Extrapolation and minimization procedures for the PageRank vector

Claude Brezinski[1], Michela Redivo-Zaglia[2]

[1] Laboratoire Paul Painlevé, UMR CNRS 8524
UFR de Mathématiques Pures et Appliquées
Université des Sciences et Technologies de Lille
59655–Villeneuve d'Ascq cedex, France.
Claude.Brezinski@univ-lille1.fr
[2] Università degli Studi di Padova
Dipartimento di Matematica Pura ed Applicata
Via Trieste 63, 35121–Padova, Italy.
Michela.RedivoZaglia@unipd.it

Abstract. An important problem in Web search is to determine the importance of each page. This problem consists in computing, by the power method, the left principal eigenvector (the PageRank vector) of a matrix depending on a parameter c which has to be chosen close to 1. However, when c is close to 1, the problem is ill-conditioned, and the power method converges slowly. So, the idea developed in this paper consists in computing the PageRank vector for several values of c, and then to extrapolate them, by a conveniently chosen rational function, at a point near 1. The choice of this extrapolating function is based on the mathematical considerations about the PageRank vector.

Keywords. Extrapolation, PageRank, Web matrix, eigenvector computation.

1 The problem

The mathematical problem behind web search is the computation of the nonnegative left eigenvector of a $p \times p$ matrix P corresponding to its dominant eigenvalue 1, where p is the number of pages in Google (8.06 billions at the end of March 2005). Since P is not stochastic (some rows of P may contain only zeros due to the so–called dangling nodes), it is replaced by the matrix

$$\widetilde{P} = P + \mathbf{d}\mathbf{w}^T$$

with $\mathbf{w} \in \mathbb{R}^p$ a probability vector, that is such that $\mathbf{w} \geq 0$ and $(\mathbf{w}, \mathbf{e}) = 1$ with $\mathbf{e} = (1, \ldots, 1)^T$, and $\mathbf{d} = (d_i) \in \mathbb{R}^p$ the vector with $d_i = 1$ if $\deg(i) = 0$, and 0 otherwise, where $\deg(i)$ is the outdegree of the page i, that is the number of pages it points to.

Since the matrix $\widetilde{P}$ is not irreducible, it is replaced by the matrix

$$P_c = c\widetilde{P} + (1-c)E,$$

Dagstuhl Seminar Proceedings 07071
Web Information Retrieval and Linear Algebra Algorithms
http://drops.dagstuhl.de/opus/volltexte/2007/1068

where c is a parameter between 0 and 1, and $E = \mathbf{e}\mathbf{v}^T$ with $\mathbf{e} = (1, \ldots, 1)^T \in \mathbb{R}^p$ and $\mathbf{v}$ a probability vector. Thus $P_c\mathbf{e} = \mathbf{e}$.

The unique nonnegative dominant left eigenvector of P_c is denoted $\mathbf{r}_c$. So, $\mathbf{r}_c = P_c^T\mathbf{r}_c$. This vector can be computed by the power method which consists in the iterations

$$\mathbf{r}_c^{(n+1)} = P_c^T\mathbf{r}_c^{(n)}, \qquad n = 0, 1, \ldots$$

with $\mathbf{r}_c^{(0)} = \mathbf{v}$. These iterations converge to $\mathbf{r}_c$ as c^n, and originally Google chose $c = 0.85$, which insures a good rate of convergence. Anyway, since the computation of the pagerank vector can take several days, various methods for their acceleration have been proposed [7,2].

The vector $\widetilde{\mathbf{r}} = \lim_{c\to1}\mathbf{r}_c$ is uniquely determined as the limit, when c tends to 1, of the family of vectors $\mathbf{r}_c$. However, it is just one of the infinitely many solutions of $P^T\mathbf{r} = \mathbf{r}$, $\mathbf{r} \geq 0$, $(\mathbf{r}, \mathbf{e}) = 1$, which form a nontrivial convex set. Notice that the conditioning of the matrix P_c grows as $(1-c)^{-1}$, but that the function $\mathbf{r}_c$ is analytic in a small neighbourhood of 1 in the complex plane [6]. For a detailed analysis of the sensitivity of the vector $\mathbf{r}_c$, see [10]. We refer to [8] for detailed explanations about the origin, the mathematical properties, and the treatment of this problem.

An idea for obtaining approximations of $\lim_{c\to1}\mathbf{r}_c$ is to compute the vector $\mathbf{r}_c$ for different values of c away from 1, to interpolate them by some vector function, and finally to extrapolate this function at the point $c = 1$, or at any other point close to 1. Of course, in order to obtain good results, the interpolating function has to mimic as closely as possible the exact behavior of $\mathbf{r}_c$ with respect to c. This behavior was analyzed by Serra–Capizzano [9] and Horn and Serra–Capizzano [6], who proved that $\mathbf{r}_c$ is a rational function with a numerator of degree $p-1$ with vector coefficients, and a scalar denominator of degree $p-1$. Extrapolation methods following this analysis were given in [5]. The idea is to compute the vector $\mathbf{r}_c$ for various values of c, and to interpolate them by a vector rational function with a much smaller degree $k \leq p-1$, and then to compute this rational function at a point outside the interval containing the values of c used before ($c = 0.85$, or $c = 1$, or any other value of c close to 1).

Although, in our extrapolation procedures, the vector $\mathbf{r}_c$ has to be computed for different values of the parameter c, it is very important to notice that the power method has not to be restarted for each value of c. The total number of iterations needed by our procedures is the one required for the highest value of c, and no additional iteration is needed; see [1] and [2, Prop. 8 and 9].

We will now discuss such extrapolation procedures. More details about these procedures can be found in [3], where numerical experiments are also reported.

2 Vector rational extrapolation

Let us describe in more details an algorithm for vector rational extrapolation which was first given in [5].

We begin by interpolating the vectors $\mathbf{r}_c \in \mathbb{R}^p$ corresponding to several values of the parameter c by the vector rational function

$$\mathbf{p}(c) = \frac{\mathbf{P}_k(c)}{Q_k(c)}, \tag{1}$$

where $\mathbf{P}_k$ and Q_k are polynomials of degree $k \leq p-1$. The coefficients of $\mathbf{P}_k$ are vectors, while those of Q_k are scalars. Then, an approximate value of $\mathbf{r}_c$, for an arbitrary value of c (in general outside the interval containing the interpolation points, thus the name of the procedure) will be given by $\mathbf{p}(c)$.

Following an idea introduced in [4], the coefficients of $\mathbf{P}_k$ and Q_k are obtained by solving the interpolation problem

$$Q_k(c_i)\mathbf{p}_i = \mathbf{P}_k(c_i), \qquad i = 0, \ldots, k, \tag{2}$$

with $\mathbf{p}_i = \mathbf{r}_{c_i}$, and the c_i's distinct points in $]0, 1[$.

The polynomials $\mathbf{P}_k$ and Q_k are given by the Lagrange's interpolation formula

$$\begin{aligned} \mathbf{P}_k(c) &= \sum_{i=0}^{k} L_i(c)\mathbf{P}_k(c_i) \\ Q_k(c) &= \sum_{i=0}^{k} L_i(c)Q_k(c_i) \end{aligned} \tag{3}$$

with

$$L_i(c) = \prod_{\substack{j=0 \\ j\neq i}}^{k} \frac{c - c_j}{c_i - c_j}, \quad i = 0, \ldots, k.$$

Thus, from (2),

$$\mathbf{P}_k(c) = \sum_{i=0}^{k} L_i(c)Q_k(c_i)\mathbf{p}_i. \tag{4}$$

Let us now show how to compute $Q_k(c_0), \ldots, Q_k(c_k)$. We assume that, for $c^* \neq c_i$, $i = 0, \ldots, k$, the vector $\mathbf{r}_{c^*}$ is known. Following (1) and (4), we will approximate it by

$$\mathbf{p}(c^*) = \sum_{i=0}^{k} L_i(c^*)a_i(c^*)\mathbf{p}_i, \tag{5}$$

with $a_i(c^*) = Q_k(c_i)/Q_k(c^*)$.

Let $\mathbf{s}_0, \ldots, \mathbf{s}_k$ be $k+1$ linearly independent vectors. After taking their scalar products with the vector $\mathbf{p}(c^*)$, given by (5), and with the vector $\mathbf{r}_{c^*}$, we will look for $a_0(c^*), \ldots, a_k(c^*)$ solution of the system of $k+1$ linear equations

$$\sum_{i=0}^{k} (\mathbf{p}_i, \mathbf{s}_j)L_i(c^*)a_i(c^*) = (\mathbf{r}_{c^*}, \mathbf{s}_j), \qquad j = 0, \ldots, k. \tag{6}$$

Once the $a_i(c^*)$'s have been obtained as the solution of the system (6), the $Q_k(c_i)$'s could be computed. For that, it is necessary to know the value of $Q_k(c^*)$. But, as we will see now, it is even unnecessary to know these quantities.

Indeed, for an arbitrary value of c, we obtain an approximation of $\mathbf{r}_c$ as

$$\mathbf{p}(c) = \frac{\mathbf{P}_k(c)}{Q_k(c)} = \frac{\sum_{i=0}^k L_i(c) Q_k(c_i) \mathbf{p}_i}{\sum_{i=0}^k L_i(c) Q_k(c_i)}.$$

Dividing the numerator and the denominator by $Q_k(c^*)$ finally leads to the extrapolation formula

$$\mathbf{p}(c) = \frac{\sum_{i=0}^k L_i(c) a_i(c^*) \mathbf{p}_i}{\sum_{i=0}^k L_i(c) a_i(c^*)}. \tag{7}$$

From Formula (7), it is easy to see that $\mathbf{p}(c_j) = \mathbf{p}_j$ for $j = 0, \ldots, k$, and that, in general, $\mathbf{p}(c^*) \neq \mathbf{r}_{c^*}$.

When $k = p-1$, $\mathbf{p}(c^*) = \mathbf{r}_{c^*}$, and, by a uniqueness argument, it follows that, for all c, $\mathbf{p}(c) = \mathbf{r}_c$.

We see that the computation of $\mathbf{p}(c)$ by our extrapolation method needs the knowledge of $\mathbf{r}_c$ for $k+2$ distinct values of c, namely $c_0, \ldots, c_k$ and c^*.

The complete vector rational extrapolation procedure is as follows

1. Choose $k+2$ distinct values of c: $c_0, \ldots, c_k$ and c^*.
2. Compute $\mathbf{p}_i = \mathbf{r}_{c_i}$ for $i = 0, \ldots, k$, and $\mathbf{r}_{c^*}$.
3. Choose $k+1$ linearly independent vectors $\mathbf{s}_0, \ldots, \mathbf{s}_k$, or take $\mathbf{s}_i = \mathbf{p}_i$ for $i = 0, \ldots, k$.
4. Solve the system (6), and compute the unknowns $a_0(c^*), \ldots, a_k(c^*)$.
5. Compute an approximation of $\mathbf{r}_c$ by (7).

3 A simpler vector rational extrapolation

Let us now consider a vector rational extrapolation method where the extrapolating function has the

$$\mathbf{p}(c) = \mathbf{y} + (1-c)\frac{1}{1-c\lambda}\mathbf{z}. \tag{8}$$

The two unknown vectors $\mathbf{y}$ and $\mathbf{z}$, and the unknown scalar λ will be computed by an interpolation procedure needing only 3 values of c.

As above, let $\mathbf{p}_i = \mathbf{r}_{c_i}$, and let the c_i's be distinct values in $]0, 1[$. We consider the interpolation condition

$$\mathbf{p}_i = \mathbf{y} + \frac{1-c_i}{1-c_i\lambda}\mathbf{z}.$$

The difference $\mathbf{p}_i - \mathbf{p}_j$ eliminates $\mathbf{y}$, and we have

$$\mathbf{p}_i - \mathbf{p}_j = \frac{(c_j - c_i)(1-\lambda)}{(1-c_i\lambda)(1-c_j\lambda)}\mathbf{z}.$$

We now need to compute the scalar λ and the vector $\mathbf{z}$. Let $\mathbf{q}$ be a vector so that the scalar products $(\mathbf{p}_i - \mathbf{p}_j, \mathbf{q})$ and $(\mathbf{p}_k - \mathbf{p}_j, \mathbf{q})$ are different from zero. We have

$$r_{ijk} = \frac{(\mathbf{p}_i - \mathbf{p}_j, \mathbf{q})}{(\mathbf{p}_k - \mathbf{p}_j, \mathbf{q})} = \frac{c_j - c_i}{c_j - c_k}\frac{1 - c_k\lambda}{1 - c_i\lambda},$$

which gives

$$\lambda = \frac{r_{ijk}(c_j - c_k) - (c_j - c_i)}{c_i r_{ijk}(c_j - c_k) - c_k(c_j - c_i)}. \tag{9}$$

Then $\mathbf{z}$ follows

$$\mathbf{z} = \frac{(1-c_i\lambda)(1-c_j\lambda)}{(c_j - c_i)(1-\lambda)}(\mathbf{p}_i - \mathbf{p}_j). \tag{10}$$

Finally, $\mathbf{y}$ is given by

$$\mathbf{y} = \mathbf{p}(1) = \mathbf{p}_i - \frac{1-c_i}{1-c_i\lambda}\mathbf{z}. \tag{11}$$

Thus, from the expressions (9), (10), and (11), Formula (8) leads to the rational vector extrapolation procedure (8), that is $\mathbf{p}(c) \simeq \mathbf{r}_c$.

4 A minimization procedure

Any scalar combination of different vectors $\mathbf{p}_i = \mathbf{r}_{c_i}$ can be considered as an extrapolation procedure (indeed, compare with (7)). So, we will now build an approximation $\mathbf{p}(c)$ of $\mathbf{r}_c$ of the form

$$\mathbf{p}(c) = (1-\alpha)\mathbf{p}_0 + \alpha\mathbf{p}_1 = \mathbf{p}_0 + \alpha(\mathbf{p}_1 - \mathbf{p}_0),$$

where the parameter α is chosen so that the euclidean norm of the vector $P_c^T\mathbf{p}(c) - \mathbf{p}(c)$ is minimum, that is

$$\alpha = -\frac{(P_c^T(\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{p}_1 - \mathbf{p}_0), P_c^T\mathbf{p}_0 - \mathbf{p}_0)}{\|P_c^T(\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{p}_1 - \mathbf{p}_0)\|^2}.$$

Let us mention that the products $P_c^T\mathbf{p}_i$ are cheap and easy to compute [2,7,8], and only two of them are required in this procedure.

Obviously this strategy could be extended to a more general form of minimization where

$$\mathbf{p}(c) = \alpha_0\mathbf{p}_0 + \cdots + \alpha_k\mathbf{p}_k \quad \mathit{with} \quad \alpha_0 + \cdots + \alpha_k = 1.$$

References

1. P. Boldi, M. Santini, S. Vigna, PageRank as a function of the damping factor, in *Proceedings of the 14th International World Wide Web Conference*, ACM Press, New York, 2005, pp. 557–566.
2. C. Brezinski, M. Redivo Zaglia, The PageRank vector: properties, computation, approximation, and acceleration, SIAM J. Matrix Anal. Appl., 28 (2006) 551–575.
3. C. Brezinski, M. Redivo Zaglia, Rational extrapolation for the PageRank vector, Math. Comput., to appear.
4. C. Brezinski, M. Redivo Zaglia, G. Rodriguez, S. Seatzu, Extrapolation techniques for ill–conditioned linear systems, Numer. Math., 81 (1998) 1–29.
5. C. Brezinski, M. Redivo Zaglia, S. Serra–Capizzano, Extrapolation methods for PageRank computations, C.R. Math. Acad. Sci. Paris, 340 (2005) 393–397.
6. R.A. Horn, S. Serra–Capizzano, A general setting for the parametric Google matrix, Internet Math., to appear.
7. S.D. Kamvar, T.H. Haveliwala, C.D. Manning, G.H. Golub, Extrapolations methods for accelerating PageRank computations, in *Proceedings of the 12th International World Wide Web Conference*, ACM Press, New York, 2003, pp. 261–270.
8. A.N. Langville, C.D. Meyer, *Google's PageRank and Beyond. The Science of Search Engine Rankings*, Princeton University Press, Princeton and Oxford, 2006.
9. S. Serra–Capizzano, Jordan canonical form of the Google matrix: a potential contribution to the PageRank computation, SIAM J.Matrix Anal. Appl., 27 (2005) 305–312.
10. S. Serra–Capizzano, Google pageranking problem: the model and the analysis, this seminar.