.7	63
Link Reversal $\hat{F}$	4	1	1	59	24
Link Reversal $\hat{B}$	4	0	1	-	24
HITS Hub A^{hub}	5	0	1	-	122
HITS Auth A^{auth}	5	0	0	-	-

Table 2: A comparison of algorithm results applied to the Kleinberg section, with the two search terms “turtle” and “magic”. The “Points” column gives the number of pages which are placed in the top 10 by each method, and that are also found in Google’s top 150 hits (the start set). The “Rank” column gives the average Google ranking for these same pages. Our “tentative performance rank” is based upon both of these measures, plus other results noted in the text. A tuning parameter value $\epsilon = 0.1$ was used in all cases having a tuning parameter.

such as the HITS operators can disconnect a graph if it has too many linear chains.

The other poor performer is link reversal (LR). This approach has a tuning parameter, but we have only performed limited testing, and have not found better performance than that reported here. We believe that the link reversal idea should be tested further, since it is extremely simple to implement. This method might be more successful in handling less fragmented link structures.

PageRank is known for giving good results on the whole Web. We find that the results we obtain for Kleinberg slices are rather sensitive to the value of ϵ . That is, PR gives (consistently) results like those shown in table 2 for a range of ϵ values ≤ 0.2 . If however we increase ϵ to 0.5, then the results improve remarkably (and, again, consistently). Hence, it seems that this large value of ϵ is needed to counteract the anomalous topology of a Kleinberg slice.

We have also explored a range of ϵ values for F+RS and B+RS. These methods, in contrast to PageRank, give results that are quite insensitive to ϵ , up to and including $\epsilon = 0.5$. This insensitivity to tuning parameter might be viewed as a plus, if it were to carry over into other types of datasets.

It is also worth noting that both B+RS and F+RS performed fairly well in our tests—and yet, they do not give the same sets of top hits. That is, each method appears to give useful information that might be augmented into a consistent union. This is consistent with the intuition that $\hat{F}$ singles out pages that score high at ‘being pointed to’ (as does Kleinberg’s $A^{\text{auth}} = \hat{F}\hat{B}$), while $\hat{B}$ singles out pages that are good at ‘pointing’ (similar to Kleinberg’s $A^{\text{hub}} = \hat{B}\hat{F}$).

These ideas can be stated in another way. We can compare the average ranking of the Google-ranked pages found for each method, B+RS and F+RS. The latter gave consistently higher rankings. For example, the averages for 'magic' were 31.3 for F+RS, and 66.3 for B+RS. This difference holds also for the results with 'turtle', and is consistent with the idea that PageRank (which, we assume, plays a significant role in Google's ranking procedure) is a 'pointing-to' or 'forward' score.

These comments also hold true for the pumped methods. Pumping performs well with either $\hat{B}$ or $\hat{F}$; yet, again, the results differ depending on which operator we choose (and, again, pumping $\hat{F}$ gives a higher average ranking for ranked hits than does pumping $\hat{B}$). We also note that the pumped methods have an even larger advantage if one looks at the top 50 ($p = 50$) rather than the top 10 ($p = 10$): for example, for F+P, and for both slices, almost half of the hits in the top 50 are 'good' hits by our criterion. (B+P also did well by this criterion, although not as well as F+P.) These results are encouraging: they imply that pumping may be a promising new alternative to the random surfer operator for link analysis.

Finally, we note that the results for link reversal (LR) were quite insensitive to the choice of forward or backward operator. That is, each case turned up only a few 'good' hits—but they were the same 'good' hits in each case. This is untypical of the methods examined here. It suggests that our Kleinberg slices are so poorly linked that doing link reversal largely erases the difference between forward and backward. Reversing the turtle slice added a number of new links which was equal to about 96% of the number of original links. Correspondingly, for the magic slice, the ratio (new links)/(original links) is about 83%. We suspect that Link Reversal may perform better if it is used with starting topologies which do not call for such a high number of new links.

We end this section with some comments about our tests. We have used the term 'good' in quotes in this section, precisely because we have no objective measure of goodness, and because we have (so far) confined our evaluation to a simple, machine-testable criterion of 'goodness', namely, being ranked in the top 150 by the Google search engine. We believe that this measure, although less than perfect, gives useful information that motivates further tests.

Furthermore, we note that there is an *objective* aspect to these tests: that is, the numbers in Table 2 simply tell how well a given approach is able to find the start set of a Kleinberg slice. This is a purely topological statement, which involves no consideration of goodness of pages. The results in Table 2 thus (while lacking any statistical validity) point to clear differences among the various methods in this regard. Understanding why some methods perform better than others in this (objective) regard requires understanding both the topology of the Kleinberg slices (for which we have given some information here) and the properties of the link analysis approach. We have offered some tentative remarks in this section, but leave a thorough investigation of this question, along with tests of these methods involving real users, to future work.

9 Summary

Our goal in this work has been to find useful and meaningful importance weights, generated purely from the topology of general directed graphs. Our approach has been to generalize the notion of eigenvector centrality, thus obtaining importance scores from eigenvectors of the adjacency matrix and related operators.

In this paper we have presented the possibilities for the dominant eigenvectors obtained from a set of related operators, all derived from the adjacency matrix. We find that the set of choices for directed graphs is larger, richer, and less obvious than that for undirected graphs. We also find that, for a general directed graph, there is no one operator giving a consistent and positive weight over the entire graph.

Hence one needs further manipulations, beyond the extraction of a principal eigenvector, to obtain a global score distribution with a meaningful interpretation. We have discussed the PageRank [7] and HITS [12] techniques, which represent two rather distinct methods for obtaining such a global weight function. These approaches have been subjected to extensive study in earlier work [15, 16, 17, 18]. We have gone on to propose other methods for obtaining a global weight function—methods which have not, to our knowledge, been studied before. We have proven that these new methods give such a global positive weight distribution, and we have discussed the properties of new and old methods in terms of the topological cost (added links) and the associated computational cost. Finally, we have carried out some limited comparative testing of new and old methods, using a pair of search terms and a slightly modified Kleinberg test, in which high ranking by Google was equated with a ‘good’ page. These tests, limited as they are, single out rather clearly our new ‘pumping’ method as a promising method, from among both the new and the old methods evaluated. A thorough comparative evaluation of these methods remains for future work.

A natural criterion for a mechanism that augments graphs in order to obtain a global importance ranking function, is that the mechanism ought not to perturb the given topology too strongly. If the topology is strongly perturbed, the ranking function will in some sense not represent the true structure of the graph. On the other hand, significant additional connections are needed to fully connect a directed graph in the general case. Thus, the criterion is easily stated, but difficult to quantify. We have offered some discussion of this problem in section 7.3; but there is clearly still much work to be done in this area.

Most of the work that we are familiar with, which is directed towards the same goal as ours, involves importance functions obtained from eigenvectors [7, 12, 14]. In this area, at least, we find little work done with the very basic and simple operators $A = \hat{B}$, $A^T = \hat{F}$, and $\hat{b}$ (the normalized version of $\hat{B}$). Hence we suggest further work elucidating similarities and differences between weightings and rankings from these operators, and from the (much more studied) $\hat{f}$ operator of the PageRank method. These methods could be studied in combination with the random-surfer mechanism for rendering the graph strongly connected, and with the novel mechanisms we have proposed here. Also, higher powers of the adjacency matrix (beyond two as with HITS) should be studied.

There are many applications of directed graphs, for which an importance ranking of the nodes carries valuable information. We have mentioned only a few, but there are many more applications of directed graphs: software graphs; citation networks; various kinds of social networks with asymmetric links; metabolic networks; gene activation networks; neural networks; etc. We are confident that the techniques explored in this paper will be invaluable for understanding the structures of these networks.

We would like to thank Gerhard Weikum for a critical reading of the manuscript. GC and KEM acknowledge partial support by the Future & Emerging Technologies unit of the European Commission through Project BISON (IST-2001-38923), and by the EU within the 6th Framework Programme under contract 001907 (DELIS).

References

- [1] M. Burgess, G. Canright, and K. Engø. A graph theoretical model of computer security: from file access to social engineering. *International Journal of Information Security*, 3:70–85, 2004.
- [2] K.J. Ottenstein and L.M. Ottenstein. The program dependence graph in a software development environment. In *Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments*, pages 177–184, 1984.
- [3] S.S. Battacharyya, E.A. Lee, and P.K. Murthy. *Software Synthesis from Dataflow Graphs*. Kluwer Academic Publishers, 1996.
- [4] D. Donato, S. Leonardi, S. Millozzi, and P. Tsaparas. Mining the inner structure of the web graph. In *Proceeding of the 8th International Workshop on the Web and Databases*, pages 145–150, 2005.
- [5] P. Bonacich. Power and centrality: a family of measures. *American Journal of Sociology*, 92:1170–1182, 1987.
- [6] G. Canright and K. Engø-Monsen. A natural definition of clusters and roles in undirected graphs. *Science of Computer Programming*, 53:195, 2004.
- [7] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
- [8] Joshua O’Madadhain and Padhraic Smyth. Eventrank: a framework for ranking time-varying networks. In *LinkKDD ’05: Proceedings of the 3rd international workshop on Link discovery*, pages 9–16, New York, NY, USA, 2005. ACM Press.
- [9] Joshua O’Madadhain, Jon Hutchins, and Padhraic Smyth. Prediction and ranking algorithms for event-based network data. *SIGKDD Explor. Newsl.*, 7(2):23–30, 2005.

- [10] Y. Aumann, O. Etzioni, R. Feldman, M. Perkowitz, and T. Shmiel. Predicting event sequences: Data mining for prefetching web-pages, 1998.
- [11] Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5(1):59–68, 2003.
- [12] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [13] R. Lempel and S. Moran. Salsa: The stochastic approach for link-structure analysis. *ACM Transactions in Information Systems*, page 19, 2001.
- [14] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon. PageRank, HITS and a unified framework for link analysis. Technical Report 49372, LBNL, 2002.
- [15] A. Borodin, J.S. Rosenthal, G.O. Roberts, and P. Tsaparas. Link analysis ranking: Algorithms, theory and experiments. *ACM Transactions on Internet Technologies*, (to appear).
- [16] A. Borodin, G.O. Roberts, J.S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the world wide web. *WWW Conference*, pages 415–429, 2001.
- [17] Z. Chen, L. Tao, J. Wang, L. Wenyin, and W.Y. Ma. A unified framework for web link analysis.
- [18] A.Y. Ng, A.X. Zheng, and M.I. Jordan. Link analysis, eigenvectors and stability. *IJCAI*, pages 903–910, 2001.
- [19] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, UK, 1995.
- [20] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2:113–120, 1972.
- [21] J.J. McDonald. A product index theorem. *Linear Algebra and Its Applications*, 197,198:511–530, 1994.
- [22] R.S. Varga. *Matrix Iterative Analysis*. Prentice Hall, Englewood Cliffs, New Jersey, 1962.
- [23] H. Minc. *Nonnegative Matrices*. Wiley Interscience, New York, 1987.
- [24] See for example URL. <http://www.cs.bu.edu/steng/teaching/Spring2003/lectures/lecture16.ppt>.
- [25] K. Bharat and M.R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 104–111, Melbourne, AU, 1998.

- [26] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon. Link analysis: Hubs and authorities on the world wide web. Technical Report 47847, LBNL, May 2001.
- [27] J.C. Miller, G. Rae, and F. Schaefer. Modifications of kleinberg’s hits algorithms using matrix exponentiation and web log records. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.
- [28] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Extrapolation methods for accelerating pagerank computations, 2003.
- [29] A. Arasu, J. Novak, A. Tomkins, and J. Tomlin. Pagerank computation and the structure of the web: Experiments and algorithms, 2001.
- [30] Brian S. Kim, Sudong Chung, Anurag Dod, Michael Kim, and Yeogirl Yun. Systems and methods of retrieving relevant information. USPTO patent application 20020129014, Filed Jan. 10, 2001.
- [31] Robert E. Tarjan. Depth-First Search and Linear Graph Algorithms. *SICOMP*, 1(2):146–160, 1972.
- [32] See for example URL. Elementary graph algorithms. <http://fmt.cs.utwente.nl/courses/adc/lec6.pdf>.
- [33] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Proceedings of the 9th International World Wide Web Conference*, pages 247–256, 2000.