Finding Authorities and Hubs From Link Structures on the World Wide Web

Allan Borodin[*], Gareth O. Roberts[†], Jeffrey S. Rosenthal[‡], and Panayiotis Tsaparas[§]

ABSTRACT

Recently, there have been a number of algorithms proposed for analyzing hypertext link structure so as to determine the best "authorities" for a given topic or query. While such analysis is usually combined with content analysis, there is a sense in which some algorithms are deemed to be "more balanced" and others "more focused". We undertake a comparative study of hypertext link analysis algorithms. Guided by some experimental queries, we propose some formal criteria for evaluating and comparing link analysis algorithms.

Keywords

link analysis, web searching, hubs, authorities, SALSA, Kleinberg's algorithm, threshold, Bayesian

1. INTRODUCTION

In recent years, a number of papers [3, 8, 11, 9, 4] have considered the use of hypertext links to determine the value of different web pages. In particular, these papers consider the extent to which hypertext links between World Wide Web documents can be used to determine the relative authority values of these documents for various search queries.

We consider some of the previously published algorithms as well as introducing some new alternatives. One of our new algorithms is based on a Bayesian statistical approach as opposed to the more common algebraic/graph theoretic approach. While link analysis by itself cannot be expected to always provide reliable rankings, it is interesting to study various link analysis strategies in an attempt to understand inherent limitations, basic properties and "similarities" between the various methods. To this end, we offer definitions for several intuitive concepts relating to (link analysis) ranking algorithms and begin a study of these concepts.

We also provide some new (comparative) experimental studies of the performance of the various ranking algorithms. It can be seen that no method is completely safe from "topic drift", but some methods do seem to be more resistant than others. We shall see that certain methods have surprisingly similar rankings as observed in our experimental studies, however they cannot be said to be similar with regard to our formalization.

2. PREVIOUS ALGORITHMS

2.1 The PageRank Algorithm

One of the earliest and most commercially successful of the efforts to use hypertext link structures in web searching is the PageRank algorithm used by Brin and Page [3] in the Google search engine [7].

The page rank of a given web page i, denoted $PR(i)$, is defined recursively according to the equation

$$PR(i) = dD(i) + (1-d) \sum_{j \to i} [PR(j) \,/\, N(j)]\,,$$

where the sum is taken over all pages j which have a link to page i, $N(j)$ is the total number of links originating from page j, d is a number between 0 and 1, and D is a probability distribution (e.g. uniform) over all web pages.

Brin and Page [3] note that the value of $PR(i)$ is equivalent to the limiting fraction of time spent on page i by a random walk which proceeds at each step as follows: With probability d it jumps to a sample from the distribution $D(\cdot)$, and with probability $1-d$ it jumps uniformly at random to one of the pages linked from the current page. This idea is also used by Rafiei and Mendelzon [11] for computing the "reputation" of a page. Intuitively, the value of $PR(i)$ is a measure of the importance or authority of the web page i. This ranking is used as one component of the Google search engine, to help determine how to order the pages returned by a web search query.

[*] Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G4 and Gammasite, Hertzilya, Israel. E-mail: bor@cs.toronto.edu. Web: http://www.cs.utoronto.ca/DCS/People/Faculty/bor.html.

[†] Department of Mathematics and Statistics, Lancaster University, Lancaster, U.K. LA1 4YF. E-mail: g.o.roberts@lancaster.ac.uk. Web: http://www.maths.lancs.ac.uk/dept/people/robertsg.html.

[‡] Department of Statistics, University of Toronto, Toronto, Ontario, Canada M5S 3G3. Supported in part by NSERC of Canada. E-mail: jeff@math.toronto.edu. Web: http://markov.utstat.toronto.edu/jeff/.

[§] Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G4. E-mail: tsap@cs.toronto.edu. Web: http://www.cs.toronto.edu/∼tsap/.

WWW10, May 1-5, 2001, Hong Kong.
ACM 1-58113-348-0/01/0005.

2.2 Kleinberg's Algorithm

Independent of Brin and Page, Kleinberg [8] proposed a more refined notion for the importance of web pages. He suggested that web page importance should depend on the search query being performed. Furthermore, each page should have a separate "authority" rating (based on the links going *to* the page) and "hub" rating (based on the links going *from* the page). Kleinberg proposed first using a text-based web search engine (such as AltaVista [1]) to get a "Root Set" consisting of a short list of web pages relevant to a given query. Second, the Root Set is augmented by pages which link to pages in the Root Set, and also pages which are linked to pages in the Root Set, to obtain a larger 'Base Set" of web pages. If N is the number of pages in the final Base Set, then the data for Kleinberg's algorithm consists of an $N \times N$ adjacency matrix A, where $A_{ij} = 1$ if there are one or more hypertext links from page i to page j, otherwise $A_{ij} = 0$.

Kleinberg's algorithm assigns to each page i an authority weight a_i and a hub weight h_i. Let $\boldsymbol{a} = (a_1, a_2, \ldots, a_N)$ denote the vector of all authority weights, and $\boldsymbol{h} = (h_1, h_2, \ldots, h_N)$ the vector of all hub weights. Initially both authority and hub vectors are set to $\boldsymbol{u}= (1, 1, \ldots, 1)$. At each iteration the operations $\mathcal{I}$ ("in") and $\mathcal{O}$ ("out") are performed. The operation $\mathcal{I}$ sets the authority vector to $\boldsymbol{a} = A^T\ \boldsymbol{h}$. The operation $\mathcal{O}$ sets the hub vector to $\boldsymbol{h} = A\ \boldsymbol{a}$. A normalization step is then applied, so that the vectors $\boldsymbol{a}$ and $\boldsymbol{h}$ become unit vectors in some norm. Kleinberg proves that after a sufficient number of iterations the vectors $\boldsymbol{a}$ and $\boldsymbol{h}$ converge to the principal eigenvectors of the matrices A^TA and AA^T, respectively. The above normalization step may be performed in various ways. Indeed, *ratios* such as a_i/a_j will converge to the same value no matter how (or if) normalization is performed.

Kleinberg's Algorithm (and some of the other algorithms we are considering) converge naturally to their principal eigenvector, i.e. to the eigenvector of their transition matrix which corresponds to the largest eigenvalue. Kleinberg [8] makes an interesting (though non-precise) claim that the subsequent *non-principal* eigenvectors (or their positive and negative components) are sometimes representative of "sub-communities" of web pages. It is easy to construct simple examples which show that subsequent eigenvectors sometimes are, but sometimes are not, indicative of sub-communities; we present a few indicative such examples in the full version of this paper. The significance of non-principal eigenvectors is an important topic that we intend to pursue further.

2.3 The SALSA Algorithm

An alternative algorithm, SALSA, was proposed by Lempel and Moran [9]. Like Kleinberg's algorithm, SALSA starts with a similarly constructed Base Set. It then performs a random walk by alternately (a) going uniformly to one of the pages which links to the current page, and (b) going uniformly to one of the pages linked to by the current page. The authority weights are defined to be the stationary distribution of the two-step chain doing first step (a) and then (b), while the hub weights are defined to be the stationary distribution of the two-step chain doing first step (b) and then (a).

Formally, let $B(i) = \{k : k \to i\}$ denote the set of all nodes that point to i, that is, the nodes we can reach from i by following a link backwards, and let $F(i) = \{k : i \to k\}$ denote the set of all nodes that we can reach from i by following a forward link. The Markov Chain for the authorities has transition probabilities

$$P_a(i, j) = \sum_{k\,:\,k \in B(i) \cap B(j)} \frac{1}{|B(i)|} \frac{1}{|F(k)|}.$$

Assume for a moment that the Markov Chain is *irreducible*, that is, the underlying graph consists of a single connected component. The authors prove that the stationary distribution $\boldsymbol{a} = (a_1, a_2, ..., a_N)$ of the Markov Chain satisfies $a_i = |B(i)|\ /\ |B|$, where $B = \bigcup_i B(i)$ is the set of all (backward) links. Similarly, the Markov Chain for the hubs has transition probabilities

$$P_h(i, j) = \sum_{k\,:\,k \in F(i) \cap F(j)} \frac{1}{|F(i)|} \frac{1}{|B(k)|},$$

and the stationary distribution $\boldsymbol{h} = (h_1, h_2, ..., h_N)$ satisfies $h_i = |F(i)|\ /\ |F|$, where $F = \bigcup_i F(i)$ is the set of all (forward) links.

SALSA does not really have the same "mutually reinforcing structure" that Kleinberg's algorithm does. Indeed, since $a_i = |B(i)|/|B|$, the relative authority of site i *within a connected component* is determined from local links, not from the structure of the component. (See also the discussion of *locality* in Section 7.) We also note that in the special case of a single component, SALSA can be viewed as a one-step truncated version of Kleinberg's algorithm. That is, in the first iteration of Kleinberg's algorithm, if we perform the $\mathcal{I}$ operation first, the authority weights are set to $\boldsymbol{a} = A^T\ \boldsymbol{u}$, where $\boldsymbol{u}$ is the vector of all ones. If we normalize in the L_1 norm, then $a_i = \frac{|B(i)|}{|B|}$, which is the stationary distribution of the SALSA algorithm. A similar observation can be made for the hub weights.

If the underlying graph of the Base Set consists of more than one component, then the SALSA algorithm selects a starting point uniformly at random, and performs a random walk within the connected component that contains that node. Formally, let j be a component that contains node i, let N_j denote the number of nodes in the component, and B_j the set of (backward) links in component j. Then, the authority weight of node i is

$$a_i = \frac{N_j}{N} \frac{|B(i)|}{|B_j|}\ .$$

Led astray by the simplifying assumption of a single component, we considered a simplified version of the SALSA algorithm where the authority weight of a node is the ratio $|B(i)|/|B|$. This corresponds to the case that the starting point for the random walk is chosen with probability proportional to the "popularity" of the node, that is, the number of links that point to this node. We will refer to this variation of the SALSA algorithm as pSALSA (popularity SALSA)[1]. We will consider the original SALSA algorithm as defined in [9] in the full version of our paper.

An interesting generalization of the SALSA algorithm is considered by Rafiei and Mendelzon [11]. They propose an algorithm for computing reputations that is a hybrid of the SALSA algorithm, and the PageRank algorithm. At each step, with probability d, the Rafiei and Mendelzon algorithm jumps to a page of the collection chosen uniformly at

[1] We thank Ronny Lempel and Shlomo Moran for pointing out the difference between SALSA and pSALSA.

random, and with probability $1-d$ it performs a SALSA step.

2.4 The PHITS Algorithm

Cohn and Chang [4] propose a statistical hubs and authorities algorithm, which they call the PHITS Algorithm. They propose a probabilistic model in which a citation c of a document d is caused by a latent “factor” or “topic”, z. It is postulated that there are conditional distributions $P(c|z)$ of a citation c given a factor z, and also conditional distributions $P(z|d)$ of a factor z given a document d. In terms of these conditional distributions, they produce a *likelihood function*.

Cohn and Chang then propose using the EM Algorithm of Dempster et al. [5] to assign the unknown conditional probabilities so as to maximize this likelihood function L, and thus best “explain” the proposed data. Their algorithm requires specifying in advance the number of factors z to be considered. Furthermore, it is possible that their EM Algorithm could get “stuck” in a local maximum, without converging to the true global maximum.

3. RANDOM WALKS AND THE KLEINBERG ALGORITHM

The fact that the output of the first (half) step of the Kleinberg algorithm can be seen as the stationary distribution of a certain random walk on the underlying graph, poses the natural question of whether other intermediary results of Kleinberg's algorithm (and ultimately the output of the algorithm itself) can also be seen as the stationary distribution of a random walk. We show that this is indeed the case.

THEOREM 1. *There exist sequences $M_1^a, M_2^a, \ldots, M_n^a, \ldots$, and $M_1^h, M_2^h, \ldots, M_n^h, \ldots$ of Markov Chains, such that, for each $n \geq 1$, the stationary distribution of M_n^a is equal to the authority vector after the n^{th} iteration of Kleinberg's algorithm, and the stationary distribution of M_n^h is equal to the hub vector after the n^{th} iteration of Kleinberg's algorithm.*

PROOF. We first introduce the following notation. We say that we follow a B path if we follow a link backwards, and we say we follow an F path if we follow a link forward. We can combine these to obtain longer paths. For example a BF path is a path that first follows a link backwards, and then a link forward. Now, let $(BF)^n(i,j)$ denote the set of $(BF)^n$ paths that go from i to j, $(BF)^n(i)$ the set of $(BF)^n$ paths that leave node i, and $(BF)^n$ the set of all possible $(BF)^n$ paths. We can define similar sets for the $(FB)^n$ paths.

By definition of the $(A^TA)^n$, and $(AA^T)^n$ matrices, we have that $|(BF)^n(i,j)| = (A^TA)^n(i,j)$, and $|(FB)^n(i,j)| = (AA^T)^n(i,j)$. Also, $|(BF)^n(i)| = \sum_j (A^TA)^n(i,j)$, and $|(FB)^n(i)| = \sum_j (AA^T)^n(i,j)$. After the n^{th} operation of the Kleinberg algorithm the authority vector $\boldsymbol{a}$, and hub vector $\boldsymbol{h}$ are the unit vectors in the direction of $(A^TA)^n\boldsymbol{u}$ and $(AA^T)^n\boldsymbol{u}$, respectively. (This actually assumes that in order to compute the authority weights we switch the order of the operations $\mathcal{I}$ and $\mathcal{O}$, but asymptotically this does not make any difference). If we take the unit vectors under the L_1 norm, then we have

$$a_i = \frac{|(BF)^n(i)|}{|(BF)^n|}, \qquad \text{and} \qquad h_i = \frac{|(FB)^n(i)|}{|(FB)^n|}. \qquad (1)$$

Now, we define the undirected weighted graph $G_{(BF)^n}$ as follows. The vertex set of the graph is the set of nodes in the base set. We place an edge between two nodes i and j if there is a $(BF)^n$ path between these nodes. The weight of the edge is $|(BF)^n(i,j)|$, the number of $(BF)^n$ paths between i and j. We perform a random walk on graph $G_{(BF)^n}$. When at node i, we move to node j with probability proportional to the number of paths between i and j. The corresponding Markov Chain $M_{(BF)^n}$ has transition probabilities

$$P_a(i,j) = \frac{|(BF)^n(i,j)|}{|(BF)^n(i)|}.$$

From a standard theorem on random walks on weighted graphs (see, e.g., p. 132 of [10] for the corresponding result on unweighted graphs), the stationary distribution of $M_{(BF)^n}$ is the same as the vector $\boldsymbol{a}$ in equation (1). Similarly, we can define the graph $G_{(FB)^n}$, and the corresponding Markov Chain $M_{(FB)^n}$, for the hubs case. Setting M_n^a to $M_{(BF)^n}$, and M_n^h to $M_{(FB)^n}$ concludes the proof. □

4. SOME MODIFICATIONS TO THE KLEINBERG AND SALSA ALGORITHMS

While Kleinberg's algorithm has some very desirable properties, it also has its limitations. One potential problem is the possibility of severe “topic drift”. Roughly, Kleinberg's algorithm converges to the most “tightly-knit” community within the Base Set. It is possible that this tightly-knit community will have little or nothing to do with the proposed query topic.

A striking example of this phenomenon is provided by Cohn and Chang ([4], p. 6). They use Kleinberg's Algorithm with the search term “jaguar” (an example query suggested by Kleinberg [8]), and converge to a collection of sites about the city of Cincinnati! They determine that the cause of this is a large number of on-line newspaper articles in the Cincinnati Enquirer which discuss the Jacksonville Jaguars football team, and all link to the same standard Cincinnati Enquirer service pages. Interestingly, in a preliminary experiment with the query term “abortion” (another example query suggested by Kleinberg [8]), we also found the Kleinberg Algorithm converging to a collection of web pages about the city of Cincinnati!

Now, in both these cases, we believe it is possible to eliminate such errant behavior through more careful selection of the Base Set, and more careful elimination of intra-domain hypertext links. Nevertheless, we do feel that these examples point to a certain “instability” of Kleinberg's Algorithm.

4.1 The Hub-Averaging-Kleinberg Algorithm

We propose here a small modification of Kleinberg's algorithm to help remedy the above-mentioned instability. For motivation, consider the following. Suppose there are $M+1$ authority pages, and $M+1$ hub pages, with M large. The first M hubs link only to the first authority, while the final hub links to all $M+1$ authorities. In such a set-up, we would expect the first authority to be considered much more authoritative than all the others, and Kleinberg's algorithm does indeed do this. On the other hand, it seems that the final hub should be *worse* than the others, since in addition to linking to a good authority (the first authority), it also links to many bad authorities. However, according to Kleinberg's algorithm, it is the *best* hub, because linking to more things can only improve your hub rating.

Inspired by such considerations, we propose an algorithm which is a "hybrid" of the Kleinberg and SALSA algorithms. Namely, it does the authority rating updates $\mathcal{I}$ just like Kleinberg (i.e., giving each authority a rating equal to the sum of the hub ratings of all the pages that link to it), but does the hub rating updates $\mathcal{O}$ by instead giving each hub a rating equal to the *average* of the authority ratings of all the pages that it links to. With this modified "Hub-Averaging" algorithm, a hub is better if it links to *only* good authorities, rather than linking to both good *and* bad authorities.

4.2 The Threshold-Kleinberg Algorithms

We propose two different "threshold" modifications to Kleinberg's Algorithm. The first modification, Hub-Threshold, is applied to the in-step $\mathcal{I}$. When computing the authority weight of the i^{th} page, this algorithm does not take into account all hubs that point to page i. It only counts those hubs whose hub weight is at least the average hub weight over all the hubs that point to page i, computed using the current hub weights for the nodes. This corresponds to saying that a site should not be considered a good authority simply because a lot of very poor hubs point to it.

The second modification, Authority-Threshold, is applied to the out-step $\mathcal{O}$. When computing the hub weight of the i^{th} page, this algorithm does not take into account all authorities pointed to by page i. It only counts those authorities which are among the top K authorities, judging by current authority values. The value of K is passed as a parameter to the algorithm. This corresponds to saying that a site should not be considered a good hub simply because it points to a number of "acceptable" authorities; rather, to be considered a good hub the site must point to some of the *best* authorities. This is inspired partially by the fact that, in most web searches, a user only visits the top few authorities.

We also consider a Full-Threshold algorithm, which makes *both* the Hub-Threshold and Authority-Threshold modifications to Kleinberg's Algorithm.

4.3 The Breadth-First-Search Algorithm: A Normalized n-step Variant

When the pSALSA algorithm computes the authority weight of a page, it takes into account only the popularity of this page within its immediate neighborhood, disregarding the rest of the graph. On the other hand, the Kleinberg algorithm considers the whole graph, taking into account more the structure of the graph around the node, than the popularity of that node in the graph. Specifically, after n steps, the authority weight of the i^{th} authority is $|(BF)^n(i)|/|(BF)^n|$, where $|(BF)^n(i)|$ is the number of $(BF)^n$ paths that leave node i. Another way to think of this is that the contribution of a node $j \neq i$ to the weight of i is equal to the number of $(BF)^n$ paths that go from i to j. Therefore, if a small bipartite component intercepts the path between node j and i, the contribution of node j will increase exponentially fast. This may not always be desirable, especially if the bipartite component is not representative of the query.

We propose the Breadth-First-Search (BFS) algorithm, as a generalization of the pSALSA algorithm, and a restriction of the Kleinberg algorithm. The BFS algorithm extends the idea of popularity that appears in pSALSA from a one link neighborhood to an n-link neighborhood. However, instead of considering the number of $(BF)^n$ *paths* that leave i, it considers the number of $(BF)^n$ *neighbors* of node i. We let $(BF)^n(i)$ denote the set of nodes that can be reached from i by following a $(BF)^n$ path. The contribution of node j to the weight of node i depends on the distance of the node j from i. We adopt an exponentially decreasing weighting scheme. Therefore, the weight of node i is determined as follows:

$$a_i = 2^{n-1}|B(i)|+2^{n-2}|BF(i)|+2^{n-3}|BFB(i)|+\ldots+|(BF)^n(i)|.$$

The algorithm starts from node i, and visits its neighbors in BFS order. At each iteration it takes a Backward or a Forward step (depending on whether it is an odd, or an even iteration), and it includes the *new* nodes it encounters. The weight factors are updated accordingly. Note that each node is considered only once, when it is first encountered by the algorithm.

5. A BAYESIAN ALGORITHM

A different type of algorithm is given by a fully Bayesian statistical approach to authorities and hubs. Suppose there are M hubs and N authorities (which could be the same set). We suppose that each hub i has an (unknown) real parameter e_i, corresponding to its "general tendency to have hypertext links", and also an (unknown) non-negative parameter h_i, corresponding to its "tendency to have *intelligent* hypertext links to *authoritative* sites". We further suppose that each authority j has an (unknown) non-negative parameter a_j, corresponding to its level of authority.

Our statistical model is as follows. The *a priori* probability of a link from hub i to authority j is given by

$$\mathbf{P}(i \to j) \;=\; \frac{\exp(a_j h_i + e_i)}{1+\exp(a_j h_i + e_i)}\,, \tag{2}$$

with the probability of no link from i to j given by

$$\mathbf{P}(i \not\to j) \;=\; \frac{1}{1+\exp(a_j h_i + e_i)}\,. \tag{3}$$

This reflects the idea that a link is more likely if e_i is large (in which case hub i has large tendency to link to *any* site), or if *both* h_i and a_j are large (in which case i is an intelligent hub, and j is a high-quality authority).

To complete the specification of the statistical model from a Bayesian point of view (see, e.g., Bernardo and Smith [2]), we must assign *prior* distributions to the $2M+N$ unknown parameters e_i, h_i, and a_j. (These priors should be general and uninformative, and should *not* depend on the observed data. For large graphs, the choice of priors should have only a small impact on the results.) To do this, we let $\mu = -5.0$ and $\sigma = 0.1$ be fixed parameters, and let each e_i have prior distribution $N(\mu, \sigma^2)$, a normal distribution with mean μ and variance σ^2. We further let each h_i and a_j have prior distribution Exp(1) (since they have to be non-negative), meaning that for $x \geq 0$, $\mathbf{P}(h_i \geq x) = \mathbf{P}(a_j \geq x) = \exp(-x)$.

The (standard) Bayesian inference method then proceeds from this fully-specified statistical model, by *conditioning* on the observed data, which in this case is the matrix A of actual observed hypertext links in the Base Set. Specifically, when we condition on the data A we obtain a *posterior density* $\pi : \mathbf{R}^{2M+N} \to [0,\infty)$ for the parameters $(e_1,\ldots,e_M,h_1,\ldots,h_M,a_1,\ldots,a_N)$. This density is defined so that

$$\mathbf{P}\Big((e_1,\ldots,e_M,h_1,\ldots,h_M,a_1,\ldots,a_N)\in S \;\Big|\; \{A_{ij}\}\Big)$$
$$= \int_S \pi(e_1,\ldots,e_M,h_1,\ldots,h_M,a_1,\ldots,a_N)$$
$$de_1\ldots de_M dh_1\ldots dh_M da_1\ldots da_N \qquad (4)$$

for any (measurable) subset $S \subseteq \mathbf{R}^{2M+N}$, and also

$$\mathbf{E}\Big(g(e_1,\ldots,e_M,h_1,\ldots,h_M,a_1,\ldots,a_N) \;\Big|\; \{A_{ij}\}\Big)$$
$$= \int_{\mathbf{R}^{2M+N}} g(e_1,\ldots,e_M,h_1,\ldots,h_M,a_1,\ldots,a_N)$$
$$\pi(e_1,\ldots,e_M,h_1,\ldots,h_M,a_1,\ldots,a_N)$$
$$de_1\ldots de_M dh_1\ldots dh_M da_1\ldots da_N$$

for any (measurable) function $g : \mathbf{R}^{2M+N} \to \mathbf{R}$. An easy computation gives the following.

Lemma 1. *For our model, the posterior density is given, up to a multiplicative constant, by*

$$\pi(e_1,\ldots,e_M,h_1,\ldots,h_M,a_1,\ldots,a_N)$$
$$\propto \prod_{i=0}^{M-1}\exp(-h_i)\exp[-(e_i-\mu)^2/(2\sigma^2)] \times \prod_{j=0}^{N-1}\exp(-a_j)$$
$$\times \prod_{(i,j):A_{ij}=1}\exp(a_jh_i+e_i) \Big/ \prod_{all\ i,j}(1+\exp(a_jh_i+e_i))\,.$$

Our Bayesian algorithm then reports the conditional means of the $2M+N$ parameters, according to the posterior density π. That is, it reports final values $\widehat{a}_j$, $\widehat{h}_i$, and $\widehat{e}_i$, where, for example

$$\widehat{a}_j = \int_{\mathbf{R}^{2M+N}} a_j\pi(e_1,\ldots,e_M,h_1,\ldots,h_M,a_1,\ldots,a_N)$$
$$de_1\ldots de_M dh_1\ldots dh_M da_1\ldots da_N\,.$$

To actually compute these conditional means is non-trivial. To accomplish this, we used a *Metropolis Algorithm.* (The Metropolis algorithm is an example of a *Markov chain Monte Carlo Algorithm*; for background see, e.g., Smith and Roberts [13]; Tierney [14]; Gilks et al. [6]; Roberts and Rosenthal [12]).

There is, of course, some arbitrariness in the specification of this Bayesian algorithm, e.g., in the form of the prior distributions and in the precise formula for the probability of a link from i to j. However, the model appears to work well in practice, as our experiments show. We note that it is possible that the priors for a new search query could instead depend on the performance of hub i on different *previous* searches, though we do not pursue that here.

This Bayesian algorithm is similar in spirit to the PHITS algorithm of Cohn and Chang [4] described earlier, in that both use statistical modeling, and both use an iterative algorithm to converge to an answer. However, the algorithms differ substantially in their details. Firstly, they use substantially different statistical models. Secondly, the PHITS algorithm uses a non-Bayesian (i.e. "classical" or "frequentist") statistical framework, as opposed to the Bayesian framework adopted here.

5.1 A Simplified Bayesian Algorithm

It is possible to simplify the above Bayesian model, by replacing equation (2) with $\mathbf{P}(i\to j) = (a_jh_i)/(1+a_jh_i)$ and correspondingly replace equation (3) with $\mathbf{P}(i \not\to j) = 1/(1+a_jh_i)$. This eliminates the parameters e_i entirely, so that we no longer need the prior values μ and σ.

This leads to a slightly modified posterior density $\pi(\cdot)$, now given by $\pi : \mathbf{R}^{M+N} \to \mathbf{R}^{\geq 0}$ where

$$\pi(h_1,\ldots,h_M,a_1,\ldots,a_N)$$
$$\propto \textstyle\prod_{i=0}^{M-1}\exp(-h_i) \times \prod_{j=0}^{N-1}\exp(-a_j) \times \prod_{(i,j):A_{ij}=1} a_jh_i$$
$$\Big/ \textstyle\prod_{all\ i,j}(1+a_jh_i)\,.$$

This Simplified Bayesian algorithm was designed to be to similar to the original Bayesian algorithm. Surprisingly, we will see that experimentally it often performs very similarly to the pSALSA algorithm.

6. EXPERIMENTAL RESULTS

We have implemented the algorithms presented here on various queries. Because of space limitations we only report here (see Appendix A) a representative subset of results; all of our results (including the queries "death penalty", "computational complexity" and "gun control" which are not reported here) can be obtained at `http://www.cs.toronto.edu/~tsap/experiments`. The reader may find it easier to follow the discussion in the next section by accessing the full set of results. For the generation of the Base Set of pages, we follow the specifications of Kleinberg [8] described earlier. For each of the queries, we begin by generating a Root Set that consists of the first 200 pages returned by AltaVista on the same query. The Root Set is then expanded to the Base Set by including nodes that point to, or are pointed to, by the pages in the Root Set. In order to keep the size of the Base Set manageable, for every page in the Root Set, we only include the first 50 pages returned from AltaVista that point to this page. We then construct the graph induced by nodes in the Base Set, by discovering all links among the pages in the Base Set, eliminating those that are between pages of the same domain[2].

For each query, we tested nine different algorithms on the same Base Set. We present the top ten authority sites returned by each of the algorithms. For evaluation purposes, we also include a list of the URL and title (possibly abbreviated) of each site which appears in the top five of one or more of the algorithms. For each page we also note the popularity of the page (denoted *pop* in the tables), that is, the number of different algorithms that rank it in the top ten sites. The pages that seem (to us) to be generally unrelated with the topic in hand appear bold-faced. We also present an "intersection table" which provides, for each pair of algorithms, the number of sites which were in the top ten according to *both* algorithms (maximum 10, minimum 0).

In the tables, SBayesian denotes the Simplified Bayesian algorithm, HubAvg denotes the Hub-Averaging Kleinberg algorithm, AThresh denotes the Authority-Threshold algorithm, HThresh denotes the Hub-Threshold algorithm, and FThresh denotes the Full-Threshold algorithm. For the Authority-Threshold and Full-Threshold algorithms, we (arbitrarily) set the threshold $K = 10$.

[2] If one modifies the way the Base Set or the graph is constructed, the results of the algorithms can vary dramatically. In the above-mentioned site we report the output of the algorithms for the same query, over different graphs.

6.1 Discussion of Experimental Results

We observe from the experiments that different algorithms emerge as the "best" for different queries, while there are queries for which no algorithm seems to perform well. One prominent such case is the query on "net censorship" (also on "computational complexity") where only a few of the top ten pages returned by any of the algorithms can possibly be considered as authoritative on the subject. One possible explanation is that in these cases the topic is not well represented on the web, or there is no strong interconnected community. This reinforces a common belief that any commercial search engine cannot rely solely on link information, but rather must also examine the text content of sites to prevent such difficulties as "topic drift". On the other hand, in cases such as "death penalty" (not shown here), all algorithms converge to almost the same top ten pages, which are both relevant and authoritative. In these cases the community is well represented, and strongly interconnected.

The experiments also indicate the difference between the behavior of the Kleinberg algorithm and pSALSA, first observed for the SALSA algorithm in the original paper of Lempel and Moran [9]. Specifically, when computing the top authorities, the Kleinberg algorithm tends to concentrate on a "tightly knit community" of nodes (the TKC effect), while pSALSA, like SALSA, tends to mix the authorities of different communities in the top authorities. The TKC effect becomes clear in the "genetic" query, where the Kleinberg algorithm only reports pages on biology in the top ten while pSALSA mixes these pages with pages on genetic algorithms. It also becomes poignantly clear in the "movies" query (and also in the "gun control" and the "abortion" query), where the top ten pages reported by the Kleinberg algorithm are dominated by an irrelevant cluster of nodes from the `about.com` community. A more elaborate algorithm for detecting intra-domain links could help alleviate this problem. However, these examples seem indicative of the topic drift potential of the principal eigenvector in the Kleinberg algorithm.

On the other hand, the limitations of the pSALSA algorithm become obvious in the "computational geometry" query, where three out of the top ten pages belong to the unrelated `w3.com` community. They appear in the top positions because they are pointed to by a large collection of pages by ACM, which point to nothing else. A similar phenomenon explains the appearance of the "Yahoo!" page in the "genetic" query. We thus see that the simple heuristic of counting the in-degree as the authority weight is also imperfect.

We identify two types of characteristic behavior: the Kleinberg behavior, and the pSALSA behavior. The former ranks the authorities based on the structure of the entire graph, and tends to favor the authorities of tightly knit communities. The latter ranks the authorities based on their popularity in their immediate neighborhood, and favors various authorities from different communities. To see how the rest of the algorithms fit within these two types of behaviors, we compare the behavior of algorithms on a pairwise basis, using the number of intersections in their respective top ten authorities as an indication of agreement.

The first striking observation is that the Simplified Bayesian algorithm is almost identical to the pSALSA algorithm. The pSALSA algorithm and the Simplified Bayesian have at least 80% overlap on all queries. One possible explanation for this is that both algorithms place great importance on the in-degree of a node when determining the authority weight of a node. For the pSALSA algorithm we know that it is "local" in nature, that is, the authority weight assigned to a node depends only on the links that point to this node, and not on the structure of the whole graph. The Simplified Bayesian seems to possess a similar, yet weaker property; we explore the locality issue further in the next section. On the other hand, the Bayesian algorithm appears to resemble both the Kleinberg and the pSALSA behavior, leaning more towards the first. Indeed, although the Bayesian algorithm avoids the severe topic drift in the "movies" and the "gun control" queries (but not in the "abortion" case), it usually has higher intersection numbers with Kleinberg than with pSALSA. One possible explanation for this observation is the presence of the e_i parameters in the Bayesian algorithm (but not the Simplified Bayesian algorithm), which "absorb" some of the effect of many links pointing to a node, thus causing the authority weight of a node to be less dependent on its in-degree.

Another algorithm that seems to combine characteristics of both the pSALSA and the Kleinberg behavior is the Hub-Averaging algorithm. The Hub-Averaging algorithm is by construction a hybrid of the two since it alternates between one step of each algorithm. It shares certain behavior characteristics with the Kleinberg algorithm: if we consider a full bipartite graph, then the weights of the authorities increase exponentially fast for Hub-Averaging (the rate of increase, however, is the square root of that of the Kleinberg algorithm). However, if the component becomes infiltrated, by making one of the hubs point to a node outside the component, then the weights of the authorities in the component *drop*. This prevents the Hub-Averaging algorithm from completely following the drifting behavior of the Kleinberg algorithm in the "movies" query. Nevertheless, in the "genetic" query, Hub-Averaging agrees strongly with Kleinberg, focusing on sites of a single community, instead of mixing communities as does pSALSA[3]. On the other hand, Hub-Averaging and pSALSA share a common characteristic, since the Hub-Averaging algorithm tends to favor nodes with high in-degree: if we consider an isolated component of one authority with high in-degree, the authority weight of this node will increase exponentially fast. This explains the fact that the top three authorities for "computational geometry" are the `w3.com` pages that are also ranked highly by pSALSA (with Hub-Averaging giving a very high weight to all three authorities).

For the threshold algorithms, since they are modifications of the Kleinberg Algorithm, they are usually closer to the Kleinberg behavior. This is especially true for the Hub-Threshold algorithm. However, the benefit of eliminating unimportant hubs when computing authorities becomes obvious in the "abortion" query. If one looks further than the first ten pages returned by Kleinberg, one observes that after the first nine pages, which belong to the `amazon.com` community, the rest of the pages are on topic. The Hub-Threshold algorithm escapes this cluster, and moves directly to the relevant pages. The intersection between the top ten pages of Hub-Threshold, and the set of pages in the positions 10 to 20 in the Kleinberg algorithm is 80%.

[3] In a version of the "abortion" query (denoted "refined" in the site), the Hub-Averaging algorithm did some mixing of communities, but to a smaller degree than pSALSA.

The Authority-Threshold often appears to be most similar with the Hub-Averaging algorithm. This makes sense since these two algorithms have a similar underlying motivation. The best moment for Authority-Threshold is the "movies" query, where it reports the most relevant top ten pages among all algorithms. The Full-Threshold algorithm combines elements of both the Threshold algorithms; however, usually it reports in the top ten a mixture of the results of the two algorithms, rather than the best of the two.

Finally, the BFS algorithm is designed to be a generalization of the pSALSA algorithm, that combines some elements of the Kleinberg algorithm. Its behavior resembles both pSALSA and Kleinberg, with a tendency to favor pSALSA. In the "genetic" and "abortion" queries it demonstrates some mixing, but to a lesser extent than that of pSALSA. The most successful moments for BFS are the "abortion" and the "gun control" queries where it reports a set of top ten pages that are all on topic. An interesting question to investigate is how the behavior of the BFS algorithm is altered if we change the weighting scheme of the neighbors.

7. THEORETICAL ANALYSIS

The experimental results of the previous section suggest that certain algorithms seem to share similar properties and ranking behavior. In this section, we initiate a (preliminary) formal study of fundamental properties and comparisons between ranking algorithms. For the purpose of following analysis we need some basic definitions and notation. Let $\mathcal{G}_N$ be a collection of graphs of size N. One special case is to let $\mathcal{G}_N$ be the set of *all* graphs of size N, hereafter denoted $\overline{\mathcal{G}}_N$. We define a link analysis algorithm A as a function that maps a graph $G \in \mathcal{G}_N$ to an N-dimensional vector. We call the vector $A(G)$ the *weight* vector of algorithm A on graph G. The value of the entry $A(G)(i)$ of vector $A(G)$ denotes the authority weight assigned by the algorithm A to the page i.

We can normalize the weight vector $A(G)$ under some chosen norm. The choice of normalization affects the definition of some of the properties of the algorithms, so we discriminate between algorithms that use different norms. For any norm L, we define the L-algorithm A to be the algorithm A, where the weight vector of A is normalized under L. We also examine *unnormalized* algorithms, where no normalization is performed at any stage of the algorithm. For example, the unnormalized pSALSA algorithm assigns a weight to page i equal to the in-degree of page i. For the following discussion, when not stated explicitly, we will assume that the weight vectors of the algorithms are normalized under the L_∞ norm (i.e. each weight is divided by the maximum weight); this gives weight 1 to the top authority, with other weights given as a fraction of the top weight. Due to space constraints, many proofs have been omitted in the following sections.

7.1 Monotonicity

Definition 1. An algorithm A is *monotone* if it has the following property: If j and k are two different nodes in a graph G, such that every hub which links to j also links to k, then $A(G)(k) \geq A(G)(j)$.

Monotonicity appears to be a "reasonable" property that any sensible link-analysis algorithm should satisfy.

THEOREM 2. *The algorithms we consider in this paper (including the SALSA algorithm) are all monotone.*

7.2 Similarity

Let A_1 and A_2 be two algorithms on $\mathcal{G}_N$. We shall consider the *distance* $d(A_1(G), A_2(G))$ between the weight vectors of $A_1(G)$ and $A_2(G)$, for $G \in \mathcal{G}_N$, where $d : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}$ is some function that maps the weight vectors $A_1(G)$ and $A_2(G)$ to a real number $d(A_1(G), A_2(G))$. We shall consider the Manhattan distance d_1, that is, the L_1 norm of the difference of the weight vectors, given by $d_1(w_1, w_2) = \sum_{i=1}^{N} |w_1(i) - w_2(i)|$.

For this distance function, we now define the similarity between two L_∞-algorithms as follows[4].

Definition 2. Two L_∞-algorithms A_1 and A_2 are similar on $\{\mathcal{G}_N\}$, if (as $N \to \infty$)

$$\max_{G \in \mathcal{G}_N} d_1(A_1(G), A_2(G)) = o(N) .$$

We also consider another distance function that attempts to capture the similarity between the *ordinal* rankings of two algorithms. The motivation behind this definition is that the ordinal ranking is the usual end-product seen by the user. Let $w_1 = A_1(G)$ and $w_2 = A_2(G)$ be the weight vectors of two algorithms A_1 and A_2. We define the indicator function $\mathbf{I}_{w_1 w_2}(i,j)$ as follows

$$\mathbf{I}_{w_1 w_2}(i,j) = \begin{cases} 1 & \text{if } w_1(i) < w_1(j) \text{ AND } w_2(i) > w_2(j) \\ 0 & \text{otherwise} \end{cases}$$

We note that $\mathbf{I}_{w_1 w_2}(i,j) = 0$ if and only if $w_1(i) < w_2(j) \Rightarrow w_2(i) \leq w_2(j)$. $\mathbf{I}_{w_1 w_2}(i,j)$ becomes one for each pair of nodes that are ranked differently. We define the "ranking distance" function d_r as follows.

$$d_r(w_1, w_2) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{I}_{w_1 w_2}(i,j) .$$

Note that, unlike d_1, the distance between two weight vectors under d_r does not depend upon the choice of normalization.

Definition 3. Two algorithms, A_1 and A_2, are rank matching on $\{\mathcal{G}_N\}$, if for every N, and every graph $G \in \mathcal{G}_N$,

$$d_r(A_1(G), A_2(G)) = 0 .$$

Remark: We note that by the above definition, every algorithm is rank matching with the trivial algorithm that gives the same weight to all authorities. Although this may seem somewhat bizarre, it does have an intuitive justification. For an algorithm whose goal is to produce an *ordinal* ranking, the weight vector with all weights equal conveys no information; therefore, it lends itself to all possible ordinal rankings. We also note that the d_r distance does not satisfy the triangle inequality, since, e.g., all algorithms have d_r-distance 0 to the trivial algorithm. Of course, it is straightforward to modify the definition of d_r to avoid this; however, we find the definition used here to be most natural.

[4] The definition of similarity can be generalized to any distance function d, and any normalization norm $||\cdot||$, by requiring instead that $\max_{G \in \mathcal{G}_N} d(A_1(G), A_2(G)) = o(M_N)$ as $N \to \infty$, where $M_N = \sup_{\|w_1\|=\|w_2\|=1} d(w_1, w_2)$ is the maximum distance between any two N-vectors with unit norm. Most of our results can be generalized to any L_p distance function and any L_q norm.

Proposition 1. *The L_∞-Hub-Averaging algorithm, and the L_∞-Kleinberg algorithm are neither similar, nor rank matching on $\overline{\mathcal{G}}_N$.*

Proof. Consider a graph G on $N = 3r$ nodes that consists of two disconnected components. The first component C_1 consists of a complete graph on r nodes. The second component C_2 consists of a complete graph C on r nodes, and a set of r "external" nodes E, such that each node in C points to a node in E, and no two nodes in C point to the same "external" node.

Let w_K and w_H denote the weight vectors of the Kleinberg, and the Hub-Averaging algorithm, respectively, on graph G. It is not hard to see that the Kleinberg algorithm allocates all the weight to the nodes in C_2. After normalization, for all $i \in C$, $w_K(i) = 1$, for all $j \in E$, $w_K(j) = \frac{1}{m}$, and for all $k \in C_1$, $w_K(k) = 0$. On the other hand, the Hub-Averaging algorithm allocates all the weight to the nodes in C_1. After normalization, for all $k \in C_1$, $w_H(k) = 1$, and for all $j \in C_2$, $w_H(j) = 0$.

Therefore, it is easy to see that $d_1(w_K, w_H) = 2r = \frac{2N}{3}$ which proves that the algorithms are not similar.

The proof for rank dissimilarity follows immediately from the above. For every pair of nodes (i, j) such that $i \in C_1$ and $j \in C_2$, $w_K(i) > w_K(j)$, and $w_S(i) > w_S(j)$. There are $\Theta(N^2)$ such pairs, therefore, $d_r(w_K, w_H) = \Theta(N)$. Thus, the two algorithms are not rank matching. □

Proposition 2. *The L_∞-pSALSA algorithm and the L_∞-Kleinberg algorithm are neither similar, nor rank matching on $\overline{\mathcal{G}}_N$.*

Proof. Consider a graph G on $N = 4r$ nodes that consists of two disconnected components. The first component C_1 consists of a complete graph on r nodes. Thus, each node points to, and is pointed to by, $r - 1$ nodes. The second component C_2 consists of a bipartite graph with $2r$ hubs, and r authorities. Without loss of generality assume that r is even, and enumerate all hubs and authorities. Make all "odd" hubs point to all "odd" authorities, and all "even" hubs point to all "even" authorities. Thus, each hub points to $\frac{r}{2}$ authorities, and each authority is pointed to by r authorities.

Let w_K and w_S denote the weight vectors of the Kleinberg, and the pSALSA algorithm, respectively, on graph G. It is not hard to see that the Kleinberg algorithm allocates all the weight to the nodes in C_1. After normalization, for all $i \in C_1$, $w_K(i) = 1$, while for all $j \in C_2$, $w_S(j) = 0$. On the other hand, the pSALSA algorithm distributes the weight to both components, allocating more weight to the nodes in C_2. After the normalization step, for all $j \in C_2$, $w(j) = 1$, while for all $i \in C_1$, $w_S(i) = \frac{r-1}{r}$.

Therefore, it is easy to see that

$$d_1(w_K, w_S) = r + r \cdot (1 - \frac{r-1}{r}) = \Theta(N)$$

which proves that the algorithms are not similar.

The proof for rank dissimilarity follows immediately from the above. For every pair of nodes (i, j) such that $i \in C_1$ and $j \in C_2$, $w_K(i) > w_K(j)$, and $w_S(i) > w_S(j)$. There are $\Theta(N^2)$ such pairs, therefore, $d_r(w_K, w_H) = \Theta(N)$. Thus, the two algorithms are not rank matching.

We note that a modification of this example can be used to prove the same result for the SALSA and the Kleinberg algorithms. □

Proposition 3. *The L_∞-pSALSA algorithm and the L_∞-Hub-Averaging algorithm are neither similar, nor rank matching on $\overline{\mathcal{G}}_N$.*

Proof. Consider a graph G on $N = 3r + 3$ nodes which are connected as follows. The graph consists of two sets of hubs X and Y of size r and 2, respectively, and two sets of authorities A and B, each of size r, and a single "central" authority c. Each hub in set X points to exactly one distinct authority in A, and both hubs in Y point to all authorities in B. Furthermore, all hubs in X and Y point to c.

Let w_S and w_H be the weight vectors of pSALSA and Hub-Averaging, respectively. The pSALSA algorithm allocates the most weight to the central authority, then to the authorities in B, and then to the authorities in A. After normalization, $w_S(c) = 1$, for all $i \in A$, $w_S(i) = \frac{1}{r+2}$, and for all $j \in B$, $w_S(j) = \frac{2}{r+2}$.

On the other hand, the Hub-Averaging algorithm considers each hub in X to be much better than each hub in Y. Hence, it will allocate highest weight to the authority c, nearly as high weight to the authorities in A, and much lower weight to the authorities in B. This shows that the two algorithms are neither similar nor rank matching.

We note that the same example can be used to prove the dissimilarity between SALSA and the Hub-Averaging algorithm. □

On the other hand, we have the following.

Definition 4. A link graph is "nested" if for every pair of nodes j and k, the set of in-links to j is either a subset or a superset of the set of in-links to k.

Let $\mathcal{G}_N^{nest}$ be the set of all size-N nested graphs. (Of course, $\mathcal{G}_N^{nest}$ is a rather restricted set of size-N graphs.)

Theorem 3. *If two algorithms are both monotone, then they are rank matching on $\mathcal{G}_N^{nest}$.*

Corollary 1. *The algorithms we consider in this paper (including the SALSA algorithm) are all rank matching on $\mathcal{G}_N^{nest}$.*

7.3 Stability and Locality

In the previous section we examined the similarity of two different algorithms on the same graph G. In this section we are interested on how the output of a *fixed* algorithm changes, as we alter the graph. We would like small changes in the graph to have a small effect on the weight vector of the algorithm. We capture this requirement by the definition of stability. For the following, let $E(G)$ denote the set of all edges (i.e. links) in the graph G. We assume that $E(G) = \omega(1)$, otherwise all properties that we discuss below are trivial. The following definition applies to unnormalized, and L_∞-algorithms[5].

[5] As in the case of similarity, the notion of stability can be defined for any distance function, and for any normalization norm.

Definition 5. An algorithm A is stable on $\{\mathcal{G}_N\}$ if for every fixed positive integer k, we have (as $N \to \infty$)

$$\max_{G\in\mathcal{G}_N,\ell_i\in E(G),1\le i\le k}\ \min_{\gamma>0} d_1(A(G),\gamma\cdot A(G\backslash\{\ell_1,\ldots,\ell_k\})) = o(N)\,,$$

where $G\backslash\{\ell_1,\ldots\ell_k\}$ is the graph G with the edges $\ell_1,\ldots,\ell_k$ removed.

Definition 6. An algorithm A is rank stable on $\{\mathcal{G}_N\}$ if for every k, we have (as $N \to \infty$)

$$\max_{G\in\mathcal{G}_N,\ell_i\in E(G),1\le i\le k} d_r(A(G),A(G\setminus\{\ell_1,\ldots,\ell_k\})) \;=\; o(N).$$

Stability may be a desirable property. Indeed, the algorithms all act on a base set which is generated using some other search engine (e.g. AltaVista [1]) and the associated hypertext links. Presumably with a "very good" base set, all the algorithms would perform well. However, if an algorithm is not stable, then slight changes in the base set (or its link structure) may lead to large changes in the rankings given by the algorithm. Thus, stability may provide "protection" from poor base sets. We note that the parameter γ used in the definition of stability allows for an arbitrary scaling of the second weight vector, thus eliminating instability which is caused solely by different normalization factors.

Proposition 4. *The L_∞-Kleinberg and L_∞-Hub-Averaging algorithms are neither stable, nor rank stable.*

We now introduce the idea of "locality". The idea behind locality is that a change in the in-links of a node should have only a small effect on the weights of the rest of the nodes.

Definition 7. An algorithm A is local if for graph G, and every link $\ell \in E(G)$, $|A(G)(i) - A(G\setminus\{\ell\})(i)| = 0$ for all $i \in G\setminus\{p^*\}$, where p^* is the page pointed to by the link ℓ.

Definition 8. An algorithm A is pairwise local if for every graph G, and every link $\ell \in E(G)$, $\frac{A(G)(i)}{A(G)(j)} = \frac{A(G\setminus\{\ell\})(i)}{A(G\setminus\{\ell\})(j)}$ for all $i,j \in G\setminus\{p^*\}$, where p^* is the document linked to by the link ℓ.

Definition 9. An algorithm A is rank local if for every graph G, and every link $\ell \in E(G)$, if $w = A(G)$ and $w' = A(G\setminus\{\ell\})$, then $\mathbf{I}_{ww'}(i,j) = 0$ for all $i,j \in G\setminus\{p^*\}$, where p^* is the document linked to by the link ℓ.

We note that locality depends on the normalization used, but pairwise locality and rank locality do not. The following lemmas are direct consequences of the definitions.

Lemma 2. *If an unnormalized algorithm is local, then the corresponding normalized algorithm is pairwise local (under any normalization).*

Lemma 3. *If an algorithm A is pairwise local, then it is rank local.*

We have the following.

Theorem 4. *If an algorithm is rank local, then it is rank stable (under any normalization).*

Proof. Let w be the weight vector of the algorithm on a graph G, and let w' be the weight vector of the algorithm on the modified graph $G\setminus\{\ell_1,\ell_2,...,\ell_k\}$. Let $P = \{p_1,p_2,...,p_m\}$, be the set of *distinct* pages pointed to by links $\ell_1,...,\ell_k$. Since the algorithm is rank local, $\mathbf{I}_{ww'}(i,j) = 0$ for all $i,j \notin P$. Therefore,

$$d_r(w,w') = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}\mathbf{I}_{ww'}(i,p_j)\ .$$

But $\mathbf{I}_{ww'}(i,p_j) \le 1$ for all i and p_j, so $d_r(w,w') = \frac{1}{N}\cdot N\cdot m = m$. Therefore, the algorithm is rank stable. Furthermore, rank locality is unaffected by normalization. □

Theorem 5. *If U is an unnormalized algorithm that is stable on $\{\mathcal{G}_N\}$, and A is the corresponding normalized algorithm under norm $\|\cdot\|$, and $\min_{G\in\mathcal{G}_N}\|U(G)\| = \Omega(1)$, then A is stable on $\{\mathcal{G}_N\}$.*

Proof. Let $G \in \mathcal{G}_n$ be a graph, and let $G' = G\backslash\{\ell_1,...,\ell_k\}$ be the modified graph. Let $u = U(G)$, and $u' = U(G')$, and let $w = A(G)$, and $w' = A(G')$. Since U is stable, $\sum_{i\in G}|u(i) - \gamma\, u'(i)| = o(N)$ for some $\gamma > 0$.

We have that $w(i) = u(i)/\|U(G)\|$, and $w'(i) = u'(i)/\|U(G')\|$. Set $\gamma' = \gamma\,\frac{\|U(G')\|}{\|U(G)\|}$. Then

$$\begin{aligned} d_1(w,\ \gamma'\,w') &= \sum_{i\in G}|w(i) - \gamma'\,w'(i)| \\ &= \frac{1}{\|U(G)\|}\cdot\sum_{i\in G}|u(i) - \gamma\,u'(i)| \\ &= o\left(\frac{N}{\|U(G)\|}\right)\ . \end{aligned}$$

Since $\|U(G)\| = \Omega(1)$, $d_1(w,\ \gamma'\,w') = o(N)$, therefore A is stable. □

Theorem 6. *The unnormalized pSALSA algorithm is local.*

Corollary 2. *The pSALSA algorithm (under any normalization) is both pairwise local and rank local.*

Corollary 3. *The pSALSA Algorithm (under any normalization) is rank stable.*

We originally thought that the Bayesian and Simplified Bayesian Algorithms were also local. However, it turns out that they are neither local nor pairwise local. Indeed, it is true that *conditional* on the values of h_i, e_i, and a_j, the conditional distribution of a_k for $k \ne j$ is unchanged upon removing a link from i to j. However, the *unconditional* marginal distribution of a_k, and hence also its posterior mean $\widehat{a}_k$ (or even ratios $\widehat{a}_k/\widehat{a}_q$ for $q \ne j$), may still be changed upon removing a link from i to j. (Indeed, we have computed experimentally that $\widehat{a}_3/\widehat{a}_4$ may change upon removing a link from 1 to 2, even for a simple example with just four nodes.) Hence, neither the Bayesian nor the Simplified Bayesian Algorithm is local or pairwise local.

Theorem 7. *The unnormalized pSALSA algorithm is stable.*

Corollary 4. *The pSALSA Algorithm (under any normalization) is stable.*

Finally, we use locality and "label-independence" to prove a uniqueness property of the pSALSA algorithm.

Definition 10. An algorithm is *label-independent* if permuting the labels of the graph nodes only causes the authority weights to be correspondingly permuted.

All of our algorithms are clearly label-independent, and we would expect this property of any reasonable algorithm. We have the following.

Theorem 8. *Suppose an algorithm A is local, monotone, and label-independent. Then A and pSALSA are rank matching on $\overline{\mathcal{G}}_N$.*

Proof. Let G be any graph, and let i be a node in G. Let $w_A = A(G)$. Since A is local, the value of $w_A(G)(i)$ is unchanged if we remove all links in G that do not point to i. Therefore, $w_A(i)$ depends solely on the set of in-links to i. Furthermore, since A is label-independent, it follows that $w_A(i)$ depends only on the *number* of in-links to i.

In particular, if two nodes i and j have the same number of in-links, then A assigns equal authority weight to the two nodes. Assume now that j has fewer in-links than i. Since A is local, we may modify the graph so that the nodes that point to j are a subset of those that point to i, without affecting $w_A(i)$ or $w_A(j)$. Since A is monotone, this implies that $w_A(j) \leq w_A(i)$.

Therefore, if w_S is the weight vector of the pSALSA algorithm on G, then $w_S(j) < w_S(i) \Rightarrow w_A(j) \leq w_A(i)$. Hence, $\mathbf{I}_{w_S w_A}(i,j) = 0$, for all i and j in G. It follows that $d_r(w_S, w_A) = 0$, as required.

We note that any normalized, or unnormalized variant of A is also rank matching with pSALSA. □

7.4 Symmetry

Definition 11. An algorithm A is "symmetric" if inverting all the links in a graph simply interchanges the hub and authority values produced by the algorithm.

We have by inspection:

Theorem 9. *The pSALSA (and SALSA) algorithm, the Kleinberg Algorithm, the Threshold-Kleinberg Algorithms, the BFS Algorithm, and the Simplified Bayesian Algorithm are all symmetric. However, the Hub-Averaging-Kleinberg Algorithm and the Bayesian Algorithm are NOT symmetric.*

8. SUMMARY

We have considered a number of known and some new algorithms which use the hypertext link structure of World Wide Web pages to extract information about the relative ranking of these pages. In particular, we have introduced two algorithms based on Bayesian statistical approach as well as a number of algorithms which are modifications of Kleinberg's seminal hubs and authority algorithm. Based on 8 different queries (5 presented here), we discuss some observed properties of each algorithm as well as relationships between the algorithms. We found (experimentally) that certain algorithms appear to be more "balanced", while others more "focused". The latter tend to be sensitive to the existence of tightly interconnected clusters, which may cause them to drift. The intersections between the lists of the top-ten results of the algorithms suggest that certain algorithms exhibit similar behavior and properties.

Motivated by the experimental observations, we introduced a theoretical framework for the study and comparison of link-analysis ranking algorithms. We formally defined (and gave some preliminary results for) the concepts of *monotonicity* and *locality*, as well as various concepts of *distance* and *similarity* between ranking algorithms.

Our work leaves a number of interesting open questions. The two Bayesian algorithms open the door to the use of other statistical and machine learning techniques for ranking of hyper-linked documents. Furthermore, the framework we defined suggests a number of interesting directions for the theoretical study of ranking algorithms, which we have just begun to explore in this work.

9. REFERENCES

[1] AltaVista Company. AltaVista search engine. http://www.altavista.com.

[2] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory.* John Wiley & Sons, Chichester, England, 1994.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *7th International World Wide Web Conference*, Brisbane, Australia, April 1998.

[4] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. Preprint, 2000.

[5] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[6] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in practice.* Chapman and Hall, London, 1996.

[7] Google. Google search engine. http://www.google.com.

[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM (JASM)*, 46, 1999.

[9] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In *9th International World Wide Web Conference*, Amsterdam, Netherlands, May 2000.

[10] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, Cambridge, England, June 1995.

[11] D. Rafiei and A. Mendelzon. What is this page known for? Computing web page reputations. In *9th International World Wide Web Conference*, Amsterdam, Netherlands, May 2000.

[12] G.O. Roberts and J.S. Rosenthal. Markov chain Monte Carlo: Some practical implications of theoretical results (with discussion). *Canadian Journal of Statistics*, 26:5–31, 1998.

[13] A.F.M. Smith and G.O. Roberts. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods (with discussion). *Journal of the Royal Statistical Society, Series B*, 55:3–24, 1993.

[14] L. Tierney. Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 22:1701–1762, 1994.

APPENDIX

A. EXPERIMENTS

A.1 Query: `abortion (Base Set size = 2293)`

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
1.	**P-1165**	P-717	**P-1165**	**P-1165**	P-717	**P-1165**	P-717	P-717	P-717
2.	**P-1184**	P-1461	**P-1184**	**P-1184**	P-1461	**P-1184**	P-719	P-1461	**P-1192**
3.	**P-1193**	P-719	**P-1193**	**P-1193**	P-1769	**P-1193**	P-1769	**P-1191**	**P-1165**
4.	**P-1187**	**P-1165**	**P-1187**	**P-1187**	P-719	P-1461	P-1461	P-719	**P-1191**
5.	**P-1188**	**P-1184**	**P-1188**	**P-1188**	P-1	P-719	P-962	**P-1184**	**P-1193**
6.	**P-1189**	**P-1193**	**P-1189**	**P-1189**	P-718	P-717	P-0	**P-1165**	**P-1184**
7.	**P-1190**	**P-1187**	**P-1190**	**P-1190**	P-0	P-1	P-2	**P-1192**	**P-1189**
8.	**P-1191**	**P-1188**	**P-1191**	**P-1191**	P-115	P-0	P-718	**P-1193**	**P-1188**
9.	**P-1192**	**P-1189**	**P-1192**	**P-1192**	P-2515	P-115	P-1325	**P-1188**	**P-1187**
10.	P-717	**P-1190**	P-1948	P-1948	P-962	P-607	P-1522	**P-1187**	**P-1190**

Index	pop	URL	Title
P-0	3	www.gynpages.com	Abortion Clinics OnLine
P-1	2	www.prochoice.org	NAF - The Voice of Abortion Providers
P-2	1	www.cais.com/agm/main	The Abortion Rights Activist Home Page
P-115	2	www.ms4c.org	Medical Students for Choice
P-607	1	www.feministcampus.org	Feminist Campus Activism Online: Welcome
P-717	7	www.nrlc.org	National Right to Life Organization
P-718	2	www.hli.org	Human Life International (HLI)
P-719	5	www.naral.org	NARAL: Abortion and Reproductive Rights: ...
P-962	2	www.prolife.org/ultimate	Empty title field
P-1165	7	www5.dimeclicks.com	DimeClicks.com - Web and Marketing Solutions
P-1184	7	www.amazon.com/...../youdebatecom	Amazon.com–Earth's Biggest Selection
P-1187	6	www.amazon.com/...../top-sellers.html	Amazon.com–Earth's Biggest Selection
P-1188	6	www.amazon.com/.../software/home.html	Amazon.com Software
P-1189	5	www.amazon.com/.../hot-100-music.html	Amazon.com–Earth's Biggest Selection
P-1190	5	www.amazon.com/.../gifts.html	Amazon.com–Earth's Biggest Selection
P-1191	5	www.amazon.com/.....top-100-dvd.html	Amazon.com–Earth's Biggest Selection
P-1192	5	www.amazon.com/...top-100-video.html	Amazon.com–Earth's Biggest Selection
P-1193	7	rd1.hitbox.com/.......	HitBox.com - hitbox web site ....
P-1325	1	www.serve.com/fem4life	Feminists For Life of America
P-1461	5	www.plannedparenthood.org	Planned Parenthood Federation of America
P-1522	1	www.naralny.org	NARAL/NY
P-1769	2	www.priestsforlife.org	Priests for Life Index
P-1948	2	www.politics1.com/issues.htm	Politics1: Hot Political Debates & Issues
P-2515	1	www.ohiolife.org	Ohio Right To Life

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
Kleinberg	10	8	9	9	1	4	1	8	10
pSALSA	8	10	7	7	3	6	3	8	8
HubAvg	9	7	10	10	0	3	0	7	9
AThresh	9	7	10	10	0	3	0	7	9
HThresh	1	3	0	0	10	6	7	3	1
FThresh	4	6	3	3	6	10	4	6	4
BFS	1	3	0	0	7	4	10	3	1
SBayesian	8	8	7	7	3	6	3	10	8
Bayesian	10	8	9	9	1	4	1	8	10

A.2 Query: `+net +censorship (Base Set size = 2947)`

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
1.	**P-375**	**P-371**	**P-371**	**P-371**	**P-375**	**P-375**	**P-371**	**P-371**	**P-375**
2.	**P-3163**	P-1440	**P-2874**	**P-375**	P-1344	**P-371**	**P-375**	P-1440	**P-371**
3.	**P-3180**	**P-375**	**P-2871**	**P-2871**	**P-3132**	P-1344	P-1299	**P-375**	**P-3180**
4.	**P-3177**	**P-2874**	**P-2873**	**P-2874**	**P-3163**	**P-3130**	P-1440	**P-2874**	**P-3177**
5.	**P-3173**	**P-2871**	**P-2659**	**P-3536**	**P-3166**	**P-3131**	**P-2871**	P-1299	**P-3163**
6.	**P-3172**	P-1299	**P-375**	**P-2873**	**P-3167**	**P-3132**	**P-2874**	**P-2871**	**P-3173**
7.	**P-3132**	**P-3536**	**P-3536**	**P-2659**	**P-3168**	**P-3133**	**P-3536**	**P-3536**	**P-3166**
8.	P-3193	P-1712	**P-2639**	**P-2639**	**P-3170**	**P-3135**	**P-1802**	P-1712	P-3193
9.	**P-3170**	P-268	P-1440	P-1440	**P-3171**	**P-3161**	**P-2639**	P-268	**P-3168**
10.	**P-3166**	P-1445	**P-2867**	**P-3180**	**P-3172**	**P-3162**	P-452	P-1445	**P-3132**

Index	pop	URL	Title
P-268	2	www.epic.org	Electronic Privacy Information Center
P-371	7	www.yahoo.com	Yahoo!
P-375	9	www.cnn.com	CNN.com
P-452	1	www.mediachannel.org	MediaChannel.org – A Global Network ....
P-1299	3	www.eff.org/blueribbon.html	EFF Blue Ribbon Campaign
P-1344	2	www.igc.apc.org/peacenet	PeaceNet Home
P-1440	5	www.eff.org	EFF ... - the Electronic Frontier Foundation
P-1445	2	www.cdt.org	The Center for Democracy and Technology
P-1712	2	www.aclu.org	ACLU: American Civil Liberties Union
P-1802	1	ukonlineshop.about.com	Online Shopping: UK
P-2639	3	www.imdb.com	The Internet Movie Database (IMDb).
P-2659	2	www.altavista.com	AltaVista - Welcome
P-2867	1	home.netscape.com	Empty title field
P-2871	5	www.excite.com	My Excite Start Page
P-2873	2	www.mckinley.com	Welcome to Magellan!
P-2874	5	www.lycos.com	Lycos
P-3130	1	www.city.net/countries/kyrgyzstan	Excite Travel
P-3131	1	www.bishkek.su/krg/Contry.html	ElCat. 404: Not Found.
P-3132	4	www.pitt.edu/˜cjp/rees.html	REESWeb: Programs:
P-3133	1	www.ripn.net	RIPN
P-3135	1	www.yahoo.com/.../Kyrgyzstan	Yahoo! Regional Countries Kyrgyzstan
P-3161	1	151.121.3.140/fas/fas-publications/...	Error 404 Redirector
P-3162	1	www.rferl.org/BD/KY	RFE/RL Kyrgyz Service : News
P-3163	3	www.usa.ft.com	Empty title field
P-3166	3	www.pathfinder.com/time/daily	TIME.COM
P-3167	1	travel.state.gov	US State Department - Services - Consular Affairs
P-3168	2	www.yahoo.com/News	Yahoo! News and Media
P-3170	2	www.financenet.gov	...FinanceNet is the government's official home...
P-3171	1	www.securities.com	ISI Emerging Markets
P-3172	2	www.oecd.org	OECD Online
P-3173	2	www.worldbank.org	The World Bank Group
P-3177	2	www.envirolink.org	EnviroLink Network
P-3180	3	www.lib.utexas.edu/.../Map_collection	PCL Map Collection
P-3193	2	www.wiesenthal.com	Simon Wiesenthal Center
P-3536	5	www.shareware.com	CNET.com - Shareware.com

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
Kleinberg	10	1	1	2	6	2	1	1	8
pSALSA	1	10	6	6	1	2	7	10	2
HubAvg	1	6	10	9	1	2	7	6	2
AThresh	2	6	9	10	1	2	7	6	3
HThresh	6	1	1	1	10	3	1	1	5
FThresh	2	2	2	2	3	10	2	2	3
BFS	1	7	7	7	1	2	10	7	2
SBayesian	1	10	6	6	1	2	7	10	2
Bayesian	8	2	2	3	5	3	2	2	10

A.3 Query: Movies (Base Set size = 5757)

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
1.	**P-678**	P-999	P-999	P-999	**P-678**	P-999	P-999	P-999	P-999
2.	**P-2268**	P-2832	P-2832	P-2832	P-2266	P-2832	P-2832	P-2832	P-2832
3.	**P-2304**	P-6359	**P-803**	P-6359	**P-2268**	**P-1989**	P-2827	P-6359	P-2827
4.	**P-2305**	P-2827	**P-1539**	P-2827	P-999	**P-1911**	P-2803	P-2827	P-6359
5.	**P-2306**	**P-2120**	**P-2101**	P-2838	P-2832	**P-1980**	P-5470	**P-1374**	**P-678**
6.	**P-2308**	**P-1374**	**P-1178**	P-6446	**P-2263**	**P-1983**	**P-2120**	**P-2120**	P-2838
7.	**P-2310**	**P-803**	P-6359	P-5	**P-2264**	**P-1984**	**P-4577**	**P-803**	P-2266
8.	P-2266	**P-1539**	P-1082	P-2803	**P-2265**	**P-1986**	P-5	P-6446	**P-2268**
9.	**P-2325**	P-6446	P-2827	P-2839	**P-2280**	**P-1987**	P-2838	**P-1539**	**P-2308**
10.	**P-2299**	P-2838	P-6446	P-2840	**P-2304**	**P-1993**	P-4534	P-2838	**P-2330**

Index	pop	Title	URL
P-5	2	www.movies.com	Movies.com
P-678	3	chatting.about.com	Empty title field
P-803	3	www.google.com	Google
P-999	8	www.moviedatabase.com	The Internet Movie Database (IMDb).
P-1082	1	www.amazon.com/	Amazon.com–Earth's Biggest Selection
P-1178	1	www.booksfordummies.com	Empty title field
P-1374	2	www.onwisconsin.com	On Wisconsin
P-1539	3	206.132.25.51	Washingtonpost.com - News Front
P-1911	1	people2people.com/...nytoday	People2People.com - Search
P-1980	1	newyork.urbanbaby.com/nytoday	Kids & Family
P-1983	1	tunerc1.va.everstream.com/nytoday/	Empty title field
P-1984	1	nytoday.opentable.com/	OpenTable
P-1986	1	www.nytimes.com/.../jobmarket	The New York Times: Job Market
P-1987	1	www.cars.com/nytimes	New York Today cars.com - new and used car ...
P-1989	1	www.nytodayshopping.com	New York Today Shopping - Shop for computers, ...
P-1993	1	www.nytimes.com/.../nytodaymediakit	New York Today - Online Media Kit
P-2101	1	www2.ebay.com/aw/announce.shtml	eBay Announcement Board
P-2120	3	www.mylifesaver.com	welcome to mylifesaver.com
P-2263	1	clicks.about.com/...nationalinterbank	Banking Center
P-2264	1	clicks.about.com/	Credit Report, Free Trial Offer
P-2265	1	membership.about.com/...	Member Center
P-2266	3	home.about.com/movies	About - Movies
P-2268	3	a-zlist.about.com	About.com A-Z
P-2280	1	sprinks.about.com	Sprinks : About Sprinks
P-2299	1	home.about.com/aboutaus	About Australia
P-2304	2	home.about.com/arts	About - Arts/Humanities
P-2305	1	home.about.com/autos	About - Autos
P-2306	1	home.about.com/citiestowns	About - Cities/Towns
P-2308	2	home.about.com/compute	About - Computing/Technology
P-2310	1	home.about.com/education	About - Education
P-2325	1	home.about.com/musicperform	About - Music/Performance
P-2330	1	home.about.com/recreation	About - Recreation/Outdoors
P-2803	2	www.allmovie.com	All Movie Guide
P-2827	6	www.film.com	Film.com Movie Reviews, News, Trailers...
P-2832	8	www.hollywood.com	Hollywood.com - Your entertainment source...
P-2838	5	www.mca.com	Universal Studios
P-2839	1	www.mgmua.com	MGM - Home Page
P-2840	1	www.miramax.com	Welcome to the Miramax Cafe
P-4534	1	www.aint-it-cool-news.com	Ain't It Cool News
P-4577	1	go.com	GO.com
P-5470	1	www.doubleaction.net	Double Action - Stand. Point. Laugh.
P-6359	5	www.paramount.com	Paramount Pictures - Home Page
P-6446	4	www.disney.com	Disney.com – Where the Magic Lives Online!

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
Kleinberg	10	0	0	0	4	0	0	0	4
pSALSA	0	10	7	6	2	2	5	10	5
HubAvg	0	7	10	5	2	2	3	7	4
AThresh	0	6	5	10	2	2	6	6	5
HThresh	4	2	2	2	10	2	2	2	5
FThresh	0	2	2	2	2	10	2	2	2
BFS	0	5	3	6	2	2	10	5	4
SBayesian	0	10	7	6	2	2	5	10	5
Bayesian	4	5	4	5	5	2	4	5	10

A.4 Query: `genetic (Base Set size = 3468)`

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
1.	P-2187	P-2187	P-2187	P-2187	P-2187	P-2187	P-2187	P-2187	P-2187
2.	P-1057	P-258	P-1057	P-1057	P-1057	P-1057	P-3932	P-258	P-1057
3.	P-2168	P-1057	P-3932	P-2168	P-2168	P-2168	**P-1538**	P-1057	P-2168
4.	P-2200	P-3932	P-2095	P-2200	P-2200	P-2200	P-1057	P-3932	P-2095
5.	P-2219	P-2095	P-2168	P-2219	P-2219	P-2219	P-2095	P-2095	P-2200
6.	P-2199	**P-1538**	P-2186	P-2095	P-2199	P-2095	P-258	**P-1538**	P-2219
7.	P-2095	P-2	P-941	P-3932	P-2186	P-3932	P-2168	P-2	P-3932
8.	P-2186	P-2168	P-0	P-2199	P-2095	P-2199	P-2200	P-2168	P-2199
9.	P-2193	P-941	P-2200	P-2186	P-2193	P-2186	P-2	P-941	P-2186
10.	P-3932	P-23	P-2199	P-2193	P-3932	P-2193	P-2199	P-2200	P-2193

Index	pop	URL	Title
P-0	1	www.geneticalliance.org	Genetic Alliance, Washington, DC
P-2	3	www.genetic-programming.org	genetic-programming.org-Home-Page
P-23	1	www.geneticprogramming.com	The Genetic Programming Notebook
P-258	3	www.aic.nrl.navy.mil/galist	The Genetic Algorithms Archive
P-941	3	www3.ncbi.nlm.nih.gov/Omim	OMIM Home Page – Online Mendelian Inheritance in Man
P-1057	9	gdbwww.gdb.org	The Genome Database
P-1538	3	www.yahoo.com	Yahoo!
P-2095	9	www.nhgri.nih.gov	National Human Genome Research Institute (NHGRI)
P-2168	9	www-genome.wi.mit.edu	Welcome To the ..... Center for Genome Research
P-2186	6	www.ebi.ac.uk	EBI, the European Bioinformatics Institute ....
P-2187	9	www.ncbi.nlm.nih.gov	NCBI HomePage
P-2193	5	www.genome.ad.jp	GenomeNet WWW server
P-2199	7	www.hgmp.mrc.ac.uk	UK MRC HGMP-RC
P-2200	8	www.tigr.org	The Institute for Genomic Research
P-2219	5	www.sanger.ac.uk	The Sanger Centre Web Server
P-3932	9	www.nih.gov	National Institutes of Health (NIH)

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
Kleinberg	10	5	8	10	10	10	7	6	10
pSALSA	5	10	6	5	5	5	8	9	5
HubAvg	8	6	10	8	8	8	7	7	8
AThresh	10	5	8	10	10	10	7	6	10
HThresh	10	5	8	10	10	10	7	6	10
FThresh	10	5	8	10	10	10	7	6	10
BFS	7	8	7	7	7	7	10	9	7
SBayesian	6	9	7	6	6	6	9	10	6
Bayesian	10	5	8	10	10	10	7	6	10

A.5 Query: `+computational +geometry (Base Set size = 1226)`

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
1.	P-161	P-161	**P-634**	P-161	P-0	P-161	P-0	P-161	P-161
2.	P-0	P-1	**P-632**	P-0	P-161	P-0	P-161	P-1	P-0
3.	P-1	P-0	**P-633**	P-1	P-1	P-1	P-1	P-0	P-1
4.	P-162	P-3	P-161	P-162	P-162	P-162	P-300	P-3	P-162
5.	P-3	P-280	P-1	P-3	P-280	P-280	P-299	P-280	P-3
6.	P-280	**P-634**	P-1406	P-280	P-3	P-3	P-162	**P-634**	P-280
7.	P-275	P-162	P-0	P-275	P-275	P-275	P-3	P-162	P-275
8.	P-299	P-2	P-3	P-299	P-299	P-299	P-280	P-2	P-299
9.	P-300	**P-632**	P-162	P-300	P-300	P-848	P-375	**P-633**	P-300
10.	P-848	**P-633**	P-280	P-848	P-848	P-300	P-551	**P-632**	P-848

Index	pop	URL	Title
P-0	9	www.geom.umn.edu/software/cglist	Directory of Computational Geometry Software
P-1	9	www.cs.uu.nl/CGAL	The former CGAL home page
P-2	2	link.springer.de/link/service/journals/00454	LINK: Peak-time overload
P-3	9	www.scs.carleton.ca/~csgs/resources/cg.html	Computational Geometry Resources
P-161	9	www.ics.uci.edu/~eppstein/geom.html	Geometry in Action
P-162	9	www.ics.uci.edu/~eppstein/junkyard	The Geometry Junkyard
P-275	5	www.ics.uci.edu/~eppstein	David Eppstein
P-280	9	www.geom.umn.edu	The Geometry Center Welcome Page
P-299	6	www.mpi-sb.mpg.de/LEDA/leda.html	LEDA - Main Page of LEDA Research
P-300	6	www.cs.sunysb.edu/~algorith	The Stony Brook Algorithm Repository
P-375	1	graphics.lcs.mit.edu/~seth	Seth Teller
P-551	1	www.cs.sunysb.edu/~skiena	Steven Skiena
P-632	3	www.w3.org/Style/CSS/Buttons	CSS button
P-633	3	jigsaw.w3.org/css-validator	W3C CSS Validation Service
P-634	3	validator.w3.org	W3C HTML Validation Service
P-848	5	www.inria.fr/prisme/....../cgt	CG Tribune
P-1406	1	www.informatik.rwth-aachen.de/.....	Department of Computer Science, Aachen

	Kleinberg	pSALSA	HubAvg	AThresh	HThresh	FThresh	BFS	SBayesian	Bayesian
Kleinberg	10	6	6	10	10	10	8	6	10
pSALSA	6	10	9	6	6	6	6	10	6
HubAvg	6	9	10	6	6	6	6	9	6
AThresh	10	6	6	10	10	10	8	6	10
HThresh	10	6	6	10	10	10	8	6	10
FThresh	10	6	6	10	10	10	8	6	10
BFS	8	6	6	8	8	8	10	6	8
SBayesian	6	10	9	6	6	6	6	10	6
Bayesian	10	6	6	10	10	10	8	6	10