On the Feasibility of Low-rank Approximation for Personalized PageRank[*]

András A. Benczúr Károly Csalogány Tamás Sarlós

Data Mining and Web Search Group, Informatics Laboratory
Computer and Automation Research Institute
Hungarian Academy of Sciences
and Eötvös University, Budapest

{benczur,cskaresz,stamas}@ilab.sztaki.hu

ABSTRACT

Personalized PageRank expresses backlink-based page quality around user-selected pages in a similar way to PageRank over the entire Web. Algorithms for computing personalized PageRank on the fly are either limited to a restricted choice of page selection or believed to behave well only on sparser regions of the Web. In this paper we show the feasibility of computing personalized PageRank by a $k < 1000$ low-rank approximation of the PageRank transition matrix; by our algorithm we may compute an approximate personalized PageRank by multiplying an $n \times k$, a $k \times n$ matrix and the n-dimensional personalization vector. Since low-rank approximations are accurate on dense regions, we hope that our technique will combine well with known algorithms.

Categories and Subject Descriptors

F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Performance, Experimentation, Measurement

Keywords

Personalized PageRank, Singular Value Decomposition, link analysis, low-rank approximation, Web information retrieval

1. INTRODUCTION

Topic sensitive or personalized ranking appears since the beginning of the success story of Google's PageRank [6]. Topic sensitivity can be achieved by precomputing modified PageRank measures over the entire Web [4]; unfortunately the naive computation of PPR requires a power iteration algorithm over the entire web graph, making the procedure infeasible for an on-line query response service.

All known approaches for computing personalized PageRank on the fly for a user query impose limitations. Some of them [5] restrict personalization to a limited set of a few tens of thousands of Web pages; other methods rely on fingerprints [2] that behave well only on relatively sparse portions of the Web.

We show the feasibility of low-rank approximation to the transition matrix of the Web graph for computing personalized PageRank. We use the linear time SVD algorithm of [1] by selecting a random sample of approximately 10^5 columns and compute the first few hundred singular vectors. We may only personalize on nodes well-connected into the sample; we argue the procedure works well in dense regions.

Our algorithm consists of a precomputation step to compute an $n \times k$ and a $k \times n$ matrix with the help of low-rank approximation. There is no need to store either matrices explicitly since they are expressible as products of small dense and large sparse matrices. Then we may personalize for vector x by multiplying the $n \times k$ matrix with the product of the $k \times n$ matrix and x. The last multiplication is inexpensive for a sparse x that we typically use for personalization; the first multiplication however needs either RAM-resident data or and edge scan over the Web graph.

2. THE ALGORITHM

Let M denote the transition matrix corresponding to the Web graph. The PageRank vector $\mathrm{ppr}(x)$ personalized on a distribution x on Web pages is the solution of $\mathrm{ppr}(x) = (1-d)x + d \cdot M^T \mathrm{ppr}(x)$ and can be obtained as

$$\mathrm{ppr}_M(x) = (1-d)(I - d \cdot M^T)^{-1} x \qquad (1)$$

Since matrix inversion at the scale of a Web transition matrix is infeasible, we will use low-rank approximation to solve the problem. Unfortunately it is know that low-rank approximation cannot significantly speed up general inverse computation as smallest singular values are involved [3]. Instead we use power iteration

$$\mathrm{ppr}_M(x) = (1-d)(x + \sum_{i=1}^{\infty} d^i (M^T)^i \cdot x). \qquad (2)$$

and replace M by a low-rank approximation $\tilde{M}$. If the error in approximating M^i by $\tilde{M}^i$ grows slower than d^{-i}, the total error will remain bounded; giving theoretic guarantees is however beyond the scope of this paper.

[*]Support from NKFP-2/0017/2002 project Data Riddle, ETIK and OTKA T042706

Algorithmically we proceed by the linear time SVD algorithm of [1]. We sample the columns of M with probability p_v for column v proportional to ℓ_2 length. We form matrix C by selecting c columns and multiplying each by $1/\sqrt{cp_v}$. Compute the SVD of $C = H\Sigma Y^T$; let H_k be a $n \times k$ matrix consisting of the first k singular vectors. By Theorem 4 of [1] $\tilde{M} = H_k H_k^T M$ is a low-rank approximation of M, hence

$$\begin{aligned}
\tilde{\text{ppr}}(x) &= (1-d)(x + \sum_{i=1}^{\infty} d^i (M^T H_k H_k^T)^i \cdot x) \\
&= (1-d)(x + dM^T H_k \cdot \big(\sum_{i=0}^{\infty} d^i (H_k^T M^T H_k)^i\big) \cdot H_k^T x) \\
&= (1-d)x + dM^T H_k \cdot \big(\text{ppr}_{H_k^T M^T H_k}(x)\big) \cdot H_k^T x.
\end{aligned}$$

Our algorithm hence computes the PPR over the $k \times k$ matrix $H_k^T M^T H_k$, multiplied by two $n \times k$ matrices from left and right. We may apply the power iteration (2) or use our matrix inversion algorithm of choice as in (1).

Since random sampling is involved, our algorithm may only personalize to nodes that can reach the sample on outlinks and positive PPR values are assigned only to nodes reachable in two steps from the sample. To see, note that $C = H\Sigma Y^T$ implies $H^T = \Sigma^{-1} Y^T C^T$ and hence the last term $H_k^T x$ can further be reduced to an initial computation of $C^T \cdot x$. This vector however depends only on values x_u such that for some v $C_{uv} \neq 0$, i.e. if u has an edge into node v such that column v is selected into the sample. A similar argument applies to H_k on the left hand side; finally the result is multiplied by another M^T from the left, meaning another step along the edges of the graph.

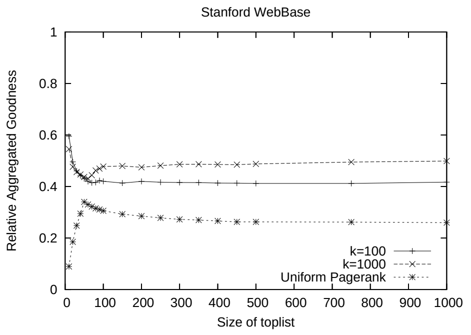
We show how sampling affects the algorithm. Columns are selected based on ℓ_2 length that will imply our algorithm behave well on dense regions. Column v is selected with higher probability if the squared sum of $1/\text{outdeg}(u)$ for all u with edge into v is large. Hence we select the node if the indegree of v is large and in particular if incoming edges are from nodes with small out-degree. Hence the base of links into the sample will cover a large portion of the Web.

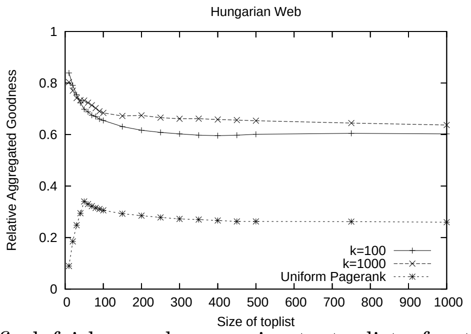
3. EMPIRICAL EVALUATION

We computed approximate PPR for the 80-million node Stanford Webbase graph and a 5-million node graph of the Hungarian Web (`.hu`) with $c = 10,000$; in order to measure the effect of k we used the choices of $k = 100$ and 1000. The longest precomputation took two days over a single 2.0 GHz AMD Opteron with 4 GB RAM. Uniform PageRank has correlation 0.52 and 0.60, rank correlation 0.28 and 0.44, respectively, with the approximate values.

We measured the average similarity of the exact and approximate personalized PageRank vectors in both cases for the top 1000 PageRank pages. We used two additional selection criteria: we require at least one link into the sample (restriction: otherwise we get a null PPR vector); and discard outdegree zero vertices (fairness: their PPR toplists are very easily reproduced, resulting in very high average goodness). For the Hungarian graph 60% and for the Stanford graph 25% of the pages is included in the measurement.

Due to its robustness to error in very small PPR values we chose the aggregate goodness measure of [7] that measures the total real rank of the first k elements output by our algorithm. *Relative aggregated goodness* calculates the sum of exact PPR values in the approximate toplist compared to the maximum value achievable. We average these values over pages in our sample. As baseline we use the (uniform) PageRank as approximation to all PPR vectors.





We find fairly good approximate toplists for the small Hungarian graph; here an order of magnitude increase in k makes little difference. The 80 million node Stanford graph performs worse though still significantly above the uniform PageRank baseline. Here even larger values of k are needed together with larger sample size in order to increase both the low value 25% of pages we could personalize on and the list of pages that may appear with positive PPR value.

4. CONCLUSIONS AND FUTURE WORK

Our experiments show that dimension reduction may be used as an ingredient in computing personalized PageRank. We believe that the combination with [5, 2] makes personalization feasible in large-scale search engines.

5. REFERENCES

[1] P. Drineas, M. W. Mahoney, and R. Kannan. Fast monte carlo algorithms for matrices II. Technical Report YALEU/DCS/TR-1270, Yale University, 2004.

[2] D. Fogaras and B. Rácz. Towards scaling fully personalized pagerank. In *WAW*, pages 105–117, 2004.

[3] H. Guo. IRR: An algorithm for computing the smallest singular value of large-scale matrices. *International J. of Computer Math.*, 77(1):89–104, 2001.

[4] T. H. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the 11th World Wide Web Conference (WWW)*, Honolulu, Hawaii, 2002.

[5] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th World Wide Web Conference (WWW)*, pages 271–279. ACM Press, 2003.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[7] P. K. C. Singitham, M. S. Mahabhashyam, and P. Raghavan. Efficiency-quality tradeoffs for vector score aggregation. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, 2004.