Modifications of Kleinberg's HITS Algorithm Using Matrix Exponentiation and Web Log Records

Joel C. Miller
Harvey Mudd College
Claremont, CA 91711
jomiller@math.hmc.edu

Gregory Rae
Harvey Mudd College
Claremont, CA 91711
grae@hmc.edu

Fred Schaefer
Harvey Mudd College
Claremont, CA 91711
fschaefe@hmc.edu

ABSTRACT

Kleinberg's HITS algorithm, a method of link analysis, uses the link structure of a network of webpages to assign authority and hub weights to each page. These weights are used to rank sources on a particular topic. We have found that certain tree-like web structures can lead the HITS algorithm to return either arbitrary or non-intuitive results. We give a characterization of these web structures. We present two modifications to the adjacency matrix input to the HITS algorithm. Exponentiated Input, our first modification, includes information not only on direct links but also on longer paths between pages. It resolves both limitations mentioned above. Usage Weighted Input, our second modification, weights links according to how often they were followed by users in a given time period; it incorporates user feedback without requiring direct user querying.

Keywords

Link analysis, HITS algorithm, Kleinberg algorithm, hubs, authorities, networks

1. INTRODUCTION

Kleinberg's HITS algorithm, "Hypertext Induced Topic Selection", is a standard algorithm of Link Analysis [3, 4]. It ranks web search results. The premise of the HITS algorithm is that a web page serves two purposes: to provide information and to provide links relevant to a topic. This gives two ways to categorize a web page. A web page is an *authority* on a topic if it provides good information, and it is a *hub* if it provides links to good authorities. The HITS algorithm is an iterative algorithm developed to quantify each page's value as a hub and an authority.

2. KLEINBERG'S HITS ALGORITHM

Consider a directed graph G on $[n]$ with adjacency matrix M. Let $\vec{h}_k$ be the vector whose i^{th} entry $h_k(i)$ is the hub weight assigned to the i^{th} node at iteration k. Similarly let $\vec{a}_k$ be the vector of authority weights. Initialize these vectors so that $h_0(i) = a_0(i) = 1/n$ for all i (other initializations can be used). In the k^{th} iteration, compute the new hub weight $h_k(i)$ by summing the authority weights of the nodes j to which node i points: set $h_k(i) = \sum_j a_{k-1}(j)$. Similarly, update the authority weights by setting $a_k(i) = \sum_j h_{k-1}(j)$, where the sum runs over the nodes j that point to node i. Then normalize so that $\sum_i a_k(i) = \sum_i h_k(i) = 1$. In linear algebra terms, we are computing $\vec{h}_k = \psi_k M \vec{a}_{k-1}$ and $\vec{a}_k = \phi_k M^T \vec{h}_{k-1}$, where ψ_k and ϕ_k are the normalization factors. Combining these formulas, we see that

$$\vec{h}_k = \psi_k \phi_{k-1} M M^T \vec{h}_{k-2}, \qquad \vec{a}_k = \phi_k \psi_{k-1} M^T M \vec{a}_{k-2}. \quad (1)$$

The eigenvalues of the real, non-negative, symmetric matrix $M^T M$ are real and non-negative, so iteration of the authority [resp. hub] weight formula in (1) converges to an eigenvector of the dominant eigenvalue of $M^T M$ [resp. MM^T].

2.1 Limitations of the HITS algorithm

The HITS algorithm does not always behave as expected. First, if the dominant eigenvalue of $M^T M$ is repeated, the HITS algorithm converges to an authority vector which is *not unique*, but depends on the initial seed $\vec{a}_0$. The authority vector can be any normalized vector in the dominant eigenvalue's eigenspace. For example, for a two-level reversed binary tree B whose edges point upwards *towards* the root, the eigenvalues of $M^T M$ are 2, 2, 2, 0, 0, 0, and 0. The authority weights for the three upper nodes can be any three positive numbers that sum to 1. Second, the HITS algorithm yields *zero* authority weights for apparently important nodes of certain graphs. For example, if a leaf is added at the left middle-level node of B, then both the hub and authority weights are zero for the root and for the right half of B. We call these limitations *non-uniqueness* and *nil-weighting*, respectively.

We have characterized the graphs G on which the HITS algorithm is non-unique or nil-weighted. Consider an undirected graph G' on $[n]$ where $\{i, j\}$ is an edge of G' if there is a k such that (k, i) and (k, j) are directed edges of G. The HITS algorithm is non-unique or nil-weighted on G if and only if there exist i, j with positive in-degree in G such that i and j are in distinct components of G'.

3. EXPONENTIATED INPUT TO HITS

The key idea is to replace the adjacency matrix used by the HITS algorithm with an *exponentiated matrix*, which

SIGIR'01, September 9-12, New Orleans, Louisiana, USA.

contains direct information on paths of length 2 or more in the graph. Namely, we replace M with the 'Taylor series' matrix

$$e^M - I = M + M^2/2! + M^3/3! + \cdots + M^m/m! + \cdots . \quad (2)$$

Recall that the number of paths of length m from node i to node j is given by the i,j^{th} entry of M^m. The scaling factors $1/m!$ assign less importance to longer paths, and could be varied somewhat without losing convergence.

Our version of the HITS algorithm with Exponentiated Input now updates the authority vector by multiplying by $(e^M - I)^T(e^M - I)$ instead of M^TM. We have proved that this modification *prevents* non-uniqueness and nil-weighting.

THEOREM 1. *If G is a weakly connected graph, then the matrix $(e^M - I)^T(e^M - I)$ has a simple dominant eigenvalue, whose eigenvector has only positive entries. Therefore the HITS algorithm with Exponentiated Input applied to G converges to a unique (normalized) eigenvector, with positive entries; in particular the algorithm cannot be non-unique or nil-weighted on weakly connected graphs.*

We prove Theorem 1 by showing that if G is weakly connected, then the matrix $(e^M - I)^T(e^M - I)$ cannot be written in block lower triangular form. (A directed graph is *weakly connected* if it is connected when the directions of the edges are ignored.) The Perron-Frobenius theorem [2, p.53] then implies that $(e^M - I)^T(e^M - I)$ has a simple dominant eigenvalue, and that the entries of the corresponding eigenvector are all positive.

For the two-level binary tree B, Exponentiated Input yields (unique!) authority weights of $1/2$ for the root, $1/4$ for the mid-level nodes, and 0 for the lowest level nodes, and hub weights of 0 for the root and $1/6$ for the other nodes. When one extra leaf is added at the lower level, Exponentiated Input yields (non-nil-weighted!) authority and hub weights which give the two-leaf middle-level node only a little less authority than the three-leaf one, and give all nodes except the root non-zero hub weight.

Note: We have found a sequence of graphs where the gap between the two largest eigenvalues of $(e^M - I)^T(e^M - I)$ seems to approach zero. Depending on arithmetic precision, this may mimic the non-uniqueness problem.

4. USAGE WEIGHTED INPUT TO HITS

Usage Weighted Input, our second modification of the adjacency matrix, replaces the adjacency matrix with a *link matrix* M' which weights connections between nodes (pages) based on the usage data from webserver logs of traffic on the website. We initialize the link matrix to 0, and then increment the link from node i to node j every time a user travels from i to j. (Notice that the resulting input matrix need not mirror the website structure, since some hyperlinks may never be followed, while users may navigate directly between pages that are not hyperlinked.) The effect is similar to that of *lifting* by gradient ascent [1] of the authority weight of a node, but does not require any direct querying of users.

When the HITS algorithm runs using M' in place of M, the information affecting the rate of change of the authority weight a^j_{k+1} of node j is the set of authority weights a^i_k of all nodes at the previous iteration, together with the link matrix entries m_{lj} for all links pointing to node j. Since these link matrix entries are larger if node j is visited more often, in each iteration the most frequently followed links play a larger role in determining new authority weights.

5. PRELIMINARY RESULTS

On the website `www.ehnc.com` (about 360 pages in December 1999), both modifications found more intuitively reasonable pages as the best hubs and authorities than the original HITS algorithm did. Only Usage Weighted Input ranked the root page as the best hub.

6. CONCLUSIONS

We have characterized the graphs on which the HITS algorithm produces ambiguous results and/or unreasonably assigns zero weights to parts of the graph. We have proved (Theorem 1) that our Exponential Input modification prevents these occurrences. In our preliminary experiments, Usage Weighted Input yielded even more satisfactory results than Exponentiated Input in finding the best hubs and authorities on a specific company website. Both input modifications were more satisfactory on this measure than the original binary adjacency matrix input.

One could also run the HITS algorithm on an exponentiated version of a usage weighted adjacency matrix for a website. This would combine the effectiveness of exponentiation, in incorporating indirect paths, with information on how users actually traversed the website. This analysis might be the most accurate indicator of the pages users have determined to be the best hubs and authorities on a website. We expect that this is where the advantages of the HITS algorithm with Exponentiated Input might be most visible, because the usage data is likely to have tree-like structure since users rarely follow the "back to homepage" links, and trees are among the graphs on which the HITS algorithm is badly behaved.

7. ACKNOWLEDGMENTS

This research was supported in part by the Harvey Mudd College Mathematics Clinic. We thank Joseph Sirosh of HNC Software Inc. for proposing and supporting the original project. For more information contact the authors or Joseph Sirosh (`sirosh@hnc.com`).

8. ADDITIONAL AUTHORS

Additional authors: Lesley A. Ward (Dept. of Mathematics, Harvey Mudd College, email: `ward@math.hmc.edu`), Thomas LoFaro (Dept. of Mathematics and Computer Science, Gustavus Adolphus College, St. Peter, MN 56082, email: `tlofaro@gustavus.edu`), and Ayman Farahat (HNC Software Inc., 5930 Cornerstone Court West, San Diego, CA 92121, email: `amfarahat1@home.com`).

9. REFERENCES

[1] H. Chang, D. Cohn, and A. McCallum. Creating customized authority lists. *Preprint*, 2000.

[2] F. Gantmacher. *Matrix Theory, v.2.* Chelsea, 1974.

[3] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, 1998.

[4] J. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604–632, 1999.