QuickRank: A Recursive Ranking Algorithm

Amy Greenwald and John Wicks
Department of Computer Science
Brown University, Box 1910
Providence, RI 02912
{amy,jwicks}@cs.brown.edu

Abstract

This paper presents QuickRank, an efficient algorithm for ranking individuals in a society, given a network that encodes their relationships, assuming that network possesses an accompanying hierarchical structure: e.g., the Enron email database together with the corporation's organizational chart. The QuickRank design is founded on the "peer-review" principle, defined herein, and an hypothesis due to Bonacich. Together, these premises leads to a recursive ranking algorithm which is scalable, parallelizable, and easily updateable. Moreover, it is also potentially more resistant to link-spamming than other popular ranking algorithms.

1 Introduction

A fundamental problem in the field of social network analysis is to rank individuals in a society according to their implicit "importance" (e.g., power or influence), derived from a network's underlying topology. More precisely, given a social network, the goal is to produce a (cardinal) *ranking*, whereby each individual is assigned a nonnegative real value, from which an ordinal ranking (an ordering of the individuals) can be extracted if desired. In this paper, we propose a solution to this problem specifically geared toward social networks that possess an accompanying hierarchical structure.

A social network is typically encoded in a *link graph*, with individuals represented by vertices and relationships represented by directed edges, or "links," annotated with weights. Given a link graph, there are multiple ways to assign meaning to the weights. On one hand, one can view the weight on a link from i to j as expressing the distance from i to j—a quantity inversely related to j's importance. On the other hand, one can view each weight as the level of endorsement, or respect, i grants j—a quantity directly proportional to j's importance. We adopt this latter interpretation.

Under either interpretation (weights as distances or weights as endorsements), a social network can be seen as a collection of judgments, one made by each individual in the society. Correspondingly, we seek a means of aggregating individual judgments into a single collective ranking. In other words, we consider the aforementioned fundamental problem in social network analysis as akin to a key question in voting: how to aggregate the preferences of many individuals into a single collective persuasion that reflects the preferences of the population as a whole.

Given a link graph, perhaps the most basic ranking scheme is degree centrality, in which i's rank is a combined measure of its indegree, the strength of the endorsements i

receives, and outdegree, the strength of the endorsements i makes. It is straightforward to compute this metric. However, it could be argued that it is also sensible to take into account inferred endorsements: e.g., if i endorses j and j endorses k, then i endorses k in a sense. At the opposite end of the spectrum lie ranking schemes that incorporate all such inferred endorsements.

Central to these alternatives is a hypothesis due to Bonacich (1972): *an individual is deemed important if he is endorsed by other important individuals*. In other words, the strength of an endorsement should be construed relative to the rank of the individual making the endorsement. In terms of our voting analogy, Bonacich suggests relating the collective ranking to the sum of all individual judgments, each weighted by its respective rank as determined by the collective. The fixed point of this averaging process—the principal eigenvector of the link graph—defines Bonacich's metric, also known as eigenvector centrality. Although intuitively appealing, the computation of this fixed point can be prohibitive in large networks.

Recently, computer scientists have developed related schemes to rank web pages based on the Web's underlying topology. Viewed as a social network, web pages are individuals and hyperlinks are links. The most prominent approach to ranking web pages is the PageRank algorithm (Page and Brin, 1998; Page et al., 1998), upon which the Google search engine is built. PageRank aggregates the information contained in the Web's hyperlinks to generate a ranking using a process much like Bonacich's method for computing eigenvector centrality.

In this paper, we present QuickRank, an efficient algorithm for computing a ranking in an *hierarchical social network*. Many social networks are hierarchical. One apt example already mentioned is the Web, where the individuals are web pages, the network structure is provided by hyperlinks from one web page to another, and an explicit hierarchical structure is given by the Web's domains, subdomains, and so on. Another fitting example is the Enron email database, where individuals are employees, the network structure is given by emails from one employee to another, and an explicit hierarchical structure is given by the corporate hierarchy. Yet another compelling example is a citation index. In this case, the individuals are publications, the network structure is dictated by the references from one publication to another, and an explicit hierarchical structure is given by the categorization of publications by fields (e.g., computer science), subfields (e.g., AI, theory, and systems), and so on.

As we sketch the key ideas behind the QuickRank algorithm in this introductory section, we allude to the sample hierarchical social network shown in Figure 1, a network of web pages within a domain hierarchy. The web pages, indicated by gray rectangles, are the individuals in this society. Social relationships between these individuals (i.e., hyperlinks between web pages) are shown as dashed lines with arrows. The domain hierarchy is drawn using solid lines with domains and subdomains as interior nodes, indicated by solid black circles, and web pages as leaves (gray rectangles).

Up to normalization, a ranking is a probability distribution. Given any normalized ranking (i.e., probability distribution) of the individuals in an hierarchical social network, by conditioning that global distribution on a particular subcommunity (e.g., CS), we can derive a *conditional* ranking of only those individuals within that subcommunity (e.g., Pr[page 1 | CS], Pr[page 2 | CS], etc.). Likewise, from the respective

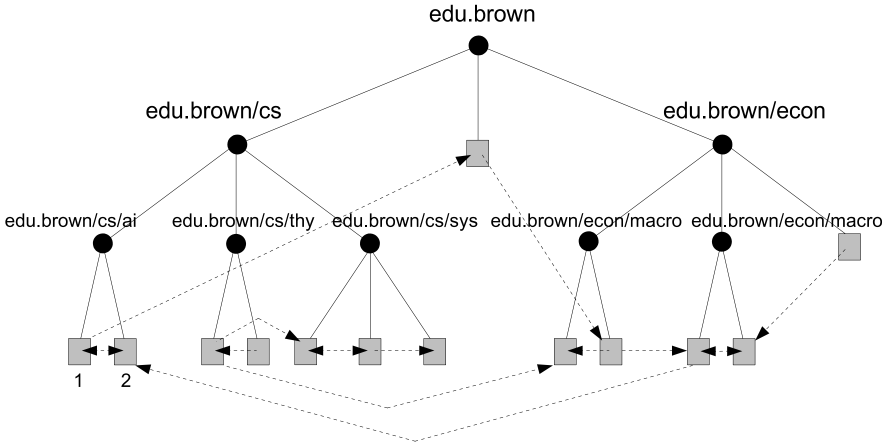


Figure 1: A sample hierarchical social network.

marginal probability of each subcommunity, we can infer what we call a *marginal* ranking[1] of subcommunities themselves (e.g., Pr[AI | CS], Pr[theory | CS], etc.). Conversely, it is straightforward to recover the global ranking by combining the conditional and marginal rankings using the chain rule. For example, Pr[page 1] = Pr[page 1 | AI] Pr[AI | CS] Pr[CS].

Hence, to compute a global ranking of the individuals in an hierarchical social network, it suffices to compute marginal rankings at all interior nodes (i.e., rank the children of all interior nodes), and combine those marginal rankings via the chain rule. To facilitate recursive implementation, QuickRank localizes the computation of each marginal ranking: any links to or from leaves outside the subtree at hand are ignored in such computations. Beyond this computational motivation, localizing marginal ranking computations can be motivated by the following "peer-review principle:" *endorsements among peers (i.e., members of the same subcommunity) should be taken at face value, while other endorsements should be considered as only approximate*.

Intuitively, it is plausible that ranking information among individuals in a tightly-knit community would be more reliable than ranking information among individuals who are only loosely connected. Recall the citation index, a natural example of an hierarchical social network. When a researcher cites a topic in his area of expertise, he is likely to select the most appropriate references. In contrast, if for some reason a researcher with expertise in one area (e.g., computer science) is citing a result in another (e.g., sociology), he may choose only somewhat relevant references. Hence, we contend that the peer-review principle, which justifies localized marginal ranking computations, befits at least some application areas.

[1] Viewing each interior node as the root of a subtree, we informally refer to the ranking of the children of an interior node as a marginal ranking, although such a ranking is technically a *conditional* marginal ranking, conditioned on the subcommunity defined by that subtree.

To fully implement the peer-review principle it is necessary to define some notion of approximate endorsements. To this end, we interpret an endorsement by an individual i in community A for another individual $j \neq i$ in another community $B \neq A$ as comprising part of an endorsement by A of B. More precisely, we aggregate endorsements by individuals in A for individuals in B into an endorsement by A of B by first scaling the endorsements from each i to each j by i's marginal rank, and then summing the resulting weighted endorsements. If we were to replace the target j of an endorsement by any other $j' \in B$, the resulting aggregate endorsement remains unchanged. In this sense, the original endorsement is viewed as "fuzzy" or "approximate." Moreover, by interpreting links originating at i as i's judgment, this aggregation process can be seen as an application of Bonacich's hypothesis (to obtain endorsements of each $j \in B$ by A) followed by a summation over all $j \in B$ (to obtain an endorsement of B).

Together, the principle of peer review and Bonacich's hypothesis lead to the QuickRank algorithm, which we illustrate on the example in Figure 1. We begin by restricting the link graph to, say, the AI subdomain, thereby constructing a local link subgraph. Next, we apply any "flat" ranking scheme (e.g., degree and eigenvector centrality and PageRank) to this link subgraph to produce a marginal ranking of the pages in the AI subdomain (i.e., a distribution over 1 and 2). Then, we scale the links from 1 to 4 and 2 to 3 by the marginal ranks of 1 and 2, respectively, to generate links from AI to 4 and 3. Finally, we sum these results to produce an aggregate link from AI to theory.

Repeating this procedure for the theory and systems subdomains, we "collapse" each of the CS subdomains into a leaf, and substitute these subdomains for their corresponding web pages in the link graph. We then proceed recursively, constructing a local link subgraph, and computing a marginal ranking of the CS subdomains. Combining this marginal ranking with the marginal rankings of the web pages in each CS subdomain yields a single marginal ranking of all the web pages in the CS domain. We repeat this process until the entire hierarchy has been collapsed into a single node, at which point we obtain a ranking of all pages in the `edu.brown` domain.

Overview This paper purports to contribute to the literature on social network analysis by introducing the QuickRank algorithm. As suggested by the previous example, QuickRank is parameterized by a "BaseRank" procedure (i.e., a flat ranking scheme, such as degree centrality) used to compute marginal rankings. We begin in the next section by precisely defining BaseRank procedures and identifying desirable properties of such procedures. In Section 3, we present pseudocode for the QuickRank algorithm. We also consider to what extent QuickRank preserves our previously identified desirable properties of BaseRank procedures. Then, in Section 4, we provide sample QuickRank calculations. Our first example illustrates the distinction between stand alone "BaseRanks" and "QuickRanks," the rankings output by these schemes. A further example shows how QuickRank is potentially more resistant to link-spamming than corresponding BaseRank procedures. We conclude in Section 5. A discussion of related work is deferred to the QuickRank technical report, currently in preparation.

2 A Unified View of Flat Ranking Algorithms

QuickRank is parametized by a flat (i.e., non-hierarchical) ranking algorithm, or a "BaseRank" procedure. In this section, we precisely define a BaseRank procedure, and we formulate the four flat ranking schemes mentioned in the introduction as such. We also present four desirable properties of BaseRank procedures, and discuss to what extent the four aforementioned ranking schemes satisfy these properties.

2.1 Preliminary Definitions

A social network encodes relationships among individuals in a society. Such a network can be represented by a *link graph*. Individuals $i, j \in \mathcal{I}$ are represented as *vertices*, and the fact that individual i relates to individual j is represented by a directed *link* from vertex i to vertex j, augmented by a nonnegative real-valued weight indicating the strength of i's relationship to j.

A *judgment* is a nonnegative, real-valued vector indexed on $\mathcal{I}$. We define an equivalence relation on judgments with r^1 and r^2 equivalent if $cr^1 = r^2$. For our purposes, a *ranking* is such an equivalence class $\langle r \rangle$ (although we often refer to a ranking by any representive of the class). A ranking has exactly one representative that is a probability distribution, which can be obtained by normalizing any other representative. Further, a ranking represents a consistent estimate of the relative merit of pairs of individuals: i.e., for all pairs of individuals i and j, the ranking of i relative to j, namely $\frac{r_i}{r_j} \in [0, \infty]$, is well-defined.

A *link graph* is a nonnegative, real-valued square matrix indexed on $\mathcal{I}$. We restrict attention to the case where the weights in the link graph may reasonably be interpreted as endorsements, rather than distances.[2] A *judgment* graph is a link graph further constrained to have *positive* diagonal entries. Each column in a judgment graph represents the judgment of one individual. The requirement that the diagonal be positive can be interpreted to mean that individuals are required to judge others relative to themselves. Whereas rankings are scale invariant, judgments are scale dependent.

In the introduction, we presented ranking schemes as operating on link graphs. That was a convenient oversimplification. More precisely, they map a judgment graph and a *prior* ranking to a *posterior* ranking. We view the inference of a judgment graph from a link graph as a preprocessing step. This step might consist of inserting self-loops: replacing zeros on the diagonal with ones. In the case of the Web or a citation database, for example, such self-loops would model each web page or publication as implicitly referring to (i.e., endorsing) itself.

Analogously, we define a *BaseRank* procedure as a higher-order function that takes a judgment graph to a mapping which infers a posterior ranking from a prior. When used within the QuickRank algorithm, we require that the posterior ranking output by the BaseRank procedure be normalized to a probability distribution. The prior ranking may be viewed as the persuasion of the "center" (i.e., the implementer of the ranking

[2] It seems conceivable that QuickRank can be suitably modified to handle the distance interpretation by redefining the peer-review notion of approximation as aggregating by taking a minimum instead of summing, but we have not yet explored any applications of this sort.

scheme). A BaseRank procedure then is a means of aggregating the judgments of the individuals in the society, and the center, into a single collective posterior ranking.

Given a judgment graph R and a prior ranking $\langle r \rangle$, Bonacich's hypothesis suggests that we may infer a collective judgment as $r' = Rr$. In this way, individual j's posterior position is the sum of each individual i's conception of j, weighted by the prior rank of i. By ignoring scale in r', we can infer the posterior ranking $\langle r' \rangle$. Note that the result of these two inference steps is well-defined, in that $\langle r' \rangle$ depends only on $\langle r \rangle$ and not on r itself. We use the term *linear* to describe a BaseRank procedure whose mapping from a prior ranking to a posterior abides by Bonacich's hypothesis.

2.2 Sample BaseRank Procedures

We now describe how the four ranking schemes mentioned in the introduction (i.e., indegree, outdegree, eigenvector centrality and PageRank) can be viewed BaseRank procedures. We assume that the link graph has been preprocessed, with self-loops inserted as necessary, to yield an "initial" judgment graph. Since the inference step is fixed, the key step in a linear BaseRank procedure is the way in which a "final" judgment graph is inferred from the initial judgment graph. The degree centrality metrics and PageRank are examples of linear BaseRank procedures, as is eigenvector centrality under certain assumptions (see Theorem 2.2).

The indegree and outdegree of individual i are defined respectively, as follows: given an initial judgment graph R,

$$\textsc{in}(i) = \sum_j R_{ij} \qquad \textsc{out}(i) = \sum_j R_{ji} \tag{1}$$

Both these centrality metrics can be understood as linear BaseRank procedures that infer a posterior ranking from a uniform prior. Indegree is simply the identity function: the initial and final judgment graphs are identical. Outdegree is the transpose operation: the initial and final judgment graphs are transposes of one another.

The PageRank algorithm is parameterized by a value $\epsilon \in (0, 1)$ and a distribution v, often referred to as a "personalization vector." In a preprocessing step, the columns of the judgment graph are normalized to yield a Markov matrix M. PageRank operates on the convex combination of M with the rank one Markov matrix vJ^t (where J ambiguously denotes any vector of all 1's), namely $M_\epsilon = (1 - \epsilon)M + \epsilon v J^t$. This matrix is easily seen to be *regular* (i.e., possessing a single closed class, cf. Wicks and Greenwald (2005)), hence with a unique stable distribution v_∞. Moreover, Haveliwala and Kamvar (2003) have shown that M_ϵ has a second largest eigenvalue of $1 - \epsilon$, so that $\lim_{k\to\infty} M_\epsilon^k v_0 = v_\infty$, for any initial distribution v_0, with convergence as $(1-\epsilon)^k$. This result follows alternatively by writing v_∞ as the limit of a geometric series:

Theorem 2.1 *If M is a Markov matrix and $M_\epsilon = (1 - \epsilon)M + \epsilon v J^t$, then*

$$v_\infty = \lim_{k\to\infty} M_\epsilon^k v_0 = \epsilon \sum_{i=0}^{\infty} (1 - \epsilon)^i M^i v \tag{2}$$

This theorem implies that PageRank is a linear BaseRank procedure, which takes an initial judgment graph M to a final judgment graph $\epsilon \sum_{i=0}^{\infty}(1-\epsilon)^i M^i$. The prior ranking corresponds to the personalization vector and the posterior ranking is a discounted sum of all the inferred rankings (including the prior).

Unlike degree centrality and PageRank, which we have shown are linear BaseRank procedures, eigenvector centrality is not. Given a judgment graph R and an prior ranking v_0, the algorithm infers a sequence of posterior rankings $v_{n+1} = \frac{Rv_n}{\|Rv_n\|_1}$. It can be shown that this sequence eventually converges to a fixed point v_∞, which can be interpreted as the collective ranking. Moreover, this iterative process can be expressed as a linear inference $v_\infty = \frac{R_\alpha v_0}{\|R_\alpha v_0\|_1}$, where α, and hence R_α, depend on the support of v_0. In particular, eigenvector centrality is a *piecewise*-linear BaseRank procedure. In the special case where the judgment graph is strongly-connected (i.e., R is irreducible), eigenvector centrality is linear, because R_α is constant (i.e., independent of α) and v_∞ is independent of v_0. Formally,

Theorem 2.2 *If a judgment graph $R \geq 0$ is irreducible with non-zero diagonal, there exists a unique ranking $v > 0$, such that $\|v\|_1 = 1$ and $Rv = \rho(R)v$, where $\rho(R)$ is the magnitude of the largest eigenvalue of R. Moreover, for any $v_0 \geq 0$, if $v_{n+1} = \frac{Rv_n}{\|Rv_n\|_1}$, $\lim_{n\to\infty} v_n = v$. That is, $v_\infty = v$ and for all α, $R_\alpha = vJ^t$.*

2.3 Generalized Proxy Voting

If we view each individual's rank as a collection of proxy (i.e., infinitely divisible and transferable) votes, then a judgment graph may be interpreted as a *proxy-vote specification* indicating how each individual is willing to assign his proxy votes to others. Given a prior ranking (i.e., an initial allocation of proxy votes), the posterior inferred by a linear BaseRank procedure is a reallocation based on the results of a single round of proxy voting. More generally, in *generalized proxy-voting* (GPV), individuals cast their votes repeatedly over time (i.e., each posterior serves as a prior in the next round), until ultimately, the sequence of posteriors is averaged into a final vote count: i.e., a final ranking.

While historically PageRank has been viewed in terms of a "random-surfer" model (cf. Page et al. (1998)), Theorem 2.1 suggests that it may be more aptly viewed as a GPV mechanism with a discount factor $\gamma \in [0,1)$. In particular, for a given prior ranking v, the posterior computed by PageRank can be expressed as $(1-\gamma)^{-1}\sum_{i=0}^{\infty}\gamma^i M^i v$. Notice that this is just the average of the inferred rankings $M^i v$, where i is distributed geometrically with mean γ. It is natural to generalize to allow weighting by arbitrary distributions, $\sum_{i=0}^{\infty}\alpha_i M^i v$, or even as the limit of such, $\lim_{N\to\infty}\sum_{i=0}^{N}\alpha_{i,N}M^i v$. Formally, we define a generalized proxy-voting mechanism as a (linear) BaseRank procedure that takes an initial judgment graph M into a final judgment graph $\lim_{N\to\infty}\sum_{i=0}^{N}\alpha_{i,N}M^i$.

Observe that all the flat ranking schemes mentioned above, except outdegree, are not only linear BaseRank procedures, but can be seen as GPV mechanisms as well. Indegree is a trivial instance of GPV with $\alpha_{i,N} = \delta_{i,1}$. By Theorem 2.1, PageRank is a GPV mechanism with $\alpha_{i,N} = \epsilon(1-\epsilon)^i$. Finally, if we restrict attention to irreducible judgment graphs, eigenvector centrality is a GPV mechanism, with $\alpha_{i,N} = \begin{cases} \frac{1}{N+1} & \text{if } 0 \le i \le N \\ 0 & \text{otherwise} \end{cases}$. This final claim follows Theorem 2.2 and the well-known fact that $\lim_{i \to \infty} s_i = \lim_{k \to \infty} \frac{1}{k} \sum_{i=0}^{k-1} s_i$. Although outdegree, which takes R to R^t is linear, it is not a GPV mechanism.

2.4 Axioms

Next, we identify two types of judgment graphs that have natural interpretations, and on which a particular behavior for a BaseRank procedure seems preferred. First, consider the identity matrix I as a judgment graph—the *identity* graph—in which each individual ranks himself infinitely superior to all others. Such a ranking graph provides no basis for modifying a prior ranking. Thus, on this input, it seems reasonable that a BaseRank procedure should act as the identity function (i.e., posterior = prior).

Second, consider the case of a *consensus* graph, that is, a judgment graph xy^t, where x is a distribution and y_i is individual i's arbitrary scaling factor. In other words, a consensus graph is a rank 1 matrix: everyone agrees on the ranking x, up to a multiple. Since there is consensus among the individuals in the society, we contend that any prior ranking should be ignored. A BaseRank procedure should simply return the consensus x. We restate these two properties succinctly, as follows:

Identity: $BaseRank(I) = \text{id}$

Consensus: $BaseRank(xy^t) = x$

Another important issue associated with ranking schemes is that of manipulation via "link spamming." The goal of link spamming is to game a ranking system by creating many false nodes, sometimes called sybils (Cheng and Friedman, 2006), that link to some node n, thereby attempting to influence the rank of node n. Web spamming is a particularly popular form of link spamming (Gyongyi and Garcia-Molina, 2004).

A judgment graph inhabited by sybils takes the following form: $M' = \begin{bmatrix} M & \overline{N} \\ 0 & \overline{M} \end{bmatrix}$, where M is the original judgment graph (i.e., without the sybils), $\overline{N}$ describes the links from the sybils to existing members of the society, and $\overline{M}$ describes the links among sybils. Since sybils are new to the community, and hence unknown its original members, we assume that there are no links from those members to sybils.

Observe that generalized proxy-voting mechanisms are spam-resistant in the following sense: Given a prior ranking which places no weight on sybils, the posterior ranking computed with respect to the modified judgment graph M' is, for all intents and purposes, equivalent to the posterior ranking computed with respect to the original judgment graph M. That is,

Property	Indegree	Outdegree	Eigenvector	PageRank
Linear	Yes	Yes	*No*	Yes
GPV	Yes	*No*	Yes	Yes
Identity	Yes	Yes	Yes	Yes
Consensus	Yes	Yes	Yes	*No*

Table 1: Some properties of ranking schemes.

Theorem 2.3 *If* $M' = \begin{bmatrix} M & \overline{N} \\ 0 & \overline{M} \end{bmatrix}$, $v' = \begin{bmatrix} v \\ 0 \end{bmatrix}$, *and* $BaseRank(\cdot) = \lim_{N\to\infty} \sum_{i=0}^{N} \alpha_{i,N} \left(\cdot\right)^i$, *then* $BaseRank(M')v' = \begin{bmatrix} BaseRank(M)v \\ 0 \end{bmatrix}$.

For example, since PageRank is a GPV mechanism, we apply Theorem 2.3 to show that the posterior ranking of non-sybils is unaffected by their presence, if we assign sybils a prior rank of 0. In other words, if sybils can be detected *a priori*, then PageRank may be rendered immune to such an attack. Although the corresponding Markov matrix need not be irreducible for such a "personalization" vector, we conclude from Theorem 2.1 that the Markov process converges for *all* prior rankings v_0. Note that this conclusion follows specifically from our interpretation of PageRank as a GPV mechanism, as opposed to the traditional "random surfer" model.

Table 1 summarizes how each of the four ranking schemes discussed in this section behave with respect to the four properties of BaseRank procedures discussed in this section. PageRank does *not* satisfy the consensus property because it is always biased to some degree by the prior ranking. However, using the notation introduced above, if we instead define $M_\epsilon = (1-\epsilon)M + \epsilon M v J^t$, the resulting algorithm satisfies all four properties. This modified PageRank corresponds to a linear BaseRank procedure with final judgment graph $\epsilon \sum_{i=0}^{\infty} (1-\epsilon)^i M^{i+1}$, that is, the posterior is a discounted sum of all inferred rankings *excluding* the prior.

Fundamentally, QuickRank's design is based on the two key ideas discussed in the introduction, namely the peer-review principle and Bonacich's hypothesis. However, as QuickRank is parameterized by a BaseRank procedure, it is also designed to preserve the Identity and Consensus properties. In the next section, we detail the algorithm and argue informally that it indeed preserves these two properties of BaseRank procedures, although it fails to preserve linearity. When we present sample calculations in Section 4, we note that QuickRank preserves the spam-resistance of its BaseRank procedure, and we illustrate its potential to resist spam even further.

3 QuickRank: The Algorithm

QuickRank operates on a hierarchical social network, that is a judgment[3] graph R whose vertices are simultaneously leaves of a tree T. At a high level, QuickRank first ranks the leaves using the link information contained in the local subgraphs; it then propagates those local[4] rankings up the tree, aggregating them at each level, until they have been aggregated into a single global ranking. Ultimately, *a node's QuickRank is the product of its own local rank and the local rank of each of its ancestors.* QuickRank is parameterized by a BaseRank procedure, which it uses to compute local rankings. It also takes as input a prior ranking of the leaves. It outputs a posterior distribution.

Although we present QuickRank pseudocode (see Algorithm 1) that is top-down and recursive, like many algorithms that operate on trees, the simplest way to visualize the QuickRank algorithm is bottom-up. From this point of view, QuickRank repeatedly identifies "collapsible" nodes in T, meaning the root nodes of subtrees of depth 1, and collapses them into leaf nodes (i.e., subtrees of depth 0) until there are no further opportunities for collapsing: i.e., until T itself is a leaf node. Collapsing node n entails: (i) computing a local ranking at n, that is a ranking of n's children, and (ii) based on this local ranking, aggregating the rankings and the judgments of n's children into a single ranking and a single judgment, both of which are associated with n.

Note that QuickRank is a well-defined algorithm: that is, the order in which local rankings are computed does not impact the global ranking. This property is immediate, since QuickRank propagates strictly local calculations up the tree in computing its global output. Moreover, the collapse operation replaces a subtree of depth 1 with a subtree of depth 0 so that QuickRank is guaranteed to terminate.

Data Structures Algorithm 1 takes as input T_n, subtree of T rooted at node n, and returns two data structures: (i) a ranking of all leaves (with support only on T_n) and (ii) a judgment, which is the average of all judgments of T_n's leaves, weighted by the ranking computed in (i). At leaf node n, the ranking is simply the probability distribution with all weight on n, denoted e_n, and the judgment is given by R_n.

Computing Local Rankings Recall that the main idea underlying QuickRank is to first compute local rankings, and to then aggregate those local rankings into a single global ranking. Given a collapsible node n, a local ranking is a ranking of n's children. To compute such a ranking, QuickRank relies on a BaseRank procedure.

There are two inputs to this BaseRank procedure. The first is n's local (i.e., marginal) prior ranking. The second is a local judgment graph M. For j and k both children of node n, the entry of M in the row corresponding to k and the column corresponding to j is the aggregation of all endorsements from leaves in T_j to leaves in T_k, equal to the sum of all entries in the jth judgment corresponding to leaves of T_k.

Aggregating Rankings and Links To aggregate the rankings of n's m children into a single ranking associated with n, QuickRank averages the rankings $r^1, \ldots, r^m$ ac-

[3] As above, we assume the link graph has been preprocessed to form a judgment graph.

[4] Whereas in the introduction, we used the term marginal, we now use the term local to refer to the ranking of a node's children. The salient point here is: this ranking is computed using strictly local information.

cording to the weights specified by the local ranking r. If we concatenate the m rankings into a matrix $Q = \left[\begin{array}{ccc} r^1 & \cdots & r^m \end{array}\right]$, then the aggregation of rankings can be expressed simply as Qr. Also associated with each child j of a collapsible node n is a judgment l^j. These judgments are aggregated in precisely the same way as rankings.

Algorithm 1 QuickRank(node n)

```
if $n$.isLeaf() then
    return $\langle n$.getJudgment()$, e_n\rangle$
else
    $m$ = $n$.numChildren()
    for $j$ = 1 to $m$ do
        $\langle l^j, r^j\rangle \leftarrow$ QuickRank($n$.getChild($j$))
        for $k$ = 1 to $m$ do
            $M_{kj}$ = Sum($l^j$, $n$.getChild($k$))
        end for
    end for
    $P = \left[\begin{array}{ccc} l^1 & \ldots & l^m \end{array}\right]$
    $Q = \left[\begin{array}{ccc} r^1 & \ldots & r^m \end{array}\right]$
    $r$ = BaseRank($M$, $n$.getLocalPriorRanking())
    return $\langle Pr, Qr\rangle$
end if
```

We now argue that if the BaseRank procedure satisfies the Identity and Consensus properties, then so, too, does QuickRank. First, notice that, when restricted to any subcommunity (i.e., square, diagonal block), an identity or consensus graph yields the same type of graph again. Moreover, aggregating links in such a community within the original graph (i.e., summing rows and averaging columns) also results in the same type of graph. Consequently, if QuickRank employs a BaseRank procedure with the Identity property, it will output the prior distribution on the identity graph, since the prior local rankings will remain unchanged at each level in the hierarchy.

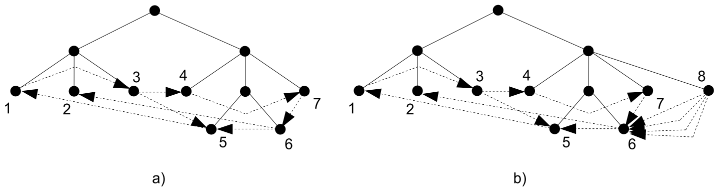
Now consider a consensus graph with ranking x *s.t.* $\|x\|_1 = 1$. Restriction to a subcommunity gives a consensus graph on the corresponding conditional distribution of x. Likewise, aggregation produces a consensus graph on the corresponding marginal distribution of x. If QuickRank employs a BaseRank algorithm with the consensus property on a consensus graph, it will gradually replace the prior distribution at the leaves with the conditional distributions of x, until it finally outputs x itself.

We conclude this section by pointing out that, even if the BaseRank procedure is linear, QuickRank may not be expressible as a linear inference. Normalizing local rankings to form distributions can introduce non-linearities. In the next section, we provide sample QuickRank calculations.

4 Examples

We now present two examples that verify our intuition regarding QuickRank and illustrate some of its novel features. Recall that QuickRank, as it operates on an hierarchical social network (HSN), is parameterized by a prior ranking and a BaseRank procedure.

First, consider the HSN shown in Figure 2a. The hierarchy is drawn using solid lines. The link graph is indicated by dotted lines between the numbered leaves. All weights are assumed to be 1. Computing QuickRanks for this HSN, varying the BaseRank procedure among indegree, eigenvector centrality, and PageRank,[5] but always assuming a uniform prior ranking, leads to the rankings, cardinal and ordinal, shown in Table 2. The values in the posterior distributions have been rounded; hence, the ordinal rankings more precisely reflect the exact values in those distributions.



a)

b)

Figure 2: Two examples of hierarchical social networks.

Table 2: BaseRanks and QuickRanks from Figure 2a and uniform prior.

		Indegree	Eigenvector	PageRank
Flat	cardinal	{0.13, 0.13, 0.13, 0.13, 0.2, 0.13, 0.13}	{0.19, 0.08, 0.16, 0.14, 0.22, 0.10, 0.12}	{0.14, 0.32, 0.11, 0.09, 0.14, 0.09, 0.11}
	ordinal	$5 > 1 = 2 = 3 = 4 = 6 = 7$	$5 > 1 > 3 > 4 > 7 > 6 > 2$	$2 > 1 > 5 > 3 > 7 > 6 > 4$
QuickRank	cardinal	{0.10, 0.10, 0.19, 0.09, 0.23, 0.11, 0.18}	{0, 0, 0.41, 0, 0.59, 0, 0}	{0.04, 0.14, 0.25, 0.04, 0.41, 0.06, 0.06}
	ordinal	$5 > 3 > 7 > 6 > 1 = 2 > 4$	$5 > 3 > 1 = 2 = 4 = 6 = 7$	$5 > 3 > 2 > 7 > 6 > 1 > 4$

For each BaseRank procedure, we list two pairs of rankings: that which results from ignoring the hierarchy, and that which results from exploiting it using QuickRank. When we ignore the hierarchy, all three algorithms rank leaf 1 above (or equal to) 3. However, since 1 defers to 3 (i.e., 1 endorses 3, but not vice versa), based on our peer-review principle, 3 should be ranked higher than 1. This outcome indeed prevails in the QuickRanks, for all three BaseRank procedures.

As an added benefit, QuickRank can be more resistant to link spamming than BaseRank procedures that do not exploit hierarchies. To demonstrate this phenomenon, in Figure 2b, we introduce a sybil, leaf 8, into our original example to try and raise the rank of 6 by recommending it highly. Note the multiplicity of links from 8 to 6.

[5] The results of ranking with outdegree are not qualititatively different, but are omitted for lack of space.

Table 3: Figure 2b with Indegree as BaseRank.

		Uniform Prior	Weighted Prior
Flat	cardinal	$\{0.10, 0.10, 0.10, 0.10, 0.10, 0.35, 0.10, 0.05\}$	$\{0.13, 0.13, 0.13, 0.13, 0.13, 0.2, 0.13, 0.0\}$
	ordinal	$6 > 1 = 2 = 3 = 4 = 5 = 7 > 8$	$6 > 1 = 2 = 3 = 4 = 5 = 7 > 8$
QuickRank	cardinal	$\{0.09, 0.09, 0.18, 0.06, 0.28, 0.14,0.11, 0.06\}$	$\{0.10, 0.10, 0.19, 0.09, 0.23, 0.11, 0.19, 0.0\}$
	ordinal	$5 > 3 > 6 > 7 > 1 = 2 > 4 = 8$	$5 > 3 > 7 > 6 > 1 = 2 > 4 > 8$

Applying QuickRank with indegree as BaseRank to this example yields the rankings shown in Table 3. Using a uniform prior, the sybil is able to raise the rank of 6 over 7 and 6 over 4, whether we exploit the hierarchy (i.e., use QuickRank) or not (i.e., compute indegrees directly). QuickRank cannot prevent this outcome, since the sybil is an accepted member of 4's and 7's community. However, the influence of the sybil is somewhat mitigated under QuickRank. Since the resulting ranking must respect the hierarchy, the effect of the sybil is to raise the ranks of *both* 5 and 6 (i.e., both values in the posterior distribution). No amount of link spam from a sybil outside their local community can increase the rank of 6 relative to 5.

Moreover, if one is able to identify sybils *a priori*, by setting the prior ranks of sybils to zero, one can reduce their influence even further. If we use a prior ranking which is weighted against the sybil, say uniform over 1-7 and zero on 8, Table 3 shows that indegree produces the same rankings as in Table 2, that is, *without* the sybil, whether we exploit the hierarchy or not. In general, Theorem 2.3 states that any BaseRank procedure which is a GPV mechanism will necessarily exhibit this same behavior. QuickRank is not a GPV scheme (recall that QuickRank is nonlinear but that GPV schemes are linear). Still, QuickRank preserves the spam-resistance property characteristic of GPV mechanisms.

5 Conclusion

Social network, or link, analysis is regularly applied to information networks to compute rankings (Garfield, 1972; Kleinberg, 1998; Page and Brin, 1998; Page et al., 1998) and to social networks (Bonacich, 1972; Hubbell, 1965; Katz, 1953; Wasserman and Faust, 1994) to determine standing. We discuss two examples of information networks with inherent hierarchical structure: the Web and citation indices. Social networks, like the Enron email database, also exhibit hierarchical structure. Simon (1962) suggests that such hierarchies are ubiquitous:

> Almost all societies have elementary units called families, which may be grouped into villages or tribes, and these into larger groupings, and so on. If we make a chart of social interactions, of who talks to whom, the clusters of dense interaction in the chart will identify a rather well-defined hierarchic[6] structure.

[6] Simon's use of the terminology "hierarchic" is slightly broader than our use of "hierarchical structure,"

Still, to our knowledge, link analysis procedures largely ignore any hierarchical structure accompanying an information or social network. In this paper, we introduced QuickRank, a link analysis technique for ranking individuals that exploits hierarchical structure. The foundational basis for QuickRank is the peer-review principle, which implies that the relative ranking between two individuals be determined by their local ranks in the smallest community to which they both belong. This principle, together with an hypothesis due to Bonacich, leads to a recursive algorithm which is scalable, parallelizable, and easily updateable.

For a large-scale network such as the Web, we anticipate that QuickRank will yield substantial computational gains over standard ranking methods (e.g., calculating PageRanks via the power method). Moreover, it appears more resistant to link-spamming than other popular ranking algorithms on contrived examples, although it remains to verify this claim empirically.

Acknowledgments

This research was supported by NSF Career Grant #0133689 and NSF Grant #0534586.

References

Phillip Bonacich. Factoring and weighting approaches to status scores and clique detection. *Journal of Mathematical Sociology*, pages 113–120, 1972.

Alice Cheng and Eric Friedman. Manipulability of pagerank under sybil strategies. In *First Workshop on the Economics of Networked Systems (NetEcon06)*, 2006. URL `http://www.cs.duke.edu/nicl/netecon06/papers/ne06-sybil.pdf`.

Eugene Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178:471–479, 1972.

Zoltan Gyongyi and Hector Garcia-Molina. Web spam taxonomy. Technical report, Stanford University Technical Report, 2004.

T. Haveliwala and S. Kamvar. The second eigenvalue of the google matrix. Technical report, Stanford University Technical Report, 2003.

Charles H. Hubbell. An input-output approach to clique identification. *Sociometry*, 28:377–399, 1965.

L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 668–677, 1998.

Lawrence Page and Sergey Brin. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference (WWW7)*, 1998.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

Herbert A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962.

Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.

John R. Wicks and Amy Greenwald. An algorithm for computing stochastically stable distributions with applications to multiagent learning in repeated games. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 623–632, 2005.

by which we mean tree structure. Still, the point remains: hierarchies (or approximations thereof) arise naturally in societies.