Incremental Page Rank Computation on Evolving Graphs*

Prasanna Desikan
Dept. of Computer Science
University of Minnesota
Minneapolis, MN 55455
USA
desikan@cs.umn.edu

Nishith Pathak+
Dept. of Computer Science
Indian Institute of Tech.
New Delhi 110016
INDIA
npathak@cs.umn.edu

Jaideep Srivastava
Dept. of Computer Science
University of Minnesota
Minneapolis, MN 55455
USA
srivasta@cs.umn.edu

Vipin Kumar
Dept. of Computer Science
University of Minnesota
Minneapolis, MN 55455
USA
kumar@cs.umn.edu

ABSTRACT
In this paper, we propose a method to incrementally compute PageRank for a large graph that is evolving. Our approach is quite general, and can be used to incrementally compute (on evolving graphs) any metric that satisfies the first order Markov property.

Categories and Subject Descriptors
I.0 [Computing Methodologies]: General

General Terms: Algorithms, Measurement, Performance, Reliability, Security, Standardization, Theory, Verification.

Keywords: PageRank, Algorithm, Evolving Graphs, Web Mining.

1. INTRODUCTION
The importance of link analysis on the Web graph has gained significant prominence after the advent of Google. The key observation is that a hyperlink from a source page to a destination page serves as an endorsement of the destination page by the (author of the) source page on some topic. Link analysis techniques have adopted different knowledge models for the measures developed for various applications on the Web. Kleinberg's Hubs and Authority is based on the observation that the Web graph has a number of bipartite cores, while Google's PageRank is based on the observation that a user's browsing of the Web can be approximated as a first order markov model. Giles, et al have used network flow models to identify web communities. Thus, a variety of models have been used to measure different properties of the Web Graph at a given time instance. Success of Google has signified the importance of Pagerank and has also led to a variety of modifications and improvisations of the basic PageRank metric. Another important dimension of Web mining is the evolution of the Web graph. The Web is changing over time, and so is the users' interaction on (and with) the Web, suggesting the need to study and develop models for the evolving Web Content, Web Structure and Web Usage. The study of such evolution of the Web would require computing the various existing measures for the Web graph at different time instances. A straightforward approach would be to compute these measures for the whole Web Graph at each time instance. However, given the size of the Web graph, this is becoming increasingly infeasible. Furthermore, if the percent of nodes that change during a typical time interval when the Web is crawled by search engines is not high, a large portion of the computation cost may be wasted on re-computing the scores for the unchanged portion. Hence, there is a need for computing metrics incrementally, to save on the computation costs.

In this paper, we describe an approach to compute PageRank in an incremental fashion. We exploit the underlying first order markov model property of the metric, to partition the graph into two portions such one of them is unchanged since the last computation, and it has only outgoing edges to the other partition. Since there are no coming edges from the other partition, the distribution of PageRank values of the nodes in this partition will not be affected by the nodes in the other partition. The other partition is the rest of the graph, which has undergone changes since the last time the metric was computed.

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

2. PROPOSED APPROACH
In the proposed approach, we exploit the underlying first order Markov Model on which the computation of PageRank is based. It should be noted that PageRank of a page depends only on the pages that point to it and is independent of the outdegree of the page. The principle idea of our approach is to find a partition such that there are no incoming links from a partition, Q (includes all changed nodes) to a partition, P. In such a case the PageRank of the partition, Q is computed separately and later scaled and merged with the rest of the graph to get the actual PageRanks of vertices in Q. The scaling is done with respect to the number of vertices in partition, P- `n(P) to the total number of nodes in the whole graph, G –n(P UQ)=V. The PageRank of the partition Q is computed, taking the border vertices that belong to the partition P and have edges pointing to the vertices in partition Q. The PageRank values of partition P are obtained by simple scaling.

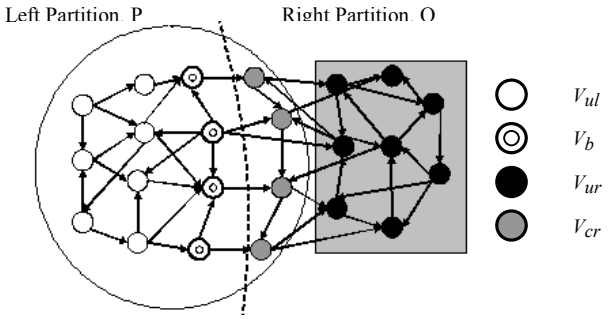

Figure 1. Graph Partitioning for incremental computation.

This basic idea of partitioning the Web graph, and computing the PageRanks for individual partitions and merging works extremely well when incrementally computing PageRank for a Web graph that has evolved over time. Given, the Web graphs at two consecutive time instances, we first determine the portion of the graph that has changed. A vertex is declared to be changed when a new edge added or deleted between the vertex and any other vertex belonging to the graph or if the weight of a node or an edge weight adjoined to that node has changed. Once the changed portion is defined, for each page we determine iteratively all the pages that will be affected by its PageRank. In this process, we include pages that remain unchanged but whose; PageRank gets affected due to the pages that have changed, in partition Q. The rest of the unchanged graph is in partition P.

The whole concept is illustrated in Figure 1. Let the graph at the new time be G(V,E), and

$v_b =$ Vertex on the border of the left partition from which there are only outgoing edges to the right partition.

v_{ul} = vertex on the left partition which remains unchanged

The set of unchanged vertices can be represented as,

$V_u = \{v_u, \forall v_u \in V\}$ where v_u is a vertex which has not changed.

$v_{ur} =$ Vertex on the right partition which remains unchanged, but whose PageRank is affected by vertices in the changed component.

$v_{cr} =$ Vertex on the right partition which has changed, or has been a new addition.

Therefore, the set of changed vertices can be represented as,

$V_c = \{v_{cr}, \forall v_{cr} \in V\}$

In order to compute PageRank incrementally, for every vertex in V_c, which is a set of changed vertices, perform a BFS to find out all vertices reachable from this set. The PageRank of these vertices will be affected by vertices in V_{cr}. These set of vertices can be denoted by the set,

$V_{ur} = \{v_{ur}, \forall v_{ur} \in V\}$

Similarly, the set of vertices v_b can be denoted as,

$V_b = \{v_b, \forall v_b \in V\}$

Hence the set of vertices whose PageRank has to be computed in the incremental approach corresponds to the partition Q described above, and can be denoted as,

$V_Q = V_c \cup V_{ur} \cup V_b$

Let an edge set, E_Q, be defined as set of edges,

$E_Q = \{e_{x,y} \mid x, y \in V_Q\}$, where $e_{x,y}$ represents a directed edge from vertex *x* to vertex *y*.

The set of partitioning edges can be defined as,

$E_{Part} = \{e_{x,y} \mid x \in V_P, y \in V_Q\}$,

The vertices in partition P can be defined as,

$V_P = V - V_Q + V_b$

And the edges that correspond to this partition can be defined as,

$E_P = \{e_{x,y} \mid x, y \in V_P\}$, where $e_{x,y}$ represents a directed edge from vertex *x* to vertex *y*.

Thus, the given graph $G(V,E)$ can be partitioned into two graphs namely, $G_P(V_P, E_P)$ and $G_Q(V_Q, E_Q)$.

Now, since we know that the graph $G_P(V_P, E_P)$ has remained unchanged from the previous time instance and the PageRank of vertices in this partition is not affected by the partition, $G_Q(V_Q, E_Q)$. Now a change in a node induces a change in the distribution of PageRank values for all its children and since all the nodes that are influenced by changes are already separated in the partition Q. The distribution of PageRank values for the nodes in partition G_P is going to be the same as it was for the corresponding nodes in the previous time instance G'. Thus the PageRank of the vertices in partition *P* could be calculated by simply scaling the scores from the previous time instance. And the scaling factor will be $n(G')/n(G)$, where G' is the graph at the previous time instance. And the PageRank for the partition, $G_Q(V_Q, E_Q)$ can be computed using the regular PageRank Algorithm and scaled for the size of the graph, G. Since the percent change in the structure of the Web is not high, the computation of the changed portion will be a smaller graph compared to the whole Web. We scale the PageRanks of nodes in V_b such that they correspond to the number of nodes for which the PageRank is actually computed. The finer details of the algorithm and border conditions can be found in the technical report [1].

3. EXPERIMENTAL RESULTS

Our experiments were performed on two different web sites- the Computer Science website and the Institute of Technology website at the University of Minnesota at different time intervals.We also simulated the focused crawling, by not considering the Web pages that have very low PageRank into our graph construction and PageRank Computation. This was to emulate the real world scenario where not all pages are crawled. We used the following approximate measure to compare the computational costs of our method versus the naïve method.

*Number of Times Faster = Num of Iterations(*PR*)/(*1 + *(fraction of changed portion)*Num of iterations(*IPR*))*

The summary of experimental results for Computer Science Website is provided in Figure 2. The significant improvement achieved using incremental computation can be seen from these results. More detailed results are provided in the technical report [1].

Computer Science	**Focussed**		**Unfocussed**	
Dates	% Change	Num Times	% Change	Num Times
July19 vs July 27th	53.14%	1.90053849	60.30%	1.86712307
July 27th vs July 29th	5.25%	9.88548051	5.57%	8.65916206
July19th vs 29th	58.35%	1.75566935	65.06%	1.74952617

4. CONCLUSIONS

In this paper we have provided an approach to compute PageRank incrementally for evolving graphs. The key observation is that evolution of the Web graph is slow, with large parts of it remaining unchanged. By carefully delineating the changed and unchanged portions and the dependence across them, it is possible to develop efficient algorithms for computing the PageRank metric incrementally.

5. REFERENCES

[1] P. Desikan, N. Pathak, J. Srivastava and V. Kumar "Incremental PageRank Computation on evolving graphs" AHPCRC Technical Report 2004-195.

* This work was been partially supported by the ARDA Agency under contract F30602-03-C-0243 and AHPCRC contract number DAAD19-01-2-0014. The content of the work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred.

+ The author is an undergraduate student from IIT Delhi, India. This work was done while he was a summer intern at University of Minnesota.