PageRank Computation and the Structure of the Web: Experiments and Algorithms

Arvind Arasu
Computer Science Department, Stanford University, CA 94305
arvinda@cs.stanford.edu

Jasmine Novak, Andrew Tomkins & John Tomlin
IBM Almaden Research Center, San Jose, CA 95120
jnovak@us.ibm.com, {tomkins,tomlin}@almaden.ibm.com

ABSTRACT

We describe some computational experiments carried out with variants of the "PageRank" model due to Page et al., with particular reference to the small and large scale structure of the Web.

Keywords

World Wide Web, PageRank, Linear Algebra, Sparse Matrices, BowTie Theory

1 INTRODUCTION

This poster summarizes the results of three sets of experiments[1] carried out with "PageRank" computation. The first set examines the convergence of the standard method, and relates it to the smaller scale structure of the Web. The second examines an alternative approach to computing PageRank which gives much faster convergence, at the cost of an additional sort. The third topic is the exploitation of the larger scale structure of the Web to partition the computation

2 MATHEMATICAL FORMULATION

We treat the HTML "points-to" relation on Web pages as a directed graph $G = (V, E)$, where the set V of vertices consists of the n pages, and the set E of directed edges (i, j), which exist iff page i has a hyperlink to page j. In practice this graph is usually pruned to remove self-loops and other forms of "spam". For convenience, we define d_i as the *out-degree* of page i (the number of hyperlinks on page i). We also define a *strongly connected component* of G to be a subset $V' \subset V$ of the vertices such that for all pairs of pages $i, j \in V'$, there exists a directed path from i to j. Finally, we will refer to a largest strongly connected component of a graph as *the SCC*.

PageRank is a static ranking of Web pages [5], used at the core of the Google search engine (http://www.google.com). In "ideal" form page i is assigned rank x_i as a function of the rank of the pages which point to it:

$$x_i = \sum_{(j,i) \in E} d_j^{-1} x_j \tag{1}$$

This recursive definition gives each page a fraction of the rank of each page pointing to it—inversely weighted by the number of links out of that page. We may write this in matrix form as:

$$x = Ax \tag{2}$$

where $a_{ij} = d_{ij}^{-1}$ if $(j, i) \in E$ and zero otherwise.

In practice many pages have no in-links and the eigenvector of (2) is mostly zero. To get around this difficulty, we may use an "actual model":

$$x_i = (1 - \alpha) + \alpha \sum_{(j,i) \in E} d_j^{-1} x_j \quad \forall i \tag{3}$$

or in matrix terms:

$$x = (1 - \alpha)e + \alpha A x \tag{4}$$

where e is the vector of all 1's, and α $(0 < \alpha < 1)$ is a parameter. If we scale x so that $e^T x = n$ this is equivalent to:

$$x = [(1 - \alpha)\frac{ee^T}{n} + \alpha A]x \tag{5}$$

Unless stated otherwise we use a value of 0.9 for α, but others report using a value of 0.85. This modification clearly overcomes the problem of identically zero PageRank—we may think of (3) as "seeding" each page with a rank of $(1 - \alpha)$.

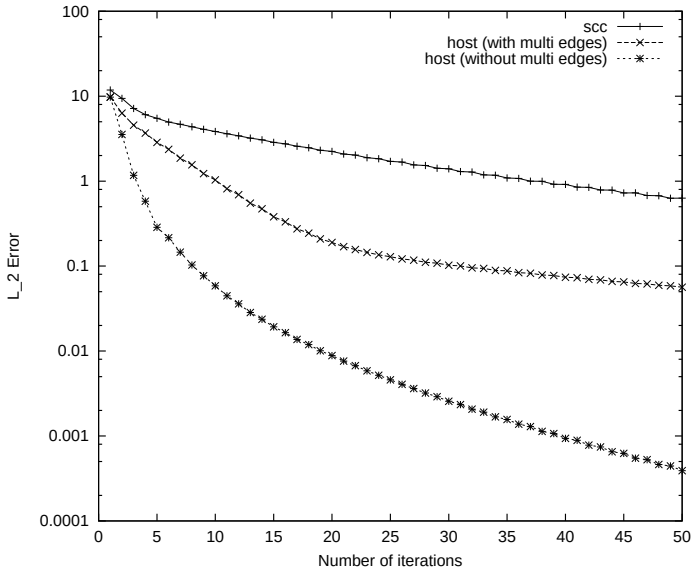

Figure 1.

Convergence of Host graph vs Web graph

3 POWER ITERATION AND CONVERGENCE

Power iteration[3] is the most straightforward technique for computing the principal eigenvector of a matrix. Elementary theory tells us that as α is reduced from 1 the gap between the principal and second eigenvalue will increase, and power iteration will converge more quickly. This is commonly observed. What is not commonly observed is that convergence may depend on the structure of the graph. We can naturally partition the graph into links inside Web sites, and links across Web sites. We consider the convergence of power iteration on the *host graph*, in which each vertex corresponds to an entire Web site. Any edge between pages in different sites is represented as an edge between the corresponding site vertices in the host graph. We consider two variants: in the first we allow multiple edges between sites if multiple pages on one site contained links to the same destination site. In the second, we collapse all such hyperlinks into a single edge. Figure 1 shows the convergence of power iteration for these two variants of the hostgraph against the convergence for the entire SCC for the Stanford WebBase graph[4].

Note that the normalized error measures (residuals) in Figure 1 are plotted in log scale. Clearly, residuals are orders of magnitude smaller for the host graph than in the whole SCC. Furthermore, we were able to compute numerically the second eigenvalue of the host graph, which is 0.9896. Thus, the eigenvalue gap is larger than 1%, implying quite rapid convergence.

On the other hand, the eigenvalue gap for the SCC was too small for us to complete computation of the second eigenvalue in reasonable time.

We conclude that connectivity between Web sites is strong and robust. However, once we shift from the granularity of the host graph back to the granularity of the Web graph itself, the complex structure within individual sites causes convergence to suffer.

4 EIGENSYSTEMS OR EQUATIONS?

Instead of computing an eigenvector from (5), we now consider the completely equivalent problem of solving the set of equations (4), which we can write more conventionally as:

$$(I - \alpha A)x = (1 - \alpha)e \tag{6}$$

The simplest iterative method—Jacobi iteration—for solving (6), requiring us to compute at iteration k

$$x_i^{(k+1)} = (1 - \alpha) + \alpha \sum_{(j,i) \in E} a_{ij} x_j^{(k)} \quad \forall i, \tag{7}$$

turns out to produce the same sequence of $x^{(k)}$ as power iteration carried out on (5). The next step is to use the *Gauss-Seidel* iteration:

$$x_i^{(k+1)} = (1 - \alpha) + \alpha \sum_{j<i} a_{ij} x_j^{(k+1)} + \alpha \sum_{j>i} a_{ij} x_j^{(k)} \quad \forall i \tag{8}$$

which uses the most recent values $x_j^{(k+1)}$ wherever possible. In Figure 2 we plot the convergence of the two methods using the sample SCC of the previous section. In this experiment we first computed the eigenvector (for $\alpha = 0.9$) by other means to full single precision accuracy. We then ran the Power Iteration and Gauss-Seidel methods, at each step computing the 2-norm of the deviation of the current iterate $x^{(k)}$ from the known solution.

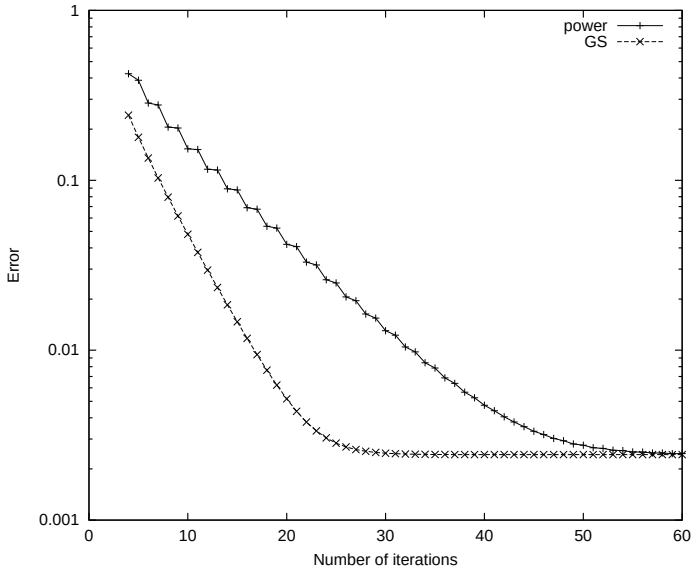

Figure 2.

Logarithmic Convergence of PageRank for Power Iteration and Gauss-Seidel with $\alpha = 0.9$

The Gauss-Seidel method clearly converges much faster than the Power/Jacobi methods. The only caveat is that Gauss-Siedel requires us to sort A by row, while Power/Jacobi can use it in any order. Experiments indicate that the improved convergence more than makes up for the sort. Whole classes of iterative methods[3] remain untried for this problem.

5 LARGE SCALE STRUCTURE OF THE WEB AND ITS IMPLICATIONS

So far we have considered only small-scale structure of the Web graph and its associated matrices. When we consider the large-scale structure of the Web, the arguments for using an equation-solving approach become even stronger. As is now well known, the graph structure of the Web may be characterized by the "bow tie" of Figure 3 (see Broder et al[2])

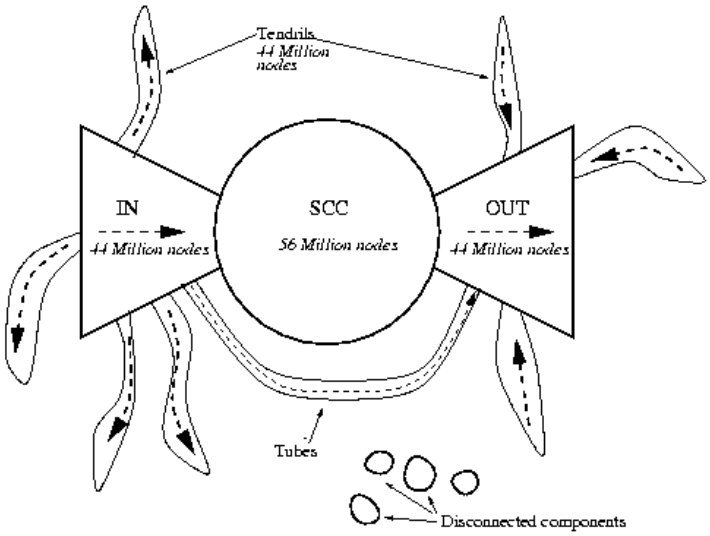

Figure 3.

Bowtie Structure of the Web

Ignoring the disconnected components, which can be dealt with separately, it is easy to see that this structure corresponds to a special case of a "block upper triangular" permutation of our coefficient matrix:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1N} \\ & A_{22} & A_{23} & \dots & A_{2N} \\ & & A_{33} & \dots & A_{3N} \\ & & & \ddots & \vdots \\ & & & & A_{NN} \end{pmatrix} \tag{9}$$

and indeed we may view the bow tie as a visual model which captures some, but by no means all, of the structure which can be extracted from the matrix A.

The structure (9), where each diagonal block is irreducible , may be extracted by Depth First Search (DFS)[6]. The largest diagonal block corresponds to the SCC, and this is the only really efficient way that we know of to find it. We may therefore consider the finding of the entire detailed block structure (9), a linear complexity process, equivalent to the work involved in just the first step of finding the SCC for the bow tie.

When an equation solving method is used we may conformably partition x with the matrix so that

$$x^T = (x_1^T, x_2^T, x_3^T, \dots, x_N^T)$$

and equation (6) can be solved via a sequence of smaller problems of the general form:

$$(I - \alpha A_{ii})x_i = (1-\alpha)e + \alpha \sum_{j=i+1}^{N} A_{ij}x_j \tag{10}$$

for $i = N, \dots, 2, 1$.

Strong Component Size	Number of This Size
8,188,380	1
108,131	1
30,000 - 40,000	7
20,000 - 29,000	34
10,000 - 19,999	68
1,000 - 9,999	808
2 - 1,000	221,149
1	4,771,701

Table 1.

Size and Number of Strong Components

Generally speaking, we expect the solution of these subproblems to take fewer iterations individually than is required for the huge matrix A, leading to less total arithmetic. The other payoff, perhaps *the* payoff, is in reduced I/O.

When we solve the sequence of sets of equations (10) we expect the largest of the subproblems to be of the order of the SCC, and many of the A_{ii} matrices may now be able to fit in memory. To illustrate this, we show in Table 1 the size of the strong components for the Web graph of an experimental (and incomplete) crawl of the IBM intranet, with 20,813,497 pages and 333,203,246 links. We see that there is a rapid drop-off in size from the largest SCC of about 8 million pages to many much smaller components. This suggests a strategy of aggregating adjacent components (i.e. diagonal blocks in (9)) to produce many A_{ii} submatrices which fit in main memory—perhaps for all but the largest SCC. In this situation the total amount of buffering (I/O) required is considerably reduced.

6 CONCLUSION

Our limited experiments suggest that PageRank computation can be significantly speeded up. We await the results of larger crawls being carried out by our colleagues to test the methods we have discussed here on a full Web scale.

References

[1] A. Arasu, J. Novak, A. Tomkins and J. Tomlin, "PageRank Computation and the Structure of the Web: Experiments and Algorithms", Technical Report, IBM Almaden Research Center, Nov. 2001.

[2] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins and J. Wiener, "Graph Structure in the Web", Proc. WWW9 conference, 309-320, May 2000. See also: http://www9.org/w9cdrom/160/160.html

[3] G.H. Golub and C.F. Van Loan, *Matrix Computations (3rd edition)*, Johns Hopkins University Press, Baltimore and London (1996).

[4] J. Hirai, S. Raghavan, H. Garcia-Molina and A. Paepcke, "WebBase: A Repository of Web Pages", Proc. WWW9 conference, May 2000. See also: http://www9.org/w9cdrom/296/296.html

[5] L. Page, S. Brin, R. Motwani and T. Winograd "The PageRank Citation Ranking: Bringing Order to the Web", Stanford Digital Library working paper SIDL-WP-1999-0120 (version of 11/11/1999). See: http://www-diglib.stanford.edu/cgi-bin/get/SIDL-WP-1999-0120

[6] R.E. Tarjan, "Depth-first Search and Linear Graph Algorithms", *SIAM J. Computing*, **1**, pp 146-160 (1972).