Application of non-linear Dynamical Systems to Web Searching and Ranking

Panayiotis Tsaparas
Department of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 3G4
tsap@cs.toronto.edu

ABSTRACT

In the recent years there has been a surge of research activity in the area of information retrieval on the World Wide Web, using link analysis of the underlying hypertext graph topology. Most of the algorithms in the literature can be described as a dynamical system, that is, the repetitive application of a function on a set of weights. All algorithms that rely on eigenvector analysis correspond to linear dynamical systems. In this work we examine the application of a non-linear dynamical system for ranking of web pages. We prove that the dynamical system we define converges for any initialization, and we prove that the stationary weights have interesting combinatorial properties. The study of the weights provides a clear and insightful view of the mechanics of the algorithm we propose. We also present extensive experimental results that suggest that our algorithm performs well in practice. Furthermore, we propose our algorithm as a mechanism for finding related pages, and we demonstrate that our algorithm outperforms existing link analysis techniques.

1. INTRODUCTION

In the recent years there has been a surge of research activity in the area of information retrieval on the World Wide Web, using link analysis of the underlying hypertext graph topology. Kleinberg's seminal paper [12] introduced the hubs and authorities paradigm, and proposed the HITS algorithm (Hyperlink Induced Topic Distillation) for ranking web pages. Independently, Brin and Page proposed the PageRank algorithm [5], which become an integral component of the successful Google search engine [9]. These two algorithms spawned the research field of *link analysis ranking*, and they were followed by a substantial amount of research work [4, 3, 13, 16, 2, 1, 15].

Most of link analysis ranking algorithms start with a set of web pages, interconnected with hypertext links. Given the underlying graph, the algorithms extract the principal eigenvector(s) of a matrix associated with the graph, and rank the pages according to the value of each page in this eigenvector (or in the case of multiple eigenvectors, a linear combination of the values of the page in the eigenvectors). Each of these algorithms can be described as a *dynamical system*. A dynamical system assigns an initial weight to each node, and then performs a weight-propagation scheme on the graph. The algorithm iteratively updates the weight of every node to be a function of the weights of other nodes. The final ranking is determined by the stationary weights of the nodes. If the function is a linear function, then the system is called a *linear dynamical system*. All of the algorithms that rely on eigenvector analysis to derive the ranking are linear dynamical systems.

In this work we examine the application of non-linear dynamical systems for ranking of web pages. We define two families of non-linear dynamical systems, and we study in depth a system that is a special case of both of them. Non-linear systems are less popular, since the mathematical tools for analyzing the properties of the systems are not as well developed as in the case of linear systems. We prove that the dynamical system we define has good behavior, that is, it converges for any initialization, and we study the combinatorial properties of the stationary weights of the nodes. The study of the weights provides valuable insight to the algorithm, and reveals a clear and structured mechanism for assigning weights. Furthermore, it suggests a novel algorithm for discovering related pages to a query page. We present extensive experiments of our algorithm both for ranking and for finding related pages. The results indicate that our algorithm performs well, and in many cases outperforms other link analysis approaches. We also study theoretically the properties of our algorithm.

The rest of this paper is structured as follows. Section 2 reviews some of the related work. In Section 3 we present the main concepts about dynamical systems, and we propose a (non-linear) dynamical system for link analysis ranking. In Section 4 we consider the convergence of our algorithm, and the properties of the stationary configuration. Section 5 presents experiments for our ranking algorithm, and Section 6 presents experiments with our technique for finding related pages. Section 7 examines the stability of the algorithm, and its similarity with previously defined algorithms within the framework defined in [4].

2. PREVIOUS ALGORITHMS

In this section we describe in detail the HITS and PageRank algorithms. Most of the algorithms we are interested in can be derived as modifications of these two algorithms.

The HITS algorithm stars by querying a text-based web search engine to obtain a *Root Set* consisting of a short list of web pages relevant to a given query. Then, the Root Set is augmented by including pages which point to pages in the Root Set, and pages which are pointed to by 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 input for HITS algorithm consists of an $N \times N$ adjacency matrix M, where $M_{ij} = 1$ if there are one or more

WWW2003, May 20–24, 2003, Budapest, Hungary.
ACM xxx.

hypertext links from page i to page j, otherwise $M_{ij} = 0$.

The HITS algorithm assigns to each page i an authority weight $a(i)$ and a hub weight $h(i)$. Let $a = (w(1), w(2), \ldots, w(N))$ denote the vector of all authority weights, and $h = (h(1), h(2), \ldots, h(N))$ the vector of all hub weights. Initially all authority and hub weights are set to 1. At each iteration the operations $\mathcal{I}$ ("in") and $\mathcal{O}$ ("out") are performed. The operation $\mathcal{I}$ sets the authority vector to $a = M^T h$, that is, for every authority i,$a(i) = \sum_{j:j \rightarrow i} h(j)$. The operation $\mathcal{O}$ sets the hub vector to $h = Ma$, that is, for every hub j $h(j) = \sum_{i:j \leftarrow i} a(i)$. A normalization step is then applied, so that the vectors a and h become unit vectors in some norm. Kleinberg proves that after a sufficient number of iterations the vectors a and h converge to the principal eigenvectors of the matrices $M^T M$ and MM^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. The goal of the algorithm is to rank the nodes in the graph according to their authority weights.

The PAGERANK algorithm assumes the random surfer model, where a user is following links on a graph, while at some points she performs a jump to a random page. The algorithm assigns a PageRank value to every page. The PageRank of a given web page i, $PR(i)$, can be defined as the limiting fraction of time spent on page i by a random walk which proceeds at each step as follows: With probability ϵ it jumps to a sample from a distribution $D(\cdot)$ (e.g. the uniform distribution) , and with probability $1 - \epsilon$ it jumps uniformly at random to one of the pages linked from the current page.

Both of these algorithms propagate weight on a graph. The main difference is that the PAGERANK algorithm performs a one level propagation of weight (nodes propagate the weight to their immediate neighbors in the graph), while HITS performs a two level propagation of weight (authorities propagate weight through hubs). In HITS there is a mutual re-enforcing relationship between hubs and authorities. Good hubs point to good authorities, and good authorities are pointed by good hubs. In PAGERANK good authorities are pointed to by good authorities.

The HITS and PAGERANK algorithms were followed by a substantial number of variations and enhancements. Most of the subsequent work follows a similar algebraic approach, manipulating some matrix related to the web graph [4, 3, 13, 16, 2, 1, 15]. Recently, there were some interesting attempts in applying statistical, and machine learning tools for computing authority weights [4, 6, 11].

3. DYNAMICAL SYSTEMS

A dynamical system describes a weight propagation scheme on the nodes of a graph. We construct the Base Set as described by Kleinberg [12]. Let P denote a set of nodes, where each node corresponds to a page in the Base Set. We derive the graph G on the set P by placing a directed edge (p, q), if there exists a link from page p to page q.

For the following, we deviate slightly from the terminology used by Kleinberg. We define an *authority node* as a node with non-zero in-degree, and a *hub node* as a node with non-zero out-degree. Note that a node may be both an authority and a hub node. We have that $P = A \cup H$, where A denotes the set of *authority nodes*, and H denotes the set of *hub nodes*. Let n be the size of A. A link analysis algorithm is defined as a function that takes a graph G and returns an n-dimensional real vector w that assigns authority i an *authority weight* $w(i)$.

Now, let G be a graph. We define a *configuration* as an assignment of authority weights to the nodes in the graph. A dynamical system is defined as the repeated application of a function $g : R^n \rightarrow R^n$ on the authority weights, where the function g depends on the graph G. We formally define a dynamical system as follows.

```
Initialize authority weights to $w^0$
Repeat until the weights converge:
    $w^t = g(w^{t-1})$
    Normalize $w^t$ under norm $L$
```

A *stationary configuration* of a dynamical system is a vector u for which $g(u) = u$, that is, a configuration that remains unchanged under the application of g. Our objective is to compute the stationary configuration if one exists, and use the stationary configuration of the dynamical system as the output of the ranking algorithm.

The dynamical system will be fully defined if we specify the function g. The choice of norm L is usually immaterial for the purposes of ranking. The vast majority of link analysis ranking algorithms that have appeared so far in the literature [12, 5, 13, 16, 4, 1, 15] can be described by the above general dynamical system, when the function g is a linear transformation. We call these, linear dynamical systems. For example, if M is the adjacency matrix of the graph G, HITS algorithm corresponds to the case that $g(w) = M^T M w$.

Dynamical systems have also been used for clustering of categorical data [8]. We now consider the class of dynamical systems defined by Gibson et al. [8]. In this class the function g maps a configuration $w = (w(1), w(2), \ldots, w(n))$ to a new configuration $g(w) = (w'(1), w'(2), \ldots, w'(n))$. Assume that we are given a *combiner function* $\oplus$, that takes the set of weights $w_1, w_2, \ldots, w_k$, and returns a new weight $\oplus(w_1, w_2, \ldots, w_k)$. We will define the function $\oplus$ in detail later. Denote by $B(i)$ the set of all nodes that point to authority i, and by $F(j)$ the set of all nodes that are pointed to by hub j. We define the function g as follows.

```
**Function** $g(w)$
  For every hub $j \in B(i)$
      Let $w_1, w_2, \ldots, w_k$ be the weights in $F(j)$
      $h(j) = \oplus(w_1, w_2, \ldots, w_k)$
  For every authority $j \in A$
      $w'(i) = \sum_{j \in B(i)} h(j)$
  return $w'$
```

The value $h(j)$ is called the *hub weight* of the node j. Hub weights serve the purpose of communicating the authority weights between authorities. If the combiner function is a linear function, then the dynamical system is linear. The HITS algorithm corresponds to the case that the combiner function is the addition operation.

We define the S_p family of non-linear dynamical systems indexed by the parameter $p > 0$. Given some value for p, the combiner function $\oplus$ is set to be the p norm of the weight vector of the authorities in $F(j)$. That is, if $w_1, w_2, \ldots, w_k$ are the authority weights of the authorities in $F(j)$, then $h(j) = (w_1^p + w_2^p + \cdots w_n^p)^{1/p}$. This family of dynamical systems was also considered by Gibson et al. [8]. Note that for $p = 1$, S_1 is actually the HITS algorithm.

Another family of a non linear dynamical systems is the Authority Threshold $AT(k)$ family of algorithms defined by Borodin et al. [4]. The $AT(k)$ family is indexed by the parameter k.

Given a value for k, the combiner function $\oplus$ is set to be a threshold function, that sets the hub weight $h(j)$ to be the sum of the k largest authority weights in $F(j)$. We note that in the case that k is equal to the maximum out-degree d_{out} the algorithm $AT(d_{out})$ is the HITS algorithm.

In this paper we consider the dynamical system that we obtain when we set the combiner function $\oplus$ to be the $\max$ operator. We denote by MAX this dynamical system. The MAX algorithm is shown below.

```
MAX ($w^0$)

Initialize authority weights to $w^0$
Repeat until the weights converge:
    For every hub $j \in H$
        $h(j) = \max_{i \in F(j)} w(i)$
    For every authority $i \in A$
        $w(i) = \sum_{j \in B(i)} h(j)$
    Normalize under $L_\infty$ norm
```

The MAX algorithm is a limiting case for the S_p system, when $p = \infty$, as well as a special case for the $AT(k)$, when $k = 1$. We choose the L_∞ norm for the normalization step. Although we can prove that the ratios of weights $w(i)/w(j)$ converge to the same value irrespective of the normalization, our proofs are surprisingly sensitive to the choice of normalization.

3.1 The max operator

The choice of the $\max$ operator as a combiner function may seem somewhat arbitrary, however, there is an intuitive justification for the this choice. Kleinberg [12] treats symmetrically hubs and authorities, and updates both hub and authority weights in the same way. A good hub is one that points to many good authorities, and a good authority is one that is pointed to by many good hubs. However, there is a qualitative difference between the two. A node that is pointed to by many hubs is (intuitively) a good authority, but a hub that points to many authorities is not necessarily a good hub. Consider for example the graph in Figure 1(a). If we run the HITS algorithm on this graph the authorities a, b, c will be ranked higher than authority d. This is due to the fact that the hub x is deemed to be the best hub. However, intuition suggests that authority d is the best authority, and hub x should not be rewarded simply because it points to more authorities of low quality.

In order to deal with this problem Borodin et al. [4] defined the HUB-AVERAGING algorithm that sets the hub weight of a node to be the average of the authority weights pointed by this hub. The underlying intuition is that a good hub should point *only* to good authorities. In the above example the HUB-AVERAGING algorithm will rank authority d first. However, requiring that a good hub points only to good authorities is a very strict condition. Consider for example the graph in Figure 1(b). In this graph authority f seems to be at least as good as authority d, if not better. The HUB-AVERAGING algorithm will rank d above f, because hubs w and v are penalized for pointing to other weaker authorities.

Borodin et al [4] proposed the Authority Threshold algorithm, where the value of a hub is computed according to the top k authority weights of the authorities that it points to. However, setting the value of k depends on the data set. In certain data sets, even small values for k will result in a ranking very similar to that of the HITS algorithm [17]. Furthermore, it is not clear if a hub that points to two average authorities is better than a hub that points to a good authority. The $\max$ operator reflects the intuition that a good hub should point to at least one good authority. That is, a hub is as good as the best authority that it points to. Although this may seem as an extreme choice, the algorithm appears to work well in practice, and it has some very interesting properties.

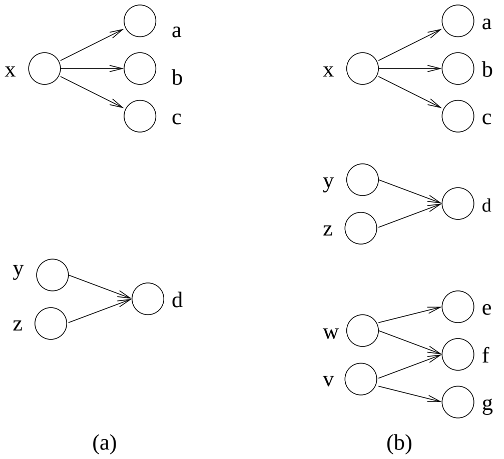

Figure 1: HITS **,** HUB-AVERAGING **and** MAX

4. ANALYSIS OF MAX

We previously stated that dynamical systems can be used as ranking algorithms, where the weights of the stationary configuration of the system are used as the authority weights of the nodes. Therefore, in order for the ranking algorithm to be well defined, we should be able to specify the conditions under which the dynamical system converges. Obviously, a dynamical system for which we have no guarantee about the convergence is an unreliable choice for a ranking algorithm. We would like the system to converge for all graphs, for all initial configurations. Ideally, given a graph, we would like the system to converge to the same stationary configuration for all initial configurations.

When the function g is linear, the stationary configuration of the system corresponds to the eigenvector of some matrix. Eigenvector computations have been studied extensively in the field of linear algebra, and the conditions for convergence are well understood. Things become more complicated when g is non-linear. The mathematics for studying non-linear systems are not as well developed, and usually it is hard to prove the convergence of the system, and even harder to prove something about the stationary configuration.

In the case of the MAX algorithm, we were able to prove the following theorem.

THEOREM 1. *The algorithm* MAX *converges for any initial configuration.*

The proof is presented in [17].

4.1 The stationary configuration

In this section we give a better understanding of the way the MAX algorithm assigns the weights to the authorities. For some node i, let d_i denote the in-degree of authority i, and let $d = \max\{d_i : i \in A\}$, denote the maximum in-degree of any authority. Let S denote the set of nodes with in-degree d. We call these nodes, the *seeds* of the algorithm. We can prove the following lemma.

LEMMA 1. *The stationary configuration of* MAX *depends solely on the stationary weights of the seed nodes.*

Lemma 1 implies that if we run the MAX algorithm and we obtain a set of weights w_S for the seed nodes, then if we set the seed weights to w_S and the non-seed nodes to zero (or to any other values), the MAX algorithm will return the exact same configuration.

For the following we need to introduce some additional terminology. Let i and j be two authorities. Recall that $B(i)$ denotes the hubs that point to authority i, and $F(j)$ the authorities pointed to by hub j. Let $B(i,j) = B(i) \cap B(j)$ denote the set of all hubs that point to both i and j in the graph G. We construct the undirected graph $G_B = (A, E_B)$ on the authority nodes A, by placing an edge between any two authorities i and j, if $B(i,j) \neq \emptyset$, that is, there exists at least one hub in G that points to both i and j. If M is the adjacency matrix of the graph G, then G_B corresponds to the graph with adjacency matrix $M^T M$, after removing all rows and columns that correspond to hub nodes.

Assume now that the graph G_B consists of k connected components $C_1, C_2, \ldots, C_k$. Let w^0 be the weight vector of the initial configuration. The weight assigned by configuration w^0 to component C_i is the sum of weights of all authorities in C_i. We define a *fair* initial configuration as a configuration that assigns non-zero weight to all components in the graph G_B. We will assume that the initial configuration is always fair. This is not a restrictive assumption. If the component C_i is assigned zero weight by the vector w^0, then the weights of the nodes in C_i will immediately converge to zero. Thus, we can disregard the nodes in the component C_i and assume that the algorithm operates on a smaller graph $\tilde{G}$, initialized to a fair configuration $\tilde{w}^0$.

Assume that the algorithm has converged, and let $w(i)$ denote the stationary weight of node i. We define the mapping $f : H \to A$, where the hub j is mapped to authority i, if authority i is the authority with the largest weight among all the authorities pointed by hub j. If there are many authorities in $F(j)$ that have the largest weight, we arbitrarily select one of the authorities (e.g. according to some predefined ordering).

We now introduce the auxiliary graph G_A. Define $H(i) = \{j \in H : f(j) = i\}$ to be the set of hubs that are mapped to authority i. We derive the *directed* weighted graph $G_A = (A, E_A)$ on the authority nodes A as follows. Let i and j be two nodes in A. We place a directed edge from i to j if the following conditions hold.

- $w(i) > w(j)$
- There exists an edge (i,j) in the graph G_B, that is, the two authorities have at least one hub in common.
- $H(i) \cap B(i,j) \neq \emptyset$, that is, at least one of the common hubs of i and j is mapped to the authority i.

The weight $c(i,j)$ of the edge (i,j) is equal to the size of the set $H(i) \cap B(i,j)$, that is, it is equal to the number of hubs that point to both authorities i and j and are mapped to i. The intuition of the directed edge (i,j) is that there are $c(i,j)$ hubs that propagate the weight of node i to node j. The graph G_A captures the flow of weight between the authorities

Now, let $N_{in}(i)$ denote the set of nodes in G_A that point to node i. By definition, the graph G_A is a DAG. Therefore, there must exist some nodes, such that no node in G_A points to them. We define a *source node* in the graph G_A to be a node x, such that $N_{in}(x) = \emptyset$ (i.e., there is no node in G_A that points to x), and $w(x) > 0$. We prove that the set of source nodes, is identical to the set of the seed nodes.

LEMMA 2. *A node is a source node of the graph G_A, if and only if it is a seed node.*

Lemma 2 implies that the seed nodes do not receive any weight from other nodes. Instead, they propagate their own weight to the non-seed nodes.

Now, let i be a non-seed node, and let $c_{in}(i) = \sum_{j \in N_{in}(i)} c(j,i)$, denote the total weight of the edges that point to i in the graph G_A. This is the number of hubs in the graph G that point to i, but are mapped to some node with weight greater than i. These are the hubs that bring in the weight of other nodes. The remaining $d_i - c_{in}(i)$ hubs (if any) are mapped to node i, or to some node with weight equal to the weight of i. These hubs recycle the weight $w(i)$ back to node i. We set $h_i = d_i - c_{in}(i)$. The number h_i is also equal to the size of the set $H(i)$, the set of hubs that are mapped to node i, when all ties are broken in favor of node i. The weight of node i can be computed as a function of the weights of the nodes that point to it as follows

$$w(i) = \frac{1}{d} \left(\sum_{j \in N_{in}(x)} c(j,x) w(j) + h_i w(i) \right) . \tag{1}$$

Now, for some seed node s, let $dist(s,i)$ to be the distance of the longest path in G_A from s to i, and let $dist(i) = \max_{s \in S} dist(s,i)$, to be the maximum distance from a seed node to i, over all seed nodes. We note that the distance is well defined, since the graph G_A is a DAG. Our results about the stationary configuration of the MAX algorithm are summarized in the following theorem. We present the full proof in [17].

THEOREM 2. *Given a graph G, and an initial configuration w, let $C_1, C_2, \ldots C_k$ be the connected components of the graph G_B. For every component C_i, $1 \leq i \leq k$, if component C_i does not contain a seed node, then $w(x) = 0$, for all x in C_i. If component C_i contains a seed node, then every node x in C_i is reachable from the seed node and $w(x) > 0$. Given the weights of the seed nodes, we can recursively compute the weight of a node x at distance $\ell > 0$ using the equation*

$$w(x) = \frac{1}{d - h_x} \sum_{j \in N_{in}(x)} c(j,x) w(j) ,$$

where for all $j \in N_{in}(i)$, $dist(j) < \ell$.

Theorem 2 is in agreement with Lemma 1 which states that the weight of the non-seed nodes depends on the weight of the seed nodes. Note that Theorem 2 does not provide a constructive way of assigning weights to the nodes, since we do not know the graph G_A. However, it provides a useful insight in the mechanics of the algorithm, and in the way the weight is propagated from the seed nodes to the remaining of the authorities. This picture becomes more clear if we set the seed node to the stationary weight, and the non-seed nodes to zero. Then we can think of the weight in the graph as emanating from the seed nodes, and flowing through the graph to the remaining nodes, transferred by the hubs. The weight that a non-seed node receives, depends on the amount of co-citation of this node with the seed nodes, as well as with other non-seed nodes. As the distance from the seed node increases the weight that reaches the non-seed nodes decreases. From Equation 1 and Theorem 2 it is clear that the decrease is exponential. Finally, the weight of a non-seed node depends also on the in-degree of the node. The in-degree affects the term $1/(d - h_i)$ in the weight formula. If there exists a set of hubs that point only to node i, then these hubs recycle the weight of node i, thus increasing the value h_i, thus increasing its weight.

The uniform initial configuration case: In the case of the uniform initial configuration we can show that the seed nodes will converge immediately to weight 1, which is the maximum weight. Therefore, the MAX algorithm will rank the seed nodes first. The rest of the nodes receive less weight than the seed nodes. Their weight is determined from Theorem 2, and depends upon their relation with the seed nodes of the graph, and its own in-degree.

The arbitrary initial configuration case: Lemma 1 guarantees that the stationary configuration depends only on the seed weights. It is easy to show that for all initial configurations that initialize the seed weights to 1, the MAX algorithm converges to the same configuration as when initialized to the uniform configuration. One would hope that all seeds converge to weight 1 for all initial configurations, in which case we would always fall back to the uniform case. Nevertheless, this is not the case. One can construct simple examples of graphs that consist of multiple disconnected components, where, depending on the weight assigned to each component, the algorithm converges to a different configuration. In [17] we present a counter-example, where the graph G_B consists of a single component, yet there exists an initial configuration such that one of the seed nodes converges to weight less than 1. Furthermore, there exist non-seed nodes that have weight greater than the weight of that seed node.

However, we can prove the following proposition for the special class of graphs with a single seed node.

PROPOSITION 1. *For a graph G that contains a single seed node, the algorithm* MAX *converges for any initial configuration to the same stationary configuration as when initialized to the uniform configuration.*

5. EXPERIMENTAL EVALUATION

We experimented with the MAX algorithm on the eight different queries presented by Borodin et al. [4]. The data sets were constructed in the fashion described by Kleinberg [12]. The procedure for constructing the Base Set and the underlying graph is described in detail in [4]. In all data sets the graph G contains a giant component, and a single seed node. This implies that for these data sets the algorithm converges to the same weight vector for all initial configurations that assign non-zero weight to the giant component. The seed node is always ranked at the top.

The full set of experiments can be found at the web page http://www.cs.toronto.edu/∼tsap/experiments/journal-experiments where the MAX algorithm (denoted AThresh(1)) is compared against the algorithms proposed in [4]. The MAX algorithm appears to outperform the rest of the algorithms. Except for the "net censorship" query, for all other queries the MAX algorithm never falls victim of topic drift, and achieves 100% relevance over the top 10 results, something that cannot be claimed for the any of the other algorithms.

In this paper we will present results for the queries "abortion", "genetic" and "net censorship"[1]. For each query we present the top 10 results for the algorithms we consider. We compare MAX against the following algorithms.

- **HITS:** The original algorithm proposed by Kleinberg.
- **IN-DEGREE:** This is a naive algorithm that assigns to each page weight proportional to the in-degree of the page.
- **PAGERANK:** The original PageRank algorithm proposed by Brin and Page. Although the algorithm was proposed for ranking the whole Web, it can be applied to a subset of web pages. In our experiments it appears that the algorithm works best for high values of the parameter ϵ. This is probably due to the fact that the graphs contain multiple dense connected components. Large values of ϵ help the algorithm to avoid getting stuck for a long time in one of these clusters. Usually the value of ϵ is assumed to be low, i.e., $0.1 \leq \epsilon \leq 0.3$. We set $\epsilon = 0.25$ which we consider to be a reasonable value that produces satisfactory results.
- **CO-CITATION:** This algorithm works only in the case that the graph contains a single seed node. The algorithm finds the node with the maximum in-degree, and then assigns to each page weight proportional to the number of hubs that this page has in common with the seed node. From Section 4.1, the weights of the non seed nodes depend upon the amount of co-citation with the seed node. We experiment with the CO-CITATION algorithm in order to study the extend to which co-citation with the seed node determines the ranking produced by the MAX algorithm.

As expected the behavior of the MAX algorithm depends heavily on the quality of the seed node. Table 1 presents the results for the query "abortion". The results in bold denote pages that (according to our judgement) are not relevant to the query. The data set is rather noisy. As a result, the top 10 results for HITS, IN-DEGREE and PAGERANK are infiltrated to some extend by pages that are not relevant to the topic. The HITS algorithm is designed so that it converges to the most tightly knit community (the TKC effect, described by Lempel and Moran). In this case the most dense community consists of a set of pages from "amazon.com". These pages also have high in-degrees and appear high in the ranking of the IN-DEGREE algorithm. Note that PAGERANK does not have the mutual re-enforcement property of the HITS algorithm, so it avoids this cluster of pages, yet it still falls victim to topic drift. On the other hand, for the MAX algorithm the seed node is relevant, and as a result the MAX algorithm produces a set of results that are all relevant to the topic.

The seed page determines also the *focus* of the algorithm. In the presence of multiple communities in the data set, the algorithm will favor the community that contains the seed node. For the "abortion" query the seed node is the home page for "National Right to Life Organization". As a result the top 10 pages are dominated by pages on the pro-life aspect of the issue. This phenomenon becomes more pronounced in the "genetic" query. In this case, there are two communities, one on biology and the genome project, and one about genetic algorithms. These communities are both represented in the top 10 results of the IN-DEGREE and PAGERANK algorithms (the web pages from the genetic algorithms communities appear in italics in Table 2). The seed node for the MAX algorithm is the NCBI (National Center for Biotechnology Information) home page, which results in a set of results that on the biological aspect of the query. Table 4 shows the positions where the first 10 pages about genetic algorithms appear in the ranking of the MAX algorithm. The first page related to genetic algorithms appears in position 36 and the remaining are not in the first 100 results. The next two

[1] In the tables, we sometimes truncate titles and web addresses, so as to fit in the space.

	Max	Hits	In-Degree	Co-Citation	PageRank
1	National Right to Life Organization www.nrlc.org	**DimeClicks.com** **www5.dimeclicks.com**	National Right to Life Organization www.nrlc.org	National Right to Life Organization www.nrlc.org	**The John Birch Society** **www.jbs.org**
2	The Ultimate Pro-Life Resource List www.prolife.org/ultimate	**Amazon.com – youdebate...** **www.amazon.com/...**	*Planned Parenthood Federation* *www.plannedparenthood.org*	The Ultimate Pro-Life Resource List www.prolife.org/ultimate	**About - The Human Internet** **home.about.com**
3	Human Life International (HLI) www.hli.org	**HitBox.com – ...** **rd1.hitbox.com/...**	*Abortion and Reproductive Rights* *www.naral.org*	Human Life International (HLI) www.hli.org	**AllExperts.com** **www.allexperts.com/**
4	Priests for Life Index www.priestsforlife.org	**Amazon.com – Electronics...** **www.amazon.com/...**	**DimeClicks.com** **www5.dimeclicks.com**	Ohio Right To Life www.ohiolife.org	National Right to Life Organization www.nrlc.org
5	*Planned Parenthood Federation* *www.plannedparenthood.org*	**Amazon.com Software** **www.amazon.com/...**	**Amazon.com– youdebate...** **www.amazon.com/...**	Priests for Life Index www.priestsforlife.org	**Law – About Legal News ...** **law.miningco.com**
6	Ohio Right To Life www.ohiolife.org	**Amazon.com– Music...** **www.amazon.com/...**	**HitBox.com – ...** **rd1.hitbox.com/...**	Feminists For Life of America www.serve.com/fem4life	March For Life Fund Home Page www.marchforlife.org
7	*Abortion and Reproductive Rights* *www.naral.org*	**Amazon.com– Gifts ...** **www.amazon.com/...**	**Amazon.com – Electronics...** **www.amazon.com/...**	The Ultimate Pro-Life Resource List www.prolifeinfo.org	**American Opinion Book Services** **www.aobs-store.com**
8	Feminists For Life of America www.serve.com/fem4life	**Amazon.com– DVD ...** **www.amazon.com/...**	**Amazon.com– Music ...** **www.amazon.com/...**	Right to Life Michigan's homepage www.rtl.org	The Ultimate Pro-Life Resource List www.prolife.org/ultimate
9	The Ultimate Pro-Life Resource List www.prolifeinfo.org	**Amazon.com– Video...** **www.amazon.com/...**	**Amazon.com Software** **www.amazon.com/...**	Catholics United for Life www.mich.com/∼ buffalo	Pregnancy Centers Online www.pregnancycenters.org
10	Catholics United for Life www.mich.com/∼ buffalo	**Hot Political Debates** **www.politics1.com/...**	**Amazon.com– Gifts** **www.amazon.com/...**	Pro-Life Action League www.prolifeaction.org	Ariadne's Thread: On abortion... www.lm.com/∼jdehullu

Table 1: Results for the "abortion" query

	Max	Hits	In-Degree	Co-Citation	PageRank
1	NCBI HomePage www.ncbi.nlm.nih.gov	NCBI HomePage www.ncbi.nlm.nih.gov	NCBI HomePage www.ncbi.nlm.nih.gov	NCBI HomePage www.ncbi.nlm.nih.gov	Genetic Alliance www.geneticalliance.org
2	The Genome Database gdbwww.gdb.org	The Genome Database gdbwww.gdb.org	*The Genetic Algorithms Archive* *www.aic.nrl.navy.mil/galist*	The Genome Database gdbwww.gdb.org	NCBI HomePage www.ncbi.nlm.nih.gov
3	National Institutes of Health www.nih.gov	Whitehead Inst. for Genome www-genome.wi.mit.edu	The Genome Database gdbwww.gdb.org	Whitehead Inst. for Genome www-genome.wi.mit.edu	U.S. Department of Health www.os.dhhs.gov
4	Human Genome Research Inst. www.nhgri.nih.gov	Institute for Genomic Research www.tigr.org	National Institutes of Health www.nih.gov	European Bioinformatics Institute www.ebi.ac.uk	Nutritional supplements www.genn.com
5	Whitehead Inst. for Genome www-genome.wi.mit.edu	The Sanger Centre Web Server www.sanger.ac.uk	Human Genome Research Inst. www.nhgri.nih.gov	Institute for Genomic Research www.tigr.org	*The Genetic Algorithms Archive* *www.aic.nrl.navy.mil/galist*
6	Institute for Genomic Research www.tigr.org	UK MRC HGMP-RC www.hgmp.mrc.ac.uk	**Yahoo!** **www.yahoo.com**	National Institutes of Health www.nih.gov	*genetic-programming.org-Home-Page* *www.genetic-programming.org*
7	European Bioinformatics Institute www.ebi.ac.uk	Human Genome Research Inst. www.nhgri.nih.gov	*genetic-programming.org* *www.genetic-programming.org*	UK MRC HGMP-RC www.hgmp.mrc.ac.uk	Genetic Evolutionary Nutrition store.yahoo.com/genn
8	The Sanger Centre Web Server www.sanger.ac.uk	European Bioinformatics Institute www.ebi.ac.uk	Whitehead Inst. for Genome www-genome.wi.mit.edu	The Sanger Centre Web Server www.sanger.ac.uk	US Department of Agriculture www.usda.gov
9	UK MRC HGMP-RC www.hgmp.mrc.ac.uk	GenomeNet WWW server www.genome.ad.jp	Mendelian Inheritance in Man www3.ncbi.nlm.nih.gov/	Human Genome Research Inst. www.nhgri.nih.gov	National Institutes of Health www.nih.gov
10	GenomeNet WWW server www.genome.ad.jp	National Institutes of Health www.nih.gov	*The Genetic Programming Notebook* *www.geneticprogramming.com*	GenomeNet WWW server www.genome.ad.jp	Biological & Environmental Research www.er.doe.gov/...

Table 2: Results for the "genetic" query

	Max	Hits	In-Degree	Co-Citation	PageRank
1	**Yahoo!** **www.yahoo.com**	**CNN.com** **www.cnn.co**	**Yahoo!** **www.yahoo.com**	**Yahoo!** **www.yahoo.com**	**Yahoo!** **www.yahoo.com**
2	**Lycos** **www.lycos.com**	**FT.com Home US** **www.usa.ft.com**	Electronic Frontier Foundation www.eff.org	**My Excite Start Page** **www.excite.com**	**Free B92 - Internet radio station** **www.freeb92.net**
3	**My Excite Start Page** **www.excite.com**	**PCL Map Collection** **www.lib.utexas.edu/...**	**CNN.com** **www.cnn.com**	**Lycos** **www.lycos.com**	EFF Blue Ribbon Campaign www.eff.org/blueribbon.html
4	**CNN.com** **www.cnn.com**	**EnviroLink Network** **www.envirolink.org**	**Lycos** **www.lycos.com**	**CNN.com** **www.cnn.com**	**IETF Home Page** **www.ietf.cnri.reston.va.us**
5	**CNET.com – Shareware.com** **www.shareware.com**	**The World Bank Group** **www.worldbank.org**	**My Excite Start Page** **www.excite.com**	**AltaVista – Welcome** **www.altavista.com**	**Internet Society (ISOC)** **info.isoc.org**
6	**AltaVista – Welcome** **www.altavista.com**	**OECD Online** **www.oecd.org**	EFF Blue Ribbon Campaign www.eff.org/blueribbon.html	**Welcome to Magellan!** **www.mckinley.com**	**About - The Human Internet** **home.about.com**
7	**Welcome to Magellan!** **www.mckinley.com**	**REESWeb: Programs** **www.pitt.edu/...**	**CNET.com – Shareware.com** **www.shareware.com**	**CNET.com – Shareware.com** **www.shareware.com**	**AllExperts.com** **www.allexperts.com/**
8	**The Internet Movie Database** **www.imdb.com**	**Simon Wiesenthal Center** **www.wiesenthal.com**	ACLU: American Civil Liberties Union www.aclu.org	**The Internet Movie Database** **www.imdb.com**	SafeSurf - Making the Net Safe www.safesurf.com
9	Electronic Frontier Foundation www.eff.org	**FinanceNet** **www.financenet.gov**	www.cdt.org www.epic.org	**CNET Search.com** **www.search.com**	**The Amazing SPAM Homepage!** **www.cusd.claremont.edu/...**
10	**Netscape** **home.netscape.com**	**TIME.COM** **www.pathfinder.com/time/**	The Center for Democracy and Technology www.cdt.org	**Netscape** **home.netscape.com**	**Lycos** **www.lycos.com**

Table 3: Results for the "net censorship" query

columns record the positions of the pages in the rankings of In-Degree (In), PageRank (PR) algorithms, in order to get an idea of the importance of these pages. The last column records the position in the ranking of the Co-Citation (Co) algorithm. The number in the parenthesis is the amount of co-citation with the seed node.

Web Page Title	Max	In	PR	Co
The Genetic Algorithms Archive	36	2	5	128 (8)
Artificial Life Online	117	18	72	197 (5)
The Genetic Programming Notebook	118	10	29	234 (3)
genetic-programming.org-Home-Page	121	7	6	232 (3)
IlliGAL Home Page	165	27	56	338 (1)
GAlib: Matthew's Genetic Algorithms Library	168	17	26	315 (1)
The Genetic Algorithms Group	174	24	61	443 (0)
Genetic Algorithms on Proteins	179	127	90	201 (4)
IlliGAL Home	230	31	105	2212 (0)
Genetic Algorithms and Artificial Life Resources	242	77	166	363 (1)

Table 4: Genetic Algorithms community in the Max ranking

From Tables 1 and 2, we observe that there exists significant overlap between the rankings of the Max and Co-Citation algorithms. However, the Max algorithm differs from Co-Citation in the following two aspects. First, the Co-Citation algorithm does not produce a ranking for authorities that have no hubs in common with the seed node. Second, in addition to co-citation with the seed node, the Max algorithm takes also into account the in-degree of the node, and its co-citation with other non-seed nodes. These two differences become obvious in our experimental results. In Table 4, we observe that the pages about genetic algorithms are ranked higher in the ranking of Max , and that there are some pages for which the Co-Citation algorithm does not produce a meaningful ranking since the amount of co-citation with the seed node is zero.

Similar phenomena can be observed for the "abortion" query. For the Max algorithm, although the seed node belongs to the pro-life community, there exist two pro-choice pages in the top 10 results (marked in italics). This is not the case for the Co-Citation algorithm, where the top 10 pages consist entirely of pro-life pages. Table 5 shows the positions of the pro-choice pages in the top 20 pages of Max , and the corresponding positions in the Co-Citation algorithm. For the latter, these pages are spread over the span of 50 pages. Therefore, we may conclude that although co-citation plays an important role in the determination of the weights of the Max algorithm, the algorithm is sophisticated enough to take into account additional properties of the data set.

Web Page Title	Max	Co-Citation
Planned Parenthood Federation of America	5	16
NARAL: Abortion and Reproductive Rights	7	21
Abortion Clinics OnLine	12	45
NAF - The Voice of Abortion Providers	13	33
The Abortion Rights Activist Home Page	16	48
After Abortion: Information ...	20	20

Table 5: Pro-choice community in the Max ranking

We also present a bad case for the Max algorithm. When the seed node is not representative of the topic, the algorithm will drift. Table 3 presents the results of the Max algorithm for the "net censorship" query. In this case the seed node is the "Yahoo" home page. As a result the Max algorithm returns a collection of search engines and news agencies. In defence of our algorithm, this is a query in which all algorithms suffer from topic drift to some extend. Furthermore, the algorithm still manages to produce one relevant page within the top 10. Note that the Co-Citation algorithm does not produce this authority.

6. FINDING RELATED PAGES

The failure of the Max algorithm in the "net censorship" query inspired the following idea. When the seed node was set to be the "Yahoo!" home page, the algorithm was very efficient in discovering all the web pages of search engines. Therefore, if the seed node was actually a page we were interested in, the Max algorithm would have discovered a list of pages *related* to the seed page.

Finding pages related to a query web page is a standard feature of most modern search engines. This is an active research area with a growing literature [12, 7, 10]. The current techniques use content analysis, link analysis, or a combination of both. We propose the use of the Max algorithm as a tool for discovering web pages, related to a query web page.

The idea of using link analysis algorithms for fining related pages was fist suggested by Kleinberg [12]. This idea was later enhanced by Dean and Henzinger [7]. Given a query page q Dean and Henzinger construct a "vicinity graph" around q as follows. Let B denote a step, that follows a link backwards, and let F denote a step that follows a link forward. Starting from the query page q they collect a set of pages that can be reached by following B, F, BF, and FB paths. The vicinity graph is the underlying graph of this set of pages. The authors then propose to run the Hits algorithm, or the Co-Citation algorithm seeded with the query page.

We propose the Max algorithm as a novel alternative for discovering related pages. In order of the algorithm to work, the query page q must be the seed of the algorithm. The rest of the nodes will then be ranked according to their relation to q, where relation is defined naturally from the Max algorithm. However, it is not always the case that the page q is the seed of the vicinity graph. Therefore, we need to engineer the graph, so as to make sure that the page q has the highest in-degree. We go through the nodes of the graph and find the node with the highest in-degree d. We then add enough extra "dummy" nodes in the graph, that point only to node q, so that the in-degree of q becomes greater than d. Thus the page q becomes the seed node for the Base Set. The Max algorithm will assign maximum weight 1 to page q. Following the discussion in Section 4.1 the weight will be diffused from the seed node to the remaining of the graph, through the hubs. The amount of weight that reaches node i measures its relatedness with the seed node.

We note that link analysis by itself is not always sufficient to produce a good set of related pages. Obviously, for a page with no in and out links the algorithm will fail. Moreover, it is important that the query page has high in-degree in the vicinity graph, even if it does not have the maximum in-degree. In this case, more weight is transferred from the seed node to the remaining nodes, and the ranking is more meaningful. However, we will show, that even in the case of a "weak" seed, our algorithm is still able to produce a good set of results.

We present three different experiments for the query pages "www.travelocity.com" (an electronic travel agency – Table 6), "www.allmovie.com" (a site with movie information and movie reviews – Table 7) and "www.cs.toronto.edu/∼bor" (the home page of Allan Borodin – Table 8). We compare our algorithm against the four algorithms we described in Section 5. Further-

	MAX	HITS	IN-DEGREE	CO-CITATION	PAGERANK	GOOGLE
1	Travelocity.com www.travelocity.com	Travelocity.com www.travelocity.com	Travelocity.com www.travelocity.com	Travelocity.com www.travelocity.com	**First Click to the US Government** **firstgov.gov**	Travelocity.com www.travelocity.com
2	Expedia Travel www.expedia.com	Expedia Travel www.expedia.com	Expedia Travel www.expedia.com	Expedia Travel www.expedia.com	Travelocity.com www.travelocity.com	Expedia Travel www.expedia.com
3	MapQuest: Home www.mapquest.com	MapQuest: Home www.mapquest.com	MapQuest: Home www.mapquest.com	MapQuest: Home www.mapquest.com	Travelocity: Last minute deals travelocity.lmdeals.com	MapQuest: Home www.mapquest.com
4	Orbitz: Airline Tickets ... www.orbitz.com	*Yahoo!* *www.yahoo.com*	*Google* *www.google.com*	Orbitz: Airline Tickets ... www.orbitz.com	**The IT Industry Portal** **www.earthweb.com**	Priceline.com www.priceline.com
5	Priceline.com www.priceline.com	*Google* *www.google.com*	Orbitz: Airline Tickets ... www.orbitz.com	Priceline.com www.priceline.com	**University of Wisconsin-Madison** **www.wisc.edu**	Orbitz: Airline Tickets ... www.orbitz.com
6	*Google* *www.google.com*	weather.com - Index www.weather.com	*Yahoo!* *www.yahoo.com*	*Yahoo!* *www.yahoo.com*	**The Industry Standard Archives** **www.thestandard.net**	Trip.com www.trip.com
7	*Yahoo!* *www.yahoo.com*	*CNN.com* *www.cnn.com*	weather.com - Index www.weather.com	*Google* *www.google.com*	Travelocity travelocity.lmdeals.com/...	weather.com - Index www.weather.com
8	weather.com – Index www.weather.com	*Lycos* *www.lycos.com*	Lonely Planet Thorn Tree thorntree.lonelyplanet.com	*CNN.com* *www.cnn.com*	*nz search .. .. SearchNOW.co.nz* *searchnow.co.nz/*	Fodor’s Travel Online www.fodors.com
9	*CNN.com* *www.cnn.com*	*My Excite* *www.excite.com*	Amtrak - ... Train Travel www.amtrak.com	weather.com – Index www.weather.com	**U of Wisconsin-Madison Libraries** **www.library.wisc.edu**	Lonely Planet Online www.lonelyplanet.com
10	Trip.com www.trip.com	*HotBot* *www.hotbot.com*	Priceline.com www.priceline.com	Trip.com www.trip.com	Marriott Rewards www.Marriottrewards.com	AA.com www.aa.com

Table 6: Related pages to “www.travelocity.com”

	MAX	HITS	IN-DEGREE	CO-CITATION	PAGERANK	GOOGLE
1	All Movie Guide www.allmovie.com	All Movie Guide www.allmovie.com	All Movie Guide www.allmovie.com	All Movie Guide www.allmovie.com	The Internet Movie Database www.imdb.com	All Movie Guide www.allmovie.com
2	The Internet Movie Database www.imdb.com	The Internet Movie Database www.imdb.com	The Internet Movie Database www.imdb.com	The Internet Movie Database www.imdb.com	All Movie Guide www.allmovie.com	The Internet Movie Database www.imdb.com
3	Movie Review Query Engine www.mrqe.com	Movie Review Query Engine www.mrqe.com	Movie Review Query Engine www.mrqe.com	Movie Review Query Engine www.mrqe.com	**The Industry Desktop** **www.ifilmpro.com**	AMG All Music Guide www.allmusic.com
4	TV Guide Online - [Movies] www.tvguide.com/movies	TV Guide Online - [Movies] www.tvguide.com/movies	*ABCNEWS.com: Home* *www.abcnews.com*	TV Guide Online - [Movies] www.tvguide.com/movies	The Internet Movie Guide www.ifilm.com	Movie Review Query Engine www.mrqe.com
5	Academy of Motion Pictures www.oscars.org	The Miramax Cafe www.miramax.com	*AltaVista* *www.altavista.com*	Academy of Motion Pictures www.oscars.org	**NewspapersAtoZ.com** **www.classifiedsatoz.com**	Your entertainment source www.hollywood.com
6	*ABCNEWS.com: Home* *www.abcnews.com*	Academy of Motion Pictures www.oscars.org	*Google* *www.google.com*	*ABCNEWS.com: Home* *www.abcnews.com*	**find.info @ Find AtoZ.com** **www.FindAtoZ.com**	Real.com - Guide www.film.com
7	Real.com - Guide www.film.com	FINE LINE FEATURES www.flf.com	Academy of Motion Pictures www.oscars.org	Real.com - Guide www.film.com	**PetsAtoZ** **www.PetsAtoZ.com**	Movie Reviews www.rottentomatoes.com
8	TV Guide Online www.tvguide.com	*ABCNEWS.com: Home* *www.abcnews.com*	Real.com - Guide www.film.com	TV Guide Online www.tvguide.com	**TravelA-Z** **www.TravelA-Z.com**	Non Stop Festivals www.filmfestivals.com
9	Your entertainment source www.hollywood.com	Paramount Pictures www.paramount.com	**newspaper.info** **www.classifiedsatoz.com**	Bright Lights Film Journal www.brightlightsfilm.com	MoviesAtoZ www.moviesatoz.com	The Internet Movie Database www.imdb.com/search
10	The Miramax Cafe www.miramax.com	Bright Lights Film Journal www.brightlightsfilm.com	**find.info @ Find AtoZ.com** **www.FindAtoZ.com**	FINE LINE FEATURES www.flf.com	**RealOne Player** **www.real.com**	Academy of Motion Pictures www.oscars.org

Table 7: Related pages to “www.allmovie.com”

more, we also performed a comparison with the actual Google search engine [9]. In our experiments, for the “travelocity” and “allmovie” queries we added one dummy node, while for the “Borodin” query we needed to add 50 dummy nodes.

It is harder to judge the quality of the results in this setting since the notion of relatedness to a web page can be defined in many different ways. For example we consider Google weakly related to Travelocity since they both relate to search. Using our (subjective) judgement, we classify pages as related, weakly related , and unrelated. In the tables with the results, the weakly related pages appear in italics, while the the unrelated pages appear in boldface.

The first observation is that the link analysis algorithms perform surprisingly well. Compared to the results of Google, a search engine that uses a combination of link analysis, text analysis, and (possibly) user statistics, pure link analysis algorithms are competitive, and in the case of the “Borodin” query they outperform Google. For this query, the top 10 results of the Google search engine contain four (marginally) weakly related results, while the link analysis algorithms output results of higher quality.

The MAX algorithm performs well in all three queries. Although it reports some weakly related pages in the “travelocity”[2] and “allmovie” queries, it never reports completely unrelated pages, and it consistently retrieves highly relevant pages. The best case for the MAX algorithm is the “Borodin” query, where it outputs pages on University of Toronto, a page on the book authored by Allan Borodin, pages from ex-students (Dimitris Achlioptas, Ran El Yaniv) and collaborators of Allan Borodin, as well as researchers with similar background and interests.

From the remaining link analysis algorithms PAGERANK has the worst performance. With the exception of the “Borodin” query, it outputs many unrelated and weakly related pages. It appears that the mutual reinforcement is a desirable property for this type of queries. This is a possible explanation for the improved performance of the HITS algorithm. We are interested in discovering dense clusters around the seed node, and these are exactly the structures that the HITS algorithm tends to favor. However, the TKC effect becomes obvious in the “travelocity” query, where the algorithm favors (weakly related) home pages

[2] For the “travelocity” query, the web pages of “Yahoo”, “Excite”, “HotBot”, and “CNN” were deemed as weakly related. This may be a strict judgement, since all of these pages contain numerous links about travel.

	MAX	HITS	IN-DEGREE
1	Allan Borodin's Home Page www.cs.toronto.edu/∼bor	Allan Borodin's Home Page www.cs.toronto.edu/∼bor	Allan Borodin's Home Page www.cs.toronto.edu/∼bor
2	University of Toronto Home Page www.toronto.edu	Ran El-Yaniv's Home Page www.cs.technion.ac.il/∼rani	University of Toronto Home Page www.toronto.edu
3	Department of Computer Science www.cs.toronto.edu	Anna R. Karlin www.cs.washington.edu/homes/karlin	Department of Computer Science www.cs.toronto.edu
4	Ran El-Yaniv's Home Page www.cs.technion.ac.il/∼rani	Dimitris' Home Page research.microsoft.com/∼optas	University of Toronto Home Page www.utoronto.ca/uoft.html
5	Anna R. Karlin www.cs.washington.edu/homes/karlin	Avrim Blum's home page www.cs.cmu.edu/∼avrim	Hebrew University of Jerusalem www.huji.ac.il
6	Avrim Blum's home page www.cs.cmu.edu/∼avrim	Michael A. Bender's Homepage www.cs.sunysb.edu/∼bender	Online computation and competitive analysis www.cs.technion.ac.il/∼rani/book.html
7	Online computation and competitive analysis www.cs.technion.ac.il/∼rani/book.html	Mark Overmars homepage www.cs.ruu.nl/people/markov	Tel Aviv University www.tau.ac.il
8	Baruch Awerbuch's home page www.cs.jhu.edu/∼baruch	Bernard Chazelle's Home Page www.cs.princeton.edu/∼chazelle	Technion - Israel Institute of Technology www.technion.ac.il
9	Yossi Azar www.math.tau.ac.il/∼azar	Baruch Awerbuch's home page www.cs.jhu.edu/∼baruch	Ran El-Yaniv's Home Page www.cs.technion.ac.il/∼rani
10	Dimitris' Home Page http://research.microsoft.com/∼optas	Erik Demaine db.uwaterloo.ca/∼eddemain	**Web-Counter** **www.digits.com**

	CO-CITATION	PAGERANK	GOOGLE
1	Allan Borodin's Home Page www.cs.toronto.edu/∼bor	Allan Borodin's Home Page www.cs.toronto.edu/∼bor	Allan Borodin's Home Page www.cs.toronto.edu/∼bor
2	Ran El-Yaniv's Home Page www.cs.technion.ac.il/∼rani	University of Toronto Home Page www.toronto.edu	*Papers and presentations* *www.users.csbsju.edu/∼cburch/pub/*
3	Anna R. Karlin www.cs.washington.edu/homes/karlin	University of Toronto Home Page www.utoronto.ca/uoft.html	*ECE311S* *www.control.utoronto.ca/people/profs/ted/ece311s.html*
4	Avrim Blum's home page www.cs.cmu.edu/∼avrim	Online computation and competitive analysis www.cs.technion.ac.il/∼rani/book.html	*3rd Year ECE Course Descriptions - Electrical and Computer* *ww.ece.utoronto.ca/undergrad/courses-3rd-desc.html*
5	Online computation and competitive analysis www.cs.technion.ac.il/∼rani/book.html	Department of Computer Science www.cs.toronto.edu	*ECE310F - Linear Systems and Communications* *www.dsp.toronto.edu/∼anv/ece310f/*
6	F. T. Leighton theory.lcs.mit.edu/∼ftl	Hebrew University of Jerusalem www.huji.ac.il	Ran El-Yaniv's Home Page www.cs.technion.ac.il/∼rani/
7	**WebSTAT** **hits.webstat.com**	Tel Aviv University www.tau.ac.il	Yossi Azar www.math.tau.ac.il/∼azar/
8	Dimitris' Home Page research.microsoft.com/∼optas	University of Toronto - Faculty of Arts and Science www.artsandscience.utoronto.ca	Carl Burch www.users.csbsju.edu/∼cburch/
9	Baruch Awerbuch's home page www.cs.jhu.edu/∼baruch	Technion - Israel Institute of Technology www.technion.ac.il	Online computation and competitive analysis www.cs.technion.ac.il/∼rani/book.html
10	Yossi Azar www.math.tau.ac.il/∼azar	*UT-SFS* *www.utaps.utoronto.ca/financial_aid*	Department of Computer Science www.cs.toronto.edu/

Table 8: Related pages to "www.cs.toronto.edu/∼bor"

of search engines. Also, in the "Borodin" query, HITS focuses on a set of (related) pages of researchers in Theoretical Computer Science, missing pages like the home page of University of Toronto, or the page for the book authored by Allan Borodin.

Another interesting observation is that the IN-DEGREE algorithm performs surprisingly well, indicating that the web pages with high in-degree are related to the query page. However, the limitations of the algorithm become obvious in the "allmovie" query, where it outputs two unrelated, and many weakly related pages, indicating that the in-degree by itself is not sufficient for determining relatedness.

The CO-CITATION algorithm performs well in all three queries, and follows closely the MAX algorithm. For the "travelocity" and "allmovie" queries, the two algorithms have 100% overlap in the top 10 results. However, the qualitative difference of the two algorithms becomes obvious in the "Borodin" query. In this data set, the seed node has very little co-citation with the remaining nodes. Namely, the maximum amount of co-citation is 6, for the home page of Ran El Yaniv, while the home page of University of Toronto, and that of the Department of Computer Science, have just one hub in common with the query home page. This indicates that co-citation cannot always produce meaningful results. At the same time, the MAX algorithm, taking the in-degree into account, ranks the pages about University of Toronto high. while it avoids the unrelated page "www.digits.com", which was ranked within the top 10 by the IN-DEGREE algorithm. Therefore, it appears that the algorithm manages to combine various factors of the data set to discover related pages.

7. STABILITY AND SIMILARITY

We now examine the stability of the MAX algorithm, as well as the similarity of the MAX algorithm with the previously defined algorithms. We work within the framework defined by Borodin et al. [4]. Recall that a ranking algorithm is defined as a function $\mathcal{A}$ that takes a graph G with N nodes and produces a weight vector $\mathcal{A}(G)$. Given two weight vectors w_1, w_2, the authors define two measures of distance. The d_1 distance is the L_1 difference of the two weight vectors, $d_1(w_1, w_2) = \sum_{i=1}^{N} |w_1(i) - w_2(i)|$. The *rank distance* is defined as

$$d_r(w_1, w_2) = \frac{|\{(i,j) : w_1(i) < w_1(j) \text{ and } w_2(i) > w_2(j)\}|}{N(N-1)/2} .$$

The d_1 distance captures the distance between the actual weights assigned by the algorithms, while the rank distance captures the distance between the induced rankings of the algorithms.

Now, let $\mathcal{G}_N$ denote a class of graphs of size N. Borodin et al. [4] give the following definitions for similarity. For the following, we assume that the weight vectors are unit vectors under the L_∞ norm.

DEFINITION 1. *Two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ are similar on*

$\mathcal{G}_N$, if (as $N \to \infty$)

$$\max_{G\in\mathcal{G}_N}\ \min_{\gamma_1,\gamma_2\geq 1}\ d_1\left(\gamma_1\mathcal{A}_1(G),\gamma_2\mathcal{A}_2(G)\right)\ =\ o(N)\ .$$

The algorithms are rank similar on $\mathcal{G}_N$ if

$$\max_{G\in\mathcal{G}_N}\ d_r(\mathcal{A}_1(G),\mathcal{A}_2(G))\ =\ o(1)\ .$$

The constants γ_1,γ_2 serve the purpose of alleviating differences between vectors that are caused due to normalization.

Borodin et al. [4] define also a notion of stability of a ranking algorithm. Given a graph G, a *change* in graph G is defined as an operation ∂ on graph G, that adds and/or removes links so a to produce a new graph $G' = \partial G$. The minimum number of links that we need to add or remove to go from graph G to graph G' is called the size of the change $|\partial|$.

DEFINITION 2. *An algorithm A is stable on $\mathcal{G}_N$ if for every fixed positive integer k, independent of N, we have (as $N \to \infty$)*

$$\max_{G\in\mathcal{G}_N,\partial:|\partial|=k}\ \min_{\gamma_1,\gamma_2\geq 1}\ d_1(\gamma_1\mathcal{A}(G),\gamma_2\mathcal{A}(\partial G)) = o(N)\ .$$

The algorithm $\mathcal{A}$ is rank stable on $\mathcal{G}_N$ if

$$\max_{G\in\mathcal{G}_N,\partial:|\partial|=k}\ d_r(\mathcal{A}(G),\mathcal{A}(\partial G))\ =\ o(1).$$

For details on the definitions of stability, and similarity, we refer the reader to [4].

For the following we use $\overline{\mathcal{G}}_N$ to denote the class of all graphs of size N. We use $\mathcal{G}_N^{AC}$ to denote the class of *authority connected* graphs of size N. An authority connected graph is a graph G such that the corresponding graph G_B consists of a single connected component. We now present the following results. All proofs appear in [17].

PROPOSITION 2. *The* MAX *algorithm is unstable, and rank unstable on $\overline{\mathcal{G}}_N$.*

Furthermore, in the counter-example presented by Lempel and Moran [14] for the rank instability of HITS on $\mathcal{G}_N^{AC}$ the MAX algorithm produces the same ranking as the HITS algorithm. Therefore, we have the following proposition.

PROPOSITION 3. *The* MAX *algorithm is rank unstable on $\mathcal{G}_N^{AC}$.*

We also study the similarity of the MAX algorithm with HITS and IN-DEGREE algorithms. We prove the following (negative) result.

PROPOSITION 4. *The* MAX *algorithm is neither similar, nor rank similar with the* HITS *, and the* IN-DEGREE *algorithms on $\overline{\mathcal{G}}_N$.*

8. CONCLUSIONS

We considered the non-linear dynamical system MAX , and we proved the system converges for any initial configuration. We provided a combinatorial description of the stationary weights assigned by the MAX , and described various interesting properties of the stationary configuration. Finally, we studied the stability and similarity of the MAX algorithm in the model introduced by Borodin et al. [4].

Our work suggests as a possible future research direction the study of other non-linear systems. Note that for the S_p dynamical system, setting $p = 1$ gives the Kleinberg algorithm, while setting $p = \infty$ gives the MAX algorithm. On the other hand, for the $AT(k)$ algorithm, setting $k = 1$ gives the MAX algorithm, while setting $k = \infty$ gives the Kleinberg algorithm. Therefore, for the two extreme values of p and k, we have a very good understanding of how the algorithms S_p and $AT(k)$ behave. We would like to prove that the algorithms converge for all intermediate values of p and k. Furthermore, we would like to understand better how the weight vector changes from one extreme to the other, and the kind of ranking the intermediate algorithms produce. It may be possible that for every data set there exists an "optimal" value for k, for which the $AT(k)$ obtains the best ranking.

9. REFERENCES

[1] D. Achlioptas, A. Fiat, A. Karlin, and F. McSherry. Web search through hub synthesis. In *Proceedings of the 42nd Foundation of Computer Science (FOCS 2001),* Las Vegas, Nevada, 2001.

[2] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proceedings of the 33rd Symposium on Theory of Computing (STOC 2001),* Hersonissos, Crete, Greece, 2001.

[3] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Research and Development in Information Retrieval*, pages 104–111, 1998.

[4] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the World Wide Web. In *Proceedings of the 10th International World Wide Web Conference,* Hong Kong, 2001.

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

[6] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *Proceedings of the 17th International Conference on Machine Learning*, pages 167–174, Stanford University, 2000.

[7] J. Dean and M. R. Henzinger. Finding related pages in the world wide web. In *Proceedings of the Eighth International World-Wide Web Conference (WWW9),* 1999.

[8] D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. In *Proceedings of the 24th Intl. Conference on Very Large Databases (VLDB),* 1998.

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

[10] T. H. Haveliwala, A. Gionis, D. Klein, and P. Indyk. Similarity search on the web: Evaluation and scalability considerations. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB)*, 2002.

[11] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99,* Stockholm, 1999.

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

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

[14] R. Lempel and S. Moran. Rank Stability and Rank Similarity of Web Link-Based Ranking Algorithms. Technical Report CS-2001-22, Technion - Israel Institute of Technology, 2001.

[15] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *Proceedings of the 24th International Conference on Research and Development in Information Retrieval (SIGIR 2001)*, New York, 2001.

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

[17] P. Tsaparas. Analysis of non-linear dynamical systems for ranking. Unpublished manuscript.