THE SINKHORN-KNOPP ALGORITHM: CONVERGENCE AND APPLICATIONS

PHILIP A. KNIGHT*

Abstract. As long as a square nonnegative matrix A contains sufficient nonzero elements, then the Sinkhorn-Knopp algorithm can be used to balance the matrix, that is, to find a diagonal scaling of A that is doubly stochastic. It is known that the convergence is linear and an upper bound has been given for the rate of convergence for positive matrices. In this paper we give an explicit expression for the rate of convergence for fully indecomposable matrices.

We describe how balancing algorithms can be used to give a measure of web page significance. We compare the measure with some well known alternatives, including PageRank. We show that with an appropriate modification, the Sinkhorn-Knopp algorithm is a natural candidate for computing the measure on enormous data sets.

Key words. Matrix balancing, Sinkhorn-Knopp algorithm, PageRank, doubly stochastic matrix.

AMS subject classifications. 15A48, 15A51, 65F15, 65F35.

1. Introduction.

If a graph has the appropriate structure, we can generate a random walk on it by taking its connectivity matrix and applying a suitable scaling to transform it into a stochastic matrix. This simple idea has a wide range of applications. In particular, we can rank pages on the internet by generating the appropriate connectivity matrix, G, and applying a scaling induced by a diagonal matrix, D, of column sums so that $P_c = GD^{-1}$ is column stochastic.[1] Ordering pages according to the size of the components in the stationary distribution of P_c gives us a ranking. Roughly speaking, this is how Google's PageRank is derived.

An alternative method of generating a random walk on G is to apply a diagonal scaling to both sides of G to form a doubly stochastic matrix $P = DGE$. Of course, if we use this approach then the stationary distribution is absolutely useless for ranking purposes. However, in §5 we argue that the entries of D and E can be used as alternative measures. We will also see that if we apply the Sinkhorn-Knopp (SK) algorithm on an appropriate matrix to find D and E, we can compute our new ranking with a cost comparable to that of finding the PageRank. In order to justify this conclusion, we need to establish the rate of convergence of the SK algorithm, which we do in §4. Before that, in §2 we review pertinent details about the SK algorithm and in §3 we look at the symmetric case. Our numerical results are collected in §6.

*Department of Mathematics, University of Strathclyde, 26 Richmond Street, Glasgow G1 1XH Scotland (pk@maths.strath.ac.uk). This work was supported in part by a grant from the Carnegie Trust for the Universities of Scotland.

[1] If any of the columns are empty we first modify G by, for example, adding a column of ones.

2. The Sinkhorn-Knopp algorithm.

The SK algorithm is perhaps the simplest method for finding a doubly stochastic scaling of a nonnegative matrix, A. It does this by generating a sequence of matrices whose rows and columns are normalised alternately. The algorithm can be thought of in terms of matrices

$$A_0 = A,\ A_1,\ A_2, \ldots$$

whose limit is the doubly stochastic matrix we are after, or in terms of pairs of diagonal matrices

$$(D_0, E_0),\ (D_1, E_1),\ (D_2, E_2), \ldots$$

whose limit gives the desired scaling of A. We will predominantly use the second interpretation in this paper.

To describe the algorithm more formally, we introduce the operator $\mathcal{D} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ where $\mathcal{D}(x) = \mathrm{diag}(x)$. Starting with $D_0 = E_0 = I$, we let

$$r_k = D_{k-1} A E_{k-1} e \tag{2.1}$$

where e is a vector of ones, and $D_k = \mathcal{D}(r_k)^{-1}$. Now let

$$c_k^T = e^T D_k A E_{k-1}, \tag{2.2}$$

and $E_k = \mathcal{D}(c_k)^{-1}$.

Not surprisingly, the simplicity of the method has led to its repeated discovery. It is claimed to have first been used in the 1930's for calculating traffic flow [4] and appeared in 1937 as a method for predicting telephone traffic distribution [14][2]. In the numerical analysis community it is most usually named after Sinkhorn and Knopp, who proved convergence results for the method in the 1960's [19], but it is also known by many other names, such as the RAS method [1] and Bregman's balancing method [15].

Perhaps the simplest representation of the method is given in [12]. Suppose that $P = \mathcal{D}(r) A \mathcal{D}(c)$ is doubly stochastic. Manipulation of the identities $Pe = e$ and $P^T e = e$ gives

$$c = \mathcal{D}(A^T r)^{-1} e, \quad r = \mathcal{D}(Ac)^{-1} e, \tag{2.3}$$

which suggests the fixed point iteration

$$c_{k+1} = \mathcal{D}(A^T r_k)^{-1} e, \quad r_{k+1} = \mathcal{D}(A c_{k+1})^{-1} e. \tag{2.4}$$

[2] A more detailed history of the method can be found in [6]

It is straightforward to show that this iteration is precisely the SK algorithm when $r_0 = e$. Note that this can be achieved by repeatedly issuing the commands

$$\texttt{c} = \texttt{1./(A}' * \texttt{r)},\ \ \texttt{r} = \texttt{1./(A} * \texttt{c)}$$

in MATLAB.

Convergence of the SK algorithm depends on the nonzero structure of A. Recall that a nonnegative matrix A has total support if $A \neq 0$ and all its nonzero elements lie on a positive diagonal. Furthermore, it is fully indecomposable if it is impossible to find permutation matrices P and Q such that

$$PAQ = \begin{bmatrix} A_1 & 0 \\ A_2 & A_3 \end{bmatrix}$$

with A_1 square. Sinkhorn and Knopp proved the following result [19].

THEOREM 2.1. *(Sinkhorn-Knopp) If $A \in \mathbb{R}^{n \times n}$ is nonnegative then a necessary and sufficient condition that there exists a doubly stochastic matrix P of the form DAE where D and E are diagonal matrices with positive main diagonals is that A has total support. If P exists then it is unique. D and E are also unique up to a scalar multiple if and only if A is fully indecomposable.*

A necessary and sufficient condition that the SK algorithm converges is that A has total support.

Note that we are not claiming that D and E are unique, rather that if $D_1 A E_1 = D_2 A E_2 = P$ then there exists $\alpha > 0$ such that $D_1 = \alpha D_2$ and $E_2 = \alpha E_1$.

By thinking of the iteration in terms of the approximate doubly stochastic matrices

$$A_0, A_1, A_2, \ldots,$$

Sinkhorn and Knopp also showed that the algorithm converges whenever A has at least one positive diagonal. For example, if we were to scale the matrix

$$\begin{bmatrix} a & b \\ 0 & c \end{bmatrix}$$

repeatedly we would converge to the identity matrix, however the diagonal matrices in the identity $A_k = D_k A E_k$ would diverge.

The rate of convergence of the SK algorithm has also been studied by a number of authors. Soules [20] has shown that the algorithm is linearly convergent whenever the original matrix has total support. However he gives no explicit value for the rate of convergence. Soules establishes his result by treating the algorithm as a fixed point iteration on matrices and looking at the Jacobian matrix. Our interpretation of the method as an iteration on vectors enables us to improve this result.

Franklin and Lorenz [10] give a bound on the rate of convergence when $A > 0$. They use Hilbert's projective metric for vectors $x, y \in \mathbb{R}^n_+$, namely

$$d(x,y) = \log \max_{i,j} \frac{x_i y_j}{x_j y_i}.$$

For $A \in \mathbb{R}^{m\times n}_+$, we can define

$$\theta(A) = \sup\{d(Ax,Ay) | x,y \in \mathbb{R}^n_+\} = \max_{i,j,k,l} \frac{a_{ik}a_{jl}}{a_{jk}a_{il}}. \tag{2.5}$$

Franklin and Lorenz show that $\theta(A) = \theta(A_m)$ is constant for the sequence of matrices $\{A_m\}$ generated by the SK algorithm with initial matrix A. They are also able to show that the rate of convergence of the method is bounded above by

$$C = \left(\frac{\sqrt{\theta(A)}-1}{\sqrt{\theta(A)}+1}\right)^2. \tag{2.6}$$

This is an a priori bound on the rate of convergence, but it can be very weak in practice. Furthermore, the result only holds for positive matrices. As the smallest element of A approaches zero, it can be seen that C approaches 1. The result we establish in §4 is sharp and applies whenever A is fully indecomposable.

It is worth noting that we can generate a stopping criterion for the SK algorithm that can be computed very efficiently. We want to stop when $\mathcal{D}(r_k)Ac_k$ and $\mathcal{D}(c_k)A^Tr_k$ are both close to e. After each SK step the first of these criteria is satisfied (up to round-off error) as we will have just balanced the rows of A. To get an estimate of the error in the column sums we note that $A^Tr_k = \mathcal{D}(c_{k+1})^{-1}e$, so in the middle of the step we can estimate our error by computing

$$err_k = \|c_k \circ d_{k+1} - e\|_1, \tag{2.7}$$

where $d_{k+1} = \mathcal{D}(c_{k+1})^{-1}e$ and $\circ$ represents the Hadamard product.

Matrix balancing can be used as a simple technique for preconditioning a matrix. Given a fully indecomposable matrix $A \in \mathbb{R}^{n\times n}$ we can find two $n \times n$ diagonal matrices, D and E, such that the p-norms of the rows and columns of DAE are all equal. This idea was explored in [2, 9] as a method for finding a diagonal scaling such that $\kappa(DAE) \ll \kappa(A)$. By applying the SK algorithm to the matrix whose (i,j)th element is $|a^p_{ij}|$, it is easily seen that the problem is essentially identical for $1 < p < \infty$. The case $p = \infty$ is studied in [7, 18].

3. Balancing symmetric matrices. If A is symmetric then it it is natural to look for a diagonal matrix D such that DAD is doubly stochastic. We can do this using the SK algorithm: if $\mathcal{D}(r)A\mathcal{D}(c)$ is doubly stochastic then so is its transpose $\mathcal{D}(c)A\mathcal{D}(r)$

and since, up to a scalar factor, the balancing is unique (by Theorem 2.1), $r = \alpha c$. If $\alpha \neq 1$ we can scale our limiting vectors to regain symmetry.

During the iteration, though, symmetry is lost and an alternative approach is to generate a sequence of symmetric iterates. The symmetric analogues of (2.3) and (2.4) are

$$x = \mathcal{D}(Ax)^{-1}e. \tag{3.1}$$

and

$$x_k = \mathcal{D}(Ax_{k-1})^{-1}e. \tag{3.2}$$

We note that this iteration can be coded in MATLAB by repeated application of the single instruction `x = 1./(A*x)`, which must make it one of the most compact algorithms in numerical analysis!

While the iteration superficially retains symmetry it is in fact no different from the SK algorithm. Comparing (3.2) with (2.4) we see that for $k \geq 0$, $x_{2k} = r_k$ and $x_{2k+1} = c_{k+1}$.

Conversely, we can use the iteration given by (3.2) on nonsymmetric matrices: simply apply it to

$$S = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}. \tag{3.3}$$

This is more than an academic exercise. To establish the rate of convergence of the SK algorithm we first find the convergence rate of (3.2). This will be sufficient as, in exact arithmetic the iterates coincide.

To see this, let

$$x_k = \begin{bmatrix} y_k \\ z_k \end{bmatrix},$$

and (3.2) becomes

$$y_{k+1} = \mathcal{D}(Az_k)^{-1}e, \tag{3.4}$$

$$z_{k+1} = \mathcal{D}(A^T y_k)^{-1}e. \tag{3.5}$$

Hence

$$\begin{aligned} y_{k+1} &= \mathcal{D}(A\mathcal{D}(A^T y_{k-1})^{-1}e)^{-1}e, \\ z_{k+1} &= \mathcal{D}(A^T\mathcal{D}(Az_{k-1})^{-1}e)^{-1}e. \end{aligned}$$

However, from (2.4), we have

$$\begin{aligned} c_k &= \mathcal{D}(A\mathcal{D}(A^T c_{k-1})^{-1}e)^{-1}e, \\ r_k &= \mathcal{D}(A^T\mathcal{D}(Ar_{k-1})^{-1}e)^{-1}e, \end{aligned}$$

and we conclude that one step of the SK algorithm is equivalent to two steps of (3.2) applied to S.

Symmetric balancing is also considered in [17], where the equation $\mathcal{D}(Ax)x = e$ is solved using a Gauss-Seidel-Newton method.

4. The rate of convergence of the Sinkhorn-Knopp algorithm. We now consider the convergence of the symmetric SK algorithm in (3.2) adapting as necessary the standard tools for analysis of a fixed point iteration. At this stage, we restrict ourselves to fully indecomposable matrices as in this case (3.2) has a unique positive fixed point, but we will comment on the more general case (matrices with total support) at the end of the section.

There are two complications we have to consider when trying to establish convergence. The first is that in general the iteration does not converge as when the SK algorithm is used on a symmetric matrix the sequences $\{r_k\}$ and $\{c_k\}$ will almost surely converge to different limits. Eventually we oscillate between a pair of vectors that are scalar multiples of the fixed point. However, our ultimate goal is to establish a sharp convergence result for the general SK algorithm and it will suffice to consider the alternating subsequences.

The second complication is that around the fixed point, the Jacobian matrix has spectral radius one and so we cannot make direct use of the contraction mapping theorem. However, the nature of the subspace associated with the principal eigenvector means that this, too, can be dealt with. Soules makes similar observations regarding the SK algorithm in [20] and proves linear convergence. As we are trying to put a number to this rate, we cannot use Soules' result. Instead, using our compact representation of the iteration, we present a simple analysis that leads to an explicit value for the rate of convergence.

We first prove a couple of lemmas to confirm some of the statements made in the preceding discussion.

LEMMA 4.1. *Suppose that A is a symmetric nonnegative fully indecomposable matrix. Then there is a unique positive vector, x_*, such that $\mathcal{D}(x_*)A\mathcal{D}(x_*) = P$ where P is doubly stochastic.*

Proof. This is a trivial consequence of Theorem 2.1. For existence, suppose $\mathcal{D}(r)A\mathcal{D}(c) = P$ and let $x_* = \sqrt{\mathcal{D}(r)\mathcal{D}(c)}e$ (by symmetry r and c are collinear). If $\mathcal{D}(x)A\mathcal{D}(x) = \mathcal{D}(y)A\mathcal{D}(y)$ then, for some $\alpha > 0$, $x = \alpha y$ and $y = \alpha x$. Hence $x = y$. □

LEMMA 4.2. *Suppose that A is a symmetric nonnegative fully indecomposable matrix and that x_* is the unique positive vector such that $\mathcal{D}(x_*)A\mathcal{D}(x_*) = P$ where P is doubly stochastic. Let $f(x) = \mathcal{D}(Ax)^{-1}e$. The Jacobian matrix of $f(x)$ satisfies the following properties.*

1. *For all $x \in \mathbb{R}^n_+$, $J(x) = -\mathcal{D}(Ax)^{-2}A$.*
2. *For all $\alpha \in \mathbb{R}_+$,*

$$J(\alpha x_*) = -\frac{1}{\alpha^2}\mathcal{D}(x_*)P\mathcal{D}(x_*)^{-1}.$$

Proof.

1. This can be confirmed by a straightforward componentwise calculation, or by tensor calculus. We restrict ourselves to positive vectors to ensure that $Ax > 0$ and hence that $\mathcal{D}(Ax)$ is invertible.

2. At the fixed point, $\mathcal{D}(Ax_*) = \mathcal{D}(x_*)^{-1}$, hence $\mathcal{D}(A(\alpha x_*)) = \alpha\mathcal{D}(x_*)^{-1}$ and

$$J(\alpha x_*) = -{\frac{1}{\alpha}}^2\mathcal{D}(x_*)^2A = \mathcal{D}(x_*)(\mathcal{D}(x_*)A\mathcal{D}(x_*))\mathcal{D}(x)^{-1} = -{\frac{1}{\alpha}}^2\mathcal{D}(x_*)P\mathcal{D}(x_*)^{-1}.$$

□

We now consider the behaviour of $f(x)$ when x is in the neighbourhood of αx_*. Because of the alternating behaviour, we consider the effects of two iterations at a time.

LEMMA 4.3. *Suppose that A is a symmetric nonnegative fully indecomposable matrix and that x_* is the unique positive vector such that $\mathcal{D}(x_*)A\mathcal{D}(x_*) = P$ where P is doubly stochastic. Let $f(x) = \mathcal{D}(Ax)^{-1}e$. Let $\alpha > 0$. If $\widehat{x}$ is in an ϵ-neighbourhood of αx_* then in an appropriate norm,*

$$\min_{v\in\mathcal{V}} \|f^2(\widehat{x}) - v\| \leq |\lambda_2|^2\epsilon + o(\epsilon), \tag{4.1}$$

where $\mathcal{V}$ is the vector space spanned by x_*.

Proof. Suppose that for some $\epsilon > 0$, $\widehat{x} = \alpha x_* + d$ with $\|d\| < \epsilon$. Let $D = \mathcal{D}(x_*)$ and note that $f(\alpha x_*) = x_*/\alpha$ and $f^2(\alpha x_*) = \alpha x_*$. We can write

$$\begin{aligned}
f^2(\widehat{x}) &= f(f(\alpha x_*) + J(\alpha x_*)d + o(\epsilon)) \\
&= f^2(\alpha x_*) + J(x_*/\alpha)J(\alpha x_*)d + o(\epsilon) \\
&= \alpha x_* + (-\alpha^2 DPD^{-1})(-\alpha^{-2}DPD^{-1})d + o(\epsilon) \\
&= \alpha x_* + DP^2D^{-1}d + o(\epsilon) = \alpha x_* + J^2d + o(\epsilon),
\end{aligned}$$

where $J = DPD^{-1}$. As $\rho(P) = 1$, we cannot use the contraction mapping theorem to show that $\|f^2(\widehat{x}) - \alpha x_*\| < \|\widehat{x} - \alpha x_*\|$. However, observe that A is fully indecomposable hence P is, too, and since doubly stochastic matrices with this property are primitive [3], P has a single simple eigenvalue of modulus one. The corresponding eigenvector of J is x_*. Using Wielandt deflation [22], we can write

$$J = -(x_*y^T + J_0),$$

where

$$\sigma(J_0) = \sigma(P) - \{1\} \cup \{0\} = \{\lambda_2, \ldots, \lambda_n, 0\}$$

by choosing, for example, $y = x_*/{x_*}^T x_*$. Since $J_0 x_* = 0$,

$$\begin{aligned} f^2(\widehat{x}) &= \alpha x_* + (x_* y^T + J_0)^2 d + o(\epsilon) \\ &= J_0^2 d + (1 + y^T(J_0 + I)d)x_* + o(\epsilon). \end{aligned}$$

Choosing our norm so that $\|J_0\| \leq |\lambda_2| + \epsilon$ and letting $v = (1 + y^T(J_0 + I)d)x_*$ establishes (4.1). □

We can conclude that as our iterates approach the subspace spanned by x_*, the contribution to our iterates from other directions diminishes linearly at a rate governed by the second eigenvalue of P. The fact that we are heading for a fixed line rather than a fixed point is sufficient for us to find the scaling we crave. Since we already know that the SK algorithm converges, we can be sure that we eventually lie in a neighbourhood that satisfies the conditions of Lemma 4.3.

THEOREM 4.4. *Suppose that A is a symmetric nonnnegative fully indecomposable matrix and that x_* is the unique positive vector such that $\mathcal{D}(x_*)A\mathcal{D}(x_*) = P$ where P is doubly stochastic and let $\{x_k\}$ be the sequence of vectors generated by the iteration (3.2) with $x_0 = e$. Then for all $\epsilon > 0$ there exists $K_1 \in \mathbb{Z}$ such that if $k \geq K_1$, $x_k = \alpha_k x_* + d_k$ where $\|d_k\| < \epsilon$ and α_k is bounded. Furthermore, there exists $K_2 \in \mathbb{Z}$ such that if $k \geq K_2$,*

$$\|d_{k+2}\| \leq |\lambda_2|^2 \|d_k\|,$$

where λ_2 is the subdominant eigenvalue of P.

Proof. The existence of K_1 is guaranteed by Theorem 2.1 and our observation on the equivalence of the SK algorithm and (3.2). The existence of K_2 follows from Lemma 4.3. □

The result does not immediately extend to the nonsymmetric case as when we form S using (3.3) we lose indecomposability. This isn't a problem though.

THEOREM 4.5. *If A is fully indecomposable then the SK algorithm will converge linearly to vectors r_* and c_* such that $\mathcal{D}(r_*)A\mathcal{D}(c_*) = P$ where P is doubly stochastic. Furthermore, there exists $K \in \mathbb{Z}$ such that if $k \geq K$,*

$$\left\| \begin{bmatrix} r_{k+1} \\ c_{k+1} \end{bmatrix} - \begin{bmatrix} r_* \\ c_* \end{bmatrix} \right\| \leq \sigma_2^2 \left\| \begin{bmatrix} r_k \\ c_k \end{bmatrix} - \begin{bmatrix} r_* \\ c_* \end{bmatrix} \right\|,$$

where σ_2 is the second singular value of P.

Proof. The convergence of the algorithm is guaranteed by Theorem 2.1. To determine the rate of convergence we need to adapt Lemma 4.3. Consider the spectrum of

$J(x_*)$ when we form the matrix S using (3.3). This will be the same as the spectrum of

$$Q = \left[\begin{array}{cc} 0 & P \\ P^T & 0 \end{array} \right].$$

The conditions imposed on A ensure that P is primitive and hence so is P^TP. Since the spectrum of Q is the set of positive and negative square roots of the eigenvalues[3] of P^TP we have an additional eigenvalue of modulus one. We need to consider how the iteration behaves in the neighbourhood of the associated subspace, $\mathcal{V}$.

The two eigenvectors of $J(x_*)$ corresponding to the maximal eigenvalues take the form

$$v_1 = \left[\begin{array}{c} r_* \\ c_* \end{array} \right] \text{ and } v_2 = \left[\begin{array}{c} r_* \\ -c_* \end{array} \right],$$

Assuming $\widehat{x}$ is in an ϵ-neighbourhood of $\mathcal{V}$ we can again show that

$$\min_{v \in \mathcal{V}} \|f^2(\widehat{x}) - v\| \leq |\lambda_2(Q)|^2 \epsilon + o(\epsilon),$$

and $|\lambda_2(Q)| = \sigma_2(P)$.

We have essentially proved the theorem, we just have to identify the iterates from the symmetric algorithm that appear as iterates in the SK algorithm. Following the discussion at the end of § 3 we can identify r_k as the top half of x_{2k} and c_k as the bottom half of x_{2k-1}. This explains why the rate of convergence of the SK algorithm is σ_2^2. SK algorithm avoids the oscillations in the symmetric algorithm as r_k and c_k are formed from convergent subsequences of $\{x_k\}$. □

Theorem 2.1 states that the SK algorithm is convergent if A has total support while Theorem 4.5 only applies if A is fully indecomposable. This gap is easily reconciled: if A has total support but is not fully indecomposable then it must be a direct sum of fully indecomposable matrices. Such a matrix can be permuted into block diagonal form

$$\left[\begin{array}{cccc} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_k \end{array} \right],$$

where each diagonal block is fully indecomposable. The behaviour of the SK algorithm is unaffected by permutations (unlike the symmetric variant). If we apply the SK algorithm to the block diagonal form then clearly the convergence in each block

[3] Or singular values of P.

will be independent of all others and the doubly stochastic matrix we converge towards can be written

$$\begin{bmatrix} P_1 & & & \\ & P_2 & & \\ & & \ddots & \\ & & & P_k \end{bmatrix},$$

where each P_i is itself doubly stochastic and fully indecomposable. The asymptotic rate of convergence to P_i is $\sigma_2^2(P_i)$. If we want to talk about an overall asymptotic convergence then it will be bounded above by

$$\max_{1\leq i\leq k} \sigma_2^2(P_i).$$

However, we may not see this upper bound reached, for example, in the case that some of the A_i are already doubly stochastic.

5. Ranking web pages. The PageRank algorithm, introduced by Brin et al. [5], has proved to be an incredibly successful technique for ordering large sets of connected data. In essence, the method takes a matrix, G, representing the connectivity of a network and scales the columns so that the matrix is column stochastic.[4] The stationary distribution of this scaled matrix is then calculated, typically by using the power method, and the size of the probabilities is used to order the nodes in the network. A thorough description of the method and associated theory can be found in [16]. We note that the column scaling is trivial to achieve (requiring half a step of the SK algorithm) and the main work is in computing the stationary distribution. In this section we use the SK algorithm to compute an alternative method for ordering data which has a similar cost to PageRank but which has two principal advantages. First, for each node in our network we get two measures rather than one which we claim are analogous to the authorities and hubs of Kleinberg's HITS algorithm [13]. Second, there is no need to treat dangling nodes differently to any other whereas in the PageRank algorithm, it is necessary to preprocess the connectivity matrix in some way otherwise the column scaling fails. For example, one can make the assumption that any page on the whole web may be visited with equal probability from any dangling node.

The guiding heuristic behind the PageRank model is simple to state, namely that the random walk will visit significant web pages more frequently than insignificant ones, and the success of this graph interpretation in mimicking the subjective property of significance is one of the main reasons behind its current ubiquity.

[4] In our connectivity matrix the (i,j)th entry is one if there is a link from the jth node to the ith node.

We offer a simple heuristic to justify our application of the SK algorithm to the problem. Clearly the probabilities in the associated distribution tell us nothing as the distribution is uniform.[5] If we think in terms of the traffic flowing around the network represented by G then, our aim is to balance the flow through each node. That is, we want to scale G so that its stationary distribution is uniform, or equivalently so that it is doubly stochastic. Suppose then that $\mathcal{D}(r)G\mathcal{D}(c)$ is doubly stochastic. If node i in the unweighted graph draws traffic in disproportionately then this will have to be compensated for by r_i being relatively small. Similarly, if a node has a tendency to emit traffic then c_i will need to be relatively small. We associate the tendency of a node to emit traffic with it being a hub, a node which points to several sources of information on a topic. The tendency to draw in traffic is associated with authoritativeness, a node that contains definitive information on a particular topic. We can order the nodes with respect to each of these properties by reversing the order of the entries of r and c. This heuristic is very similar to that behind the ordinary gravity model in transport planning [1, 15], where the SK algorithm has been successfully employed.

While we believe the use of the SK algorithm in web applications is new, it is related to a technique proposed by Tomlin in [21]. Here one looks to find a vector d such that similarity transformation induced by $\mathcal{D}(d)$ on the connectivity matrix, $P = D^{-1}GD$, fixes the sum of the entries of P and, for $1 \leq i \leq n$,

$$\sum_{j=1}^{n} (p_{ij} - p_{ji}) = 0. \tag{5.1}$$

Tomlin argues that the authoritativeness of the jth node is proportional to the size of d_j while the jth row/column sum can be used as a hub measure. Tomlin's suggests an iterative algorithm for computing d, the iterative step for which can be written in MATLAB as

$$\mathtt{d = sqrt((G * d)./(G' * (1./d)));}$$

but no conditions for convergence are given although it is claimed to work in practice. A criticism of Tomlin's technique is that if G is symmetric (5.1) is satisfied with $D = I$ and the method fails to identify authorities. While G will not be symmetric in web applications, there seems to be no justification for this phenomenon.

5.1. Practicalities. On any large set of web data it is unreasonable to expect the nodes to form a single strongly connected component and our matrix is highly unlikely to be fully indecomposable. Hence it is necessary to make a perturbation to G for the SK algorithm to converge. In PageRank a damping factor is used: if P

[5] We can claim categorically that this is the worst possible method for ranking web pages!

is the column stochastic scaling of the web graph then we compute the stationary distribution of

$$P_\alpha = \alpha P + (1-\alpha)ee^T/n. \tag{5.2}$$

Inspired by this idea, we simply make a rank one perturbation to G by adding a constant γ to each element. Our justification for doing this is similar to that in PageRank: if we wish to model a random crawl on the web we have to allow a mechanism for moving between any pair of nodes. Clearly we do not want to make the perturbation explicitly as we want to take advantage of the sparsity in G, and indeed it is easily avoided. Using (2.4), and the fact that all the iterates are positive, we can write

$$c_{k+1} = \mathcal{D}((G+\gamma ee^T)^T r_k)^{-1}e = \mathcal{D}(G^T r_k + \gamma\|r_k\|_1 e)^{-1}e$$

and similarly

$$r_{k+1} = \mathcal{D}(G^T c_{k+1} + \gamma\|c_{k+1}\|_1 e)^{-1}e.$$

A MATLAB program for carrying out balancing of $G + \alpha ee^T$ using the stopping criterion for the SK algorithm (2.7) is given in Figure 5.1. All the user needs to supply is

```
function [c, r] = sk(G, tol, g)
[n, n] = size(G);
r = ones(n,1); c = r;
d = G’*r + g*sum(r);
while norm(c.*d - 1,1) > tol
    c = 1./d;
    r = 1./(G*c+ g*sum(c));
    d = G’*r + g*sum(r);
end
```

FIG. 5.1. *A balancing algorithm for web ranking.*

the connectivity graph and a choice of tolerance and the parameter γ. The cost of the algorithm is dominated by the two matrix-vector multiplies at each step. For very large values of n, the cost of the transpose is likely to be significant and the algorithm should be adapted to work with G and G^T efficiently.

The damping factor in PageRank controls the rate of convergence of the power method by fixing the size of the second eigenvalue of P_α. This is a consequence of the following theorem, a simple proof of which can be found in [8].

THEOREM 5.1. *Let P be a column-stochastic matrix with eigenvalues*

$$1, \lambda_2, \ldots, \lambda_n.$$

Then if $0 \leq \alpha \leq 1$*, the eigenvalues of* P_α*, as defined in (5.2), are* $\{1, \alpha\lambda_2, \ldots, \alpha\lambda_n\}$.

The result is also true for a more general set of rank one perturbations but if we restrict ourselves to this particular one we can extend the result to determine the singular values in the doubly stochastic case.

COROLLARY 5.2. *Let* P *be a doubly stochastic matrix with singular values*

$$1,\ \sigma_2, \ldots,\ \sigma_n.$$

Then if $0 \leq \alpha \leq 1$*, the singular values of* P_α *are* $\{1, \alpha\sigma_2, \ldots, \alpha\sigma_n\}$.

Proof. Since

$$\begin{aligned} P_\alpha^T P_\alpha &= \alpha^2 P^T P + \frac{\alpha(1-\alpha)}{n}(ee^T P + P ee^T) + \frac{(1-\alpha)^2}{n^2} ee^T ee^T \\ &= \alpha^2 P^T P + \frac{2\alpha(1-\alpha)}{n} ee^T + \frac{(1-\alpha)^2}{n} ee^T \\ &= \alpha^2 P^T P + \frac{1-\alpha^2}{n} ee^T, \end{aligned}$$

the result follows by applying Theorem 5.1 to $P^T P$. □

In many applications, α is given the value 0.85, but care must be taken to ensure that P_α sufficiently resembles P [11]. For the balancing algorithm we are unable to prove a result as strong as Theorem 5.1. However, using our convergence result for the SK algorithm, we argue that the criteria for making a good choice for the parameter γ are similar to those used in PageRank.

We can apply the Franklin-Lorenz bound (2.6) in the perturbed case to get an idea of the effect of varying γ. Since G contains only zeros and ones we have, from (2.5),

$$\theta(G + \gamma ee^T) = \max_{i,j,k,l} \frac{(g_{ik}+\gamma)(g_{jl}+\gamma)}{(g_{jk}+\gamma)(g_{il}+\gamma)} \leq \frac{(1+\gamma)^2}{\gamma^2},$$

hence the rate of convergence can be bounded above by $1/(1+2g)$. While this shows that we can expect the convergence of the algorithm to improve by increasing γ, experimental evidence shows that this severely underestimates the effect of the parameter, and a more realistic upper bound would be of the form $1/p(n,\gamma)$ for some low degree polynomial in n and γ. Such a bound is simple to prove in certain important special cases.

For example, suppose that P is doubly stochastic and we use the SK algorithm on $P' = P + \gamma ee^T$. Then $\mathcal{D}(r_k)P'\mathcal{D}(c_k)$ converges to $Q = (1+n\gamma)^{-1}P'$, since this is clearly a doubly stochastic diagonal scaling of P' and, by Theorem 2.1, such a scaling is unique. Notice that $Q = P_\alpha$ where $\alpha = (1+n\gamma)^{-1}$ and so by Corollary 5.2 and Theorem 4.5, the SK algorithm will converge asymptotically with rate $(1+n\gamma)^{-2}$. For example, choosing $\gamma = 0.1/n$ gives a convergence rate of around 0.83.

6. Results. In § 4 we showed that if the SK algorithm is used on a fully indecomposable nonnegative matrix and it converges to the doubly stochastic matrix P then the rate of convergence is asymptotically equal to the square of the second singular value of P. Generally, we have found that this asymptotic convergence rate is approached fairly quickly. This is illustrated in Figure 6.1 for three matrices. A is the 10×10 upper Hessenberg matrix whose nonzero entries are all 1, B and C are random 50×50 matrices whose nonzero entries are uniformly distributed in $[0, 1]$. They are generated so that approximately 30 percent of B's elements and 15 percent of C's elements are nonzero. The solid lines in all our graphs measure the error as the iteration progresses using (2.7), the dashed lines represent the asymptotic rates predicted by Theorem 4.5.

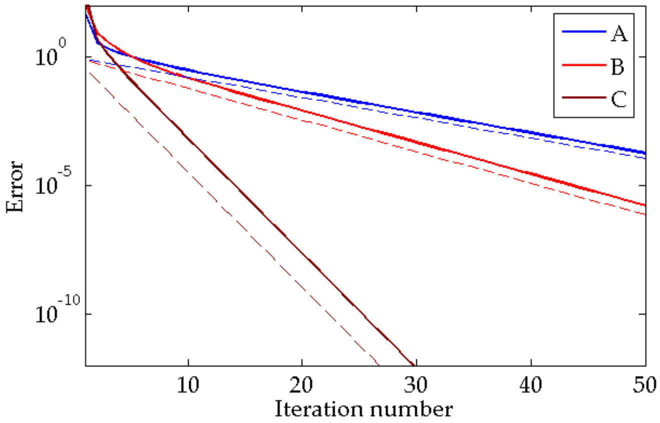

FIG. 6.1. *Rate of convergence of the Sinkhorn-Knopp algorithm.*

In § 5.1 we claimed that the rate of convergence of the SK algorithm was significantly faster when we made a uniform rank one perturbation to the original graph. In Figures 6.2 and 6.3 we provide evidence for our claim that the rate of convergence of the SK algorithm on the $n \times n$ matrix $A + \gamma ee^T$ can be bounded by $1/p(n, \gamma)$ for some low degree polynomial in n and γ.

In Figure 6.2 we show the results of varying γ on a sparse random symmetric 1000×1000 matrix with positive diagonal (which ensures that the matrix is fully indecomposable).

In Figure 6.3 we show the results of varying γ on the connectivity graph for a 2002 web crawl of Stanford University websites.[6] There are 281093 nodes and the matrix has roughly 2 million nonzero entries. In this case, if $\gamma = 0$ the matrix is not fully indecomposable. The lines show how convergence speeds up as we vary γ

[6] Available from `http://www.stanford.edu/~sdkamvar/data/stanford-web.tar.gz`.

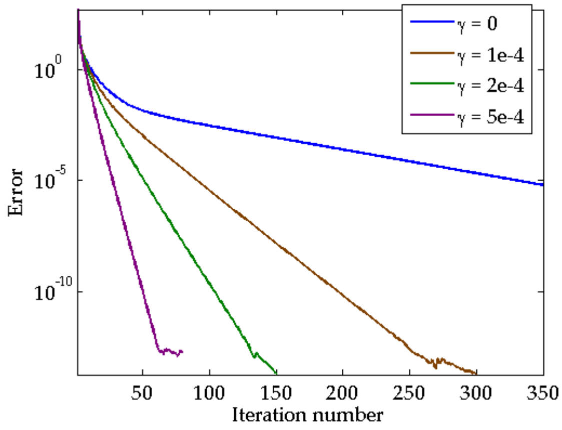

FIG. 6.2. *Varying γ for a random sparse matrix.*

through the values $0.01/n$, $0.1/n$, $0.5/n$, $1/n$, $2/n$ and $4/n$.

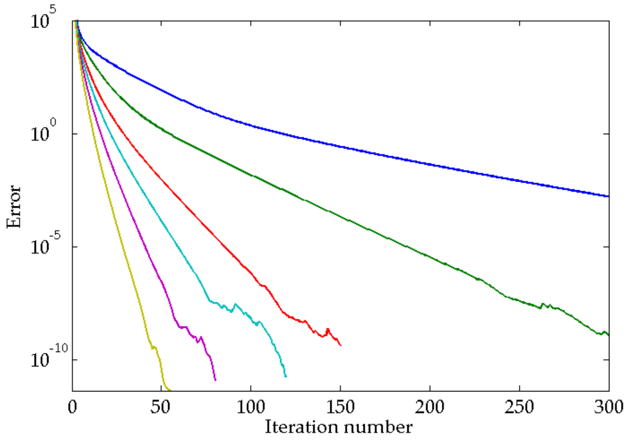

FIG. 6.3. *Varying γ for the Stanford matrix.*

We now investigate how our new measure compares with PageRank. In our first example, we look at the toy example of a graph of six webpages used in [16], whose connectivity is illustrated in Figure 6.4.

Using PageRank with $\alpha = .9$ the nodes are ordered (from most significant to least) 4, 6, 5, 2, 3, 1. Using the HITS algorithm, the order of authoritativeness is 5, 2, 6, 1, 4, 3, while the hub ordering is 3, 4, 1, 5, 6, 2. Using the algorithm in Figure 5.1 (and with $\gamma = 1/60$) we find that our ordering of authoritativeness matches the PageRank exactly. Our ordering of the hubs differs from HITS only in that nodes 1 and 4 are transposed. We should not expect the exact correspondence between PageRank and

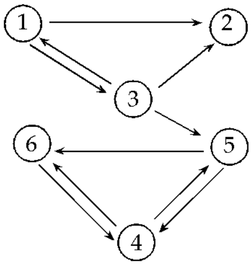

FIG. 6.4. *A miniature web graph.*

our new measure to extend to larger systems as we are trying to measure something different.

We have carried out a number of experiments on the graph of all the links between articles in the Wikipedia online database, collated in 2005. The resulting graph has just over 1.1 million nodes and there are roughly 18.3 million nonzeros in the connectivity matrix. Figure 6.5 shows a comparison of PageRank ($\alpha = .85$) against the authorities computed by the SK algorithm ($\gamma = .1/n$). The graph shows the proportion of nodes that are amongst the top N authorities and are in the top N for high PageRank for $1 \leq N \leq 1000$. We note the strong correlation between the two.

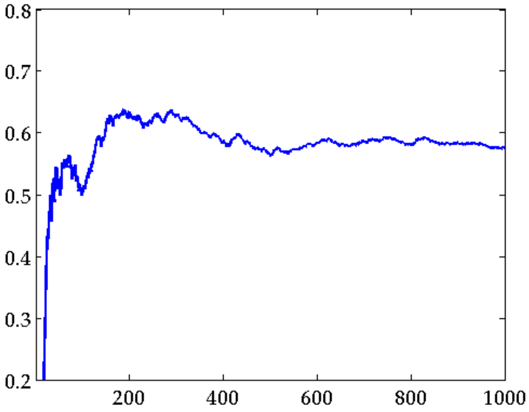

FIG. 6.5. *Comparison of web measures on Wikipedia data.*

Finally, we investigate how well the SK algorithm allows us to distinguish between hubs and authorities. Table 6.1 shows the top 10 or so nodes[7] in the Wikipedia dataset according to a variety of measures. The first column is ordered according to PageRank ($\alpha = .85$), the second according to the authorities as measured with the

[7] We have grouped certain linked terms that appeared consecutively.

SK algorithm. In the third column we have filtered out authorities whose hub rating is particularly low. Our rationale for doing this is that if an authoritative page has a high hub rating it will be linked to many other subjects and is therefore likely to be of more general interest. This is precisely what we see here, where we have only listed authorities that are also in the top 2% of hubs. The fourth column lists the top hubs, this time filtered to include only those amongst the top 25% of authorities. We note that all of the top hubs are either tables or lists.

PageRank	Authorities	Filtered Auth.	Filtered Hubs
United States	2000	2000	Political parties
Race (US Census)	Pop. density	Marriage	Environment topics
United Kingdom	km^2	US	State leaders
France	Census	2003, 2004, 2005	Airlines
2005,2004,2000	Square mile	UK/England	2 letter combinations
Canada	Marriage	Canada	Masts
England	Per capita income	Japan	Mathematicians
Cat. by country	US Census	Australia	Peerage of the UK
2003	Poverty line	2001, 2002	Record labels
Cat.:Culture	Race (US Census)	Germany	Biblical names

TABLE 6.1

Highest ranked subjects in Wikipedia.

7. Concluding Remarks. The SK algorithm can be viewed (for symmetric matrices) as a power method-like technique for solving the matrix problem $Ax = 1/x$. This connection can be seen in the similar convergence properties and costs of the two algorithms. The results of our experiments back our claim that the SK algorithm can be used to distinguish between hubs and authorities in web-type graphs at a cost similar to that of PageRank. The notion of quality of an ordering is fairly subjective, but we feel the results in Table 6.1 demonstrate that we can obtain useful information with this approach.

In order to balance speed and quality in ordering web data with the algorithm given in Figure 5.1 we suggest choosing the parameter γ to lie in the range $.01 \leq \gamma n \leq 1$. Evidence that a choice in this range can be used to compute a measure in a comparable time to PageRank is supplied by our experiments and the partial results in § 5.1.

Acknowledgements. We thank David Gleich at Stanford University for providing the data from Wikipedia.

REFERENCES

[1] MICHAEL BACHARACH, *Biproportional Matrices & Input-Output Change*, CUP, 1970.

[2] F. L. BAUER, *Optimally scaled matrices*, Numer. Math., 5 (1963), pp. 73–87.

[3] ABRAHAM BERMAN AND ROBERT J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, 1994.

[4] L. M. BREGMAN, *Proof of the convergence of Sheleikhovskii's method for a problem with transportation constraints*, U.S.S.R. Comput. Math. and Math. Phys., 1 (1967), pp. 191–204.

[5] SERGEI BRIN, LAWRENCE PAGE, R. MOTWANI AND TERRY WINOGRAD, *The PageRank citation ranking: bringing order to the Web*. Technical Report 1999-0120, Computer Science Department, Stanford University, 1999.

[6] JACK B. BROWN, PHILLIP J. CHASE AND ARTHUR O. PITTENGER, *Order independence and factor convergence in iterative scaling*, Linear Algebra Appl., 190 (1993), pp. 1–38.

[7] JAMES R. BUNCH, *Equilibration of symmetric matrices in the max-norm*, J. Assoc. Comput. Mach., 18 (1971), pp. 566–572.

[8] LARS ELDÉN, *A note on the eigenvaues of the Google matrix*. Technical Report LiTH-MAT-R-04-01, Department of Mathematics, Linköping University, 2004.

[9] GEORGE E. FORSYTHE AND E. G. STRAUS, *On best conditioned matrices*, Proc. Amer. Math. Soc., 6 (1955), pp. 340–355.

[10] JOEL FRANKLIN AND JENS LORENZ, *On the scaling of multidimensional matrices*, Linear Algebra Appl., 114/115 (1989), pp. 717–735.

[11] GENE H. GOLUB AND CHEN GRIEF, *An Arnoldi-type algorithm for computing PageRank*, to appear in BIT, 2006.

[12] BAHMAN KALANTARI AND LEONID KHACHIYAN, *On the complexity of nonnegative-matrix scaling*, Linear Algebra Appl., 240 (1996), pp. 87–103.

[13] JON M. KLEINBERG, *Authoritative sources in a hyperlinked environment*, J. Assoc. Comput. Mach., 46 (1999), pp. 604–632.

[14] R. KRUITHOF, *Telefoonverkeersrekening*, De Ingenieur, 52 (1937), pp. E15–E25.

[15] B. LAMOND AND N. F. STEWART, *Bregman's balancing method*, Transportation Research 15B (1981), pp. 239–248.

[16] AMY N. LANGVILLE AND CARL S. MEYER, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, 2006.

[17] OREN E. LIVNE AND GENE H. GOLUB, *Scaling by binormalization*, Numerical Algorithms, 35 (2004), pp. 97–120.

[18] DANIEL RUIZ, *A scaling algorithm to equilibrate both rows and columns norms in matrices*, Technical Report RT/APO/01/4, ENSEEIHT-IRIT (2001).

[19] RICHARD SINKHORN AND PAUL KNOPP *Concerning nonnegative matrices and doubly stochastic matrices*, Pacific J. Math. 21 (1967) , pp. 343–348.

[20] GEORGE W. SOULES, *The rate of convergence of Sinkhorn balancing*, Linear Algebra Appl., 150 (1991), pp. 3–40.

[21] JOHN A. TOMLIN, *A new paradigm for ranking pages on the World Wide Web*, Proceedings of the 12th international conference on World Wide Web, pp. 350–355 (2003).

[22] HELMUT WIEDLANDT, *Das Iterationsverfahren bei nicht selbstadjungierten linearen Eigenwertaufgaben*, Math. Z., 50 (1944), pp. 93–143.