Dirichlet PageRank and Ranking Algorithms Based on Trust and Distrust

Fan Chung[⋆], Alexander Tsiatas, and Wensong Xu

Department of Computer Science and Engineering
University of California, San Diego
{fan,atsiatas,w4xu}@cs.ucsd.edu

Abstract. Motivated by numerous models of representing trust and distrust within a network ranking system, we examine a quantitative vertex ranking with consideration of the influence of a subset of nodes. We propose and analyze a general ranking metric, called *Dirichlet PageRank*, which gives a ranking of vertices in a subset S of nodes subject to some specified conditions on the vertex boundary of S. In addition to the usual Dirichlet boundary condition (which disregards the influence of nodes outside of S), we consider general boundary conditions allowing the presence of negative (distrustful) nodes or edges. We give an efficient approximation algorithm for computing Dirichlet PageRank vectors. Furthermore, we give several algorithms for solving various trust-based ranking problems using Dirichlet PageRank with general boundary conditions.

1 Introduction

PageRank has proven to be a useful tool for vertex ranking in many contexts, but some refinements are needed to address many increasingly complex but crucial problems. For example, PageRank is susceptible to manipulation by link spammers, and it treats all links between nodes as positive votes for importance even if some links are meant to show distrust.

To illustrate the need for incorporating several different types of "trust" and "distrust", we consider the following four examples:

Problem 1. Suppose a smaller community holds an election. A community can be represented by a subgraph in a social network, and only edges incident to nodes in the subgraph can be used to determine the ranking of the nodes. The original interpretation of PageRank treats each edge as a vote for determining the "importance" of the nodes. A local community's election should not be influenced by interests outside of the community; one way to deal with this is to set the influence of all outside nodes to be zero. This is the so-called *Dirichlet boundary condition* that we will discuss in this paper.

[⋆] Research supported in part by ONR MURI N000140810747, and AFOR's complex networks program.

Problem 2. In the WWW graph, there are many nodes whose importance is not properly reflected by the link structure of the graph. For example, the websites of some governmental agencies are known to have high impact and authority, but they may not be highly connected to other websites. With prior knowledge of the network, it is often desirable to be able to effectively adjust the ranking of exceptional webpages.

Problem 3. Another factor that many ranking models should address is the notion of "distrust". Distrust can appear in many different ways; for example, if several vertices are known to be spammers, their neighbors are likely to be spammers as well. It is desirable to be able to quantify distrust which can then be used for protection as well as for penalizing spammers. In some cases, distrust between vertices can be built into the graph as negatively weighted edges, presenting computational challenges that the usual PageRank definition does not address. As we will see in later sections, we can use negative links to build a new network with boundary conditions chosen to appropriately propagate distrust.

Problem 4. Another vertex ranking problem arises from a distinction between types of social networks. Some networks, such as Facebook, model closer relationships between people: a social friendship where edges form presumably only between people who know each other personally. This is in direct contrast with systems such as Twitter and Google+ where the act of "following" does not necessarily indicate such a connection. Presumably, a person trusts his or her friends but is interested in "following" not just friends but many others, including celebrities, political figures, acquaintances, corporations, and even enemies.

When agents participate in both networks, it is useful to be able to rank the nodes of a larger network based on the smaller. A node v can compute a ranking on the smaller, more close-knit network and then use this to calculate a ranking of nodes in the larger network that do not appear in the more personal network. Current ranking mechanisms such as PageRank compute a global ranking and therefore are not suitable for this situation.

In this paper, we will show how Dirichlet PageRank can be used to model ranking problems, such as the above four, involving trust and distrust. We will give an efficient algorithm to compute Dirichlet PageRank approximately, which leads to efficient algorithms to solve these problems.

1.1 Related Work

The idea of ranking nodes in a graph has a rich history starting from the introduction of PageRank by Brin and Page [6]. The original PageRank definition was designed for Web search, but many researchers have developed more tailored ranking systems such as personalized PageRank [15, 16] which gives a ranking relative to some specified starting distribution $\boldsymbol{s}$.

One pitfall with PageRank as a ranking system is the fact that all edges contribute positively. In practice, an edge such as a link from one Web page to another can also represent a negative interaction or distrust between the nodes. Several related mathematical models of propagating trust and distrust in a network ranking system are given in [13], and there are numerous empirical results. Another algorithm [14] relies on a small hand-picked set of trusted nodes, but one must be careful not to allow malicious nodes to be included.

There are many other algorithms derived from PageRank that use specific heuristics to model trust or distrust in ranking schemes. [1] considers axioms that a ranking system should satisfy and develops several ranking systems accordingly. [5] and [18] systematically model distrust by modifying the PageRank equations to consider negatively-weighted edges, and [17] gives an algorithm with a similar flavor using random walks. Many of these algorithms are closely related, but rigorous analysis is desired for capturing specific phenomena. We will show that these related models can be represented by Dirichlet PageRank with appropriate boundary conditions.

Another area of research concerns spam nodes when they are identified. It has been shown that if agents can collude [3] or easily create pseudonyms [7], they can artificially boost their ranking in PageRank and other ranking systems. There has been some work done in how to effectively penalize these vertices [4], and our Dirichlet PageRank can be efficiently used to achieve the same goal.

1.2 Results in this Paper

Motivated by the continual development of new PageRank-based algorithms and the analysis of Dirichlet eigenvectors in [10], we develop and analyze Dirichlet PageRank vectors as well as an efficient algorithm to compute them. For a connected graph G, we give a Dirichlet PageRank equation and and show how to compute the unique solution with Dirichlet boundary conditions: $\mathrm{pr}(v) = 0$ for vertices v on the boundary of a specified vertex subset S.

After giving the algorithm for computing Dirichlet PageRank vectors, we generalize the boundary conditions to arbitrary values $\mathrm{pr}(v) = \sigma(v)$ for boundary vertices v. We give an efficient algorithm ApproxDirichPR to compute approximate Dirichlet PageRank vectors with any boundary condition σ. We also give a full analysis leading to the following theorem. We use the notation that $|\cdot|$ denotes the L_1-norm, and for a subset S of vertices in G, the volume of S is denoted by $\mathrm{vol}(S) = \sum_{v \in S} d_v$ where d_v is the degree of v in the graph G. Detailed definitions will be given in Section 2.

Theorem 1. *For any $\epsilon \in (0,1)$ and any teleportation constant $\alpha \in (0,1)$, the algorithm ApproxDirichPR outputs an ϵ-approximate Dirichlet PageRank vector $\widetilde{\mathrm{pr}}_S$ in time $O(\frac{\mathrm{vol}(S) \log \frac{1}{\epsilon}}{\alpha})$, which, compared to the exact Dirichlet PageRank pr_S, satisfies:*

$$|\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S| < \frac{\epsilon \mathrm{vol}(S)}{\alpha}.$$

We illustrate several applications of Dirichlet PageRank with boundary conditions below. Many of the specific PageRank variations are covered by this general framework, and we will show how its use can allow the efficient consideration of several models in [1, 4, 5].

1.3 Several applications of Dirichlet PageRank

• **Allowing negative edges in the graph.** While trust between two vertices is denoted by a positive weight, it is natural to quantify distrust as negative weights for the associated edges. There are many other types of relations in a network that can be represented with negative edges as well, and the usual PageRank vectors do not consider negative weights. We will use Dirichlet PageRank as a tool to deal with graphs containing negative edges in Section 6.1.

• **Diminishing known spammers' influence.** Many Web pages can be identified as spammers based on content or user reports. It is desirable to have a network ranking scheme that takes such considerations into account by penalizing both the known spammer nodes and others with many links to them. We will show that Dirichlet PageRank is useful for dealing with spammer nodes in Section 6.2.

• **Considering trusted friends' opinions.** A single node in a graph may have a set of trusted friends or neighbors whose opinions need to be considered strongly in designing a vertex ranking scheme. If these trusted nodes have their own independent ranking opinions, we can use Dirichlet PageRank with appropriate boundary conditions to compute a trust-based ranking. We will give an algorithm PRTrustedFriends for this problem in Sectoin 6.3.

• **Validating ranking for newly-created nodes.** Suppose that a new person enters a social network but is unsure about which nodes are trustworthy. Personalized PageRank is a useful tool for deriving quantitative information, but it raises the question of whether this ranking is susceptible to unknown spammers. Without a specific set of trusted friends, it may seem hopeless for the newcomer, but Dirichlet PageRank with boundary conditions can be used to validate and adjust its ranking with a randomly-selected pool of established nodes within the network. We will give the details for an algorithm PRValidation in Section 6.3.

• **Reconciling ranking in personal and global social networks.** Several interesting questions arise when analyzing different types of social networks. Some, like Facebook, offer a more personal viewpoint as reflected in the network structure, where edges are formed only by mutual consent between two people who usually know each other. This is in contrast with a network such as Twitter, where the connections are often (though not always) impersonal. For example, people "follow" each other based not only on friendship, but also on subject matters, celebrity appeal, advertising, and many other conceivable reasons.

With the vast array of information available on a network such as Twitter, it is important for a user to know who is trustworthy or worth following. This is a difficult problem, but a user does have some information at hand, such as its own, more close-knit, smaller social networks or even a trusted subgraph of the larger network. Using Dirichlet PageRank, a user can compute a ranking on the smaller network, and then use boundary conditions appropriately to infer a ranking on the remaining nodes in the larger network, taking its personal associations into account. We will develop an algorithm PRTrustNetwork for this problem in Section 6.3.

Finally, we can use similar ideas to tackle the problem in the reverse direction: Suppose a global ranking of the nodes in a larger, loose social network such as Twitter is known, and a user wants to develop a personalized ranking for a small subgraph or its own trusted network taking the global ranking into account. We again can use Dirichlet PageRank with appropriate boundary conditions, as outlined in an algorithm PRInferRanking, also in Section 6.3.

The rest of the paper proceeds as follows. In Section 2, we outline necessary background on PageRank and Dirichlet boundary conditions. Section 3 develops the theory of PageRank with Dirichlet boundary conditions, and Section 4 extends this theory for arbitrary boundary conditions σ. We develop and analyze the ApproxDirichPR algorithm in Section 5 and give algorithms for the previously-discussed applications in Section 6.

2 Preliminaries

For a connected undirected graph $G = (V, E)$ with n vertices and m edges, let A be the *adjacency matrix* and D be the *diagonal degree matrix* where D_{ii} is the degree of the i-th vertex. The typical random walk on G is defined by the *transition probability matrix* $D^{-1}A$. In this paper, we consider the *lazy random walk* which is defined by the *lazy transition probability matrix*

$$W = \frac{1}{2}(I + D^{-1}A).$$

The *normalized Laplacian* $\mathcal{L}$ is defined by:

$$\mathcal{L} = D^{-1/2}(D - A)D^{-1/2} = I - D^{-1/2}AD^{-1/2}.$$

The normalized Laplacian $\mathcal{L}$ and its spectrum are useful for analyzing graphs and their associated random walks (see [9] for more details).

The *restricted Laplacian* $\mathcal{L}_S$ is the submatrix of $\mathcal{L}$ restricted to $S \times S$. The *restricted Green's function* $\mathcal{G}_{S,\beta}$ is defined by:

$$\mathcal{G}_{S,\beta}(\beta I_S + \mathcal{L}_S) = I_S,$$

where $\beta \geq 0$. Note that $\mathcal{L}_S$ is positive definite [10], so $\mathcal{G}_{S,\beta}$ is well-defined.

The *PageRank vector* pr has two parameters: the *teleportation constant* $\alpha \geq 0$ and the *seed vector* s. For given α, s, the PageRank pr is defined to be the unique solution to the PageRank equation:

$$\mathrm{pr} = \alpha s + (1 - \alpha)\mathrm{pr}W, \tag{1}$$

where we treat s and pr as row vectors. This paper uses the lazy random walk transition matrix W instead of regular random walk transition matrix $D^{-1}A$, but the two PageRank definitions are equivalent up to a change in teleportation constant α (see [2]). PageRank was first introduced in [6] to measure the importance of Web pages (with the seed vector $s = \mathbf{1}/n$), and it has since been applied to many problems, including the measurement of trust in social networks [1]. Fast approximation algorithms for PageRank can be found in [2, 11].

We can also write PageRank as a geometric sum of random walks [2]:

$$\mathrm{pr} = \alpha s + \alpha \sum_{t=1}^{\infty} (1-\alpha)^t s W^t .$$

This allows us to see that pr, with the same teleportation constant α, is linear in its seed vector s.

3 PageRank with Dirichlet Boundary Conditions

Let S denote a subset of the vertex set V of G. The *volume* $\mathrm{vol}(S)$ denotes the sum of the degrees of vertices in S. The *vertex boundary* $\delta(S)$ is defined by:

$$\delta(S) = \{u | u \notin S \text{ and } \exists v \in S, (v, u) \in E\}.$$

The essential question proposed in Problem 1 is how to take into account the boundary edges. We remark that in the classical areas of differential equations defined on some geometric spaces, the boundary conditions are referred to the constraints defined on the boundaries of specified regions and are imposed on the solutions of the equations. For the PageRank equation, the lazy random walk transition matrix W is closely related to the normalized Laplacian $\mathcal{L}$ which is the analog of the Laplace-Bertromi operator in differential geometry. Thus, the PageRank equation (1) can be regarded as a discrete analog to a set of differential equations, and Dirichlet PageRank can be viewed as a solution of the PageRank equation with boundary conditions. The basic problem of deriving PageRank vectors with Dirichlet boundary conditions was also previously examined in [8].

Let S be a subset of G; for a function (or vector) $f : V \to \mathbb{R}$, we say f satisfies the *Dirichlet boundary condition* if

$$f(v) = 0 \text{ for all } v \in \delta(S).$$

The PageRank vector satisfying the Dirichlet boundary condition is the solution for the following equation, for all vertex v:

$$\mathrm{pr}(v) = \begin{cases} \alpha s(v) + (1-\alpha)\sum_{u \in V} \mathrm{pr}(u) W_{uv} & \text{if } v \in S \\ 0 & \text{otherwise .} \end{cases} \tag{2}$$

Let pr_S and s_S denote the vectors pr and s restricted to S, respectively. Similarly, let W_S, D_S, A_S denote the respective matrices restricted to $S \times S$.

Theorem 2. *For a connected graph G, vector s, and $\alpha > 0$, the above PageRank equation (2) has one and only one solution. With $\beta = \frac{2\alpha}{1-\alpha}$, it is given by*

$$\mathrm{pr}_S = \beta s_S D_S^{-1/2} \mathcal{G}_{S,\beta} D_S^{1/2}.$$

Proof. Since $\mathrm{pr}(v) = 0$ when $v \notin S$, the PageRank equation (2) is equivalent to

$$\mathrm{pr}_S = \alpha s_S + (1-\alpha)\mathrm{pr}_S W_S.$$

Since

$$W = \frac{1}{2}(I + D^{-1}A) = I - \frac{1}{2}(D^{-1/2}\mathcal{L}D^{1/2}),$$

and D is diagonal matrix, we have

$$W_S = I_S - \frac{1}{2}(D_S^{-1/2}\mathcal{L}_S D_S^{1/2}).$$

Thus, we have

$$\mathrm{pr}_S = \alpha s_S + (1-\alpha)\mathrm{pr}_S(I_S - \frac{1}{2}(D_S^{-1/2}\mathcal{L}_S D_S^{1/2})).$$

Solving for pr_S yields the theorem; uniqueness follows from the uniqueness of $\mathcal{G}_{S,\beta}$. □

Using Dirichlet PageRank vectors with Dirichlet boundary conditions, we can solve Problem 1 in a straightforward way, and it is clear that vertices outside of S will not influence the ranking. However, it is not immediately apparent as to how the boundary edges affect the ranking result. In Figures 1-3, comparisons are given between rankings obtained by three methods. For a small graph, we compute the Dirichlet PageRank vector and compare it with the following two alternative methods:

Method 1 Compute the PageRank for the entire graph and simply use these values on the subgraph.

Method 2 Delete the rest of the graph including boundary edges, then compute the PageRank for the remaining induced subgraph.

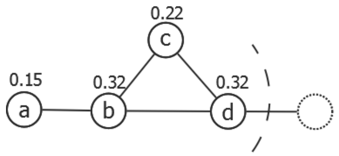


Fig. 1: Ranking computed by Method 1

We note that the difference between Method 1 and the Dirichlet PageRank is the relative ranking of vertices b and d. Using Method 1, b and d have the same ranking, while with the Dirichlet PageRank, b has a higher ranking. This is consistent with the fact that in this subgraph, b is trusted by all other vertices, while d is trusted by only two out of three other nodes. Note that d is also trusted

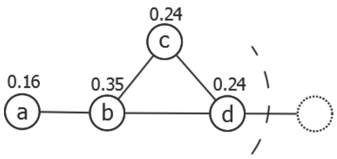


Fig. 2: Ranking computed by Method 2

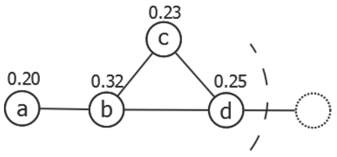


Fig. 3: Ranking computed by Dirichlet PageRank

by another vertex outside of the subgraph, but the influence of the outside vertex is not taken as significantly.

The difference between Method 2 and Dirichlet PageRank is the relative ranking of vertices c and d. By Method 2, c and d have the same ranking; by the Dirichlet PageRank, d has a slightly higher ranking. This is consistent with the fact that c and d trust each other and are trusted by b. In addition, d is trusted by one vertex from outside, reflected by a higher ranking of d.

We give another Dirichlet PageRank example in Fig. 4. In Fig. 4a, a social network [20] have nodes colored according to their PageRank (for the case of $\alpha = 0.1$). Now, suppose we have identified two spammers and want to penalize their ranking and influence, as described in Problem 3. In Fig. 4b, we simply compute the usual PageRank and set the two spammers' rank to zero. This is equivalent to the ranking algorithm proposed in [14]. In Fig. 4c, we compute the Dirichlet PageRank with the boundary condition $\sigma(u) = \sigma(v) = 0$ for the spammers u and v. It is apparent that the rankings of the nodes surrounding the spammers have been decreased, illustrating the effects of propagation of distrust.

These two examples serve as an illustration of the contributions of the boundary edges to the ranking, made more rigorous in the following lemma.

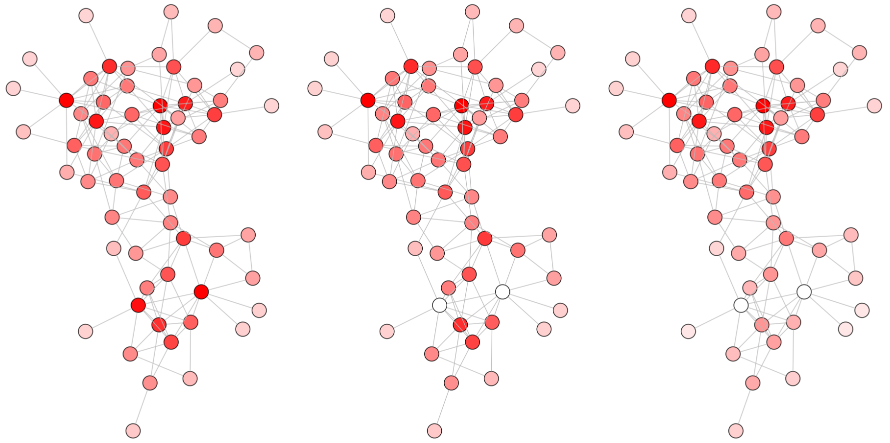
Let pr' be the PageRank vector computed by Method 1 and let pr'' denote the PageRank computed by Method 2. It is easy to see that for every $v \in S$, $\mathrm{pr}(v) \leq \mathrm{pr}'(v)$.

We define two vertex sets S_o and S_i:

$$S_o = \{v \in S | \exists u \notin S : (u, v) \in E\}, \quad \text{and} \quad S_i = S \setminus S_o.$$

Let W_{ii} and W_{oi} denote W restricted to $S_i \times S_i$ and $S_o \times S_i$, respectively. Similarly, we define two vectors w_o and w_i:

$$w_o = (1 - \alpha)\mathbf{1}W_{oi}^T, \quad \text{and} \quad w_i = \mathbf{1} - (1 - \alpha)\mathbf{1}W_{ii}^T.$$



(a) PageRank of the graph with $\alpha = 0.1$.

(b) PageRank of the graph, with two "spammers' identified with zero rank.

(c) Dirichlet PageRank of the graph, with the nodes around the two "spammers" showing decreased rank.

Fig. 4: Three rankings of a network [20]. Darker shades of red indicate higher ranking.

Lemma 1. *Suppose that both S_o and S_i are non-empty. Then we have*

$$\frac{\mathrm{pr}'_{S_o} w_o^T}{\mathrm{pr}'_{S_i} w_i^T} \geq \frac{\mathrm{pr}_{S_o} w_o^T}{\mathrm{pr}_{S_i} w_i^T} \geq \frac{\mathrm{pr}''_{S_o} w_o^T}{\mathrm{pr}''_{S_i} w_i^T}$$

Proof. Let W'' denote the lazy random walk transition probability matrix of G_S, where G_S is the subgraph of G restricted to S. Then the following equations hold:

$$\mathrm{pr}'_{S_i} = \alpha s_{S_i} + (1-\alpha)\left(\mathrm{pr}'_{S_i} W_{ii} + \mathrm{pr}'_{S_o} W_{oi}\right), \tag{3}$$

$$\mathrm{pr}_{S_i} = \alpha s_{S_i} + (1-\alpha)\left(\mathrm{pr}_{S_i} W_{ii} + \mathrm{pr}_{S_o} W_{oi}\right), \tag{4}$$

$$\mathrm{pr}''_{S_i} = \alpha s_{S_i} + (1-\alpha)\left(\mathrm{pr}''_{S_i} W''_{ii} + \mathrm{pr}''_{S_o} W''_{oi}\right). \tag{5}$$

Let $c_1 = \alpha s_{S_i} \mathbf{1}^T$; the definitions of w_o and w_i give:

$$\frac{\mathrm{pr}_{S_o} w_o^T}{\mathrm{pr}_{S_i} w_i^T} = \frac{\mathrm{pr}_{S_i} w_i^T - c_1}{\mathrm{pr}_{S_i} w_i^T}.$$

Subtracting (4) from (3) yields:

$$\left(\mathrm{pr}'_{S_i} - \mathrm{pr}_{S_i}\right)\left(I - (1-\alpha)W_{ii}\right) = \left(\mathrm{pr}'_{S_o} - \mathrm{pr}_{S_o}\right)\left((1-\alpha)W_{oi}\right).$$

Let $c_2 = \left(\mathrm{pr}'_{S_i} - \mathrm{pr}_{S_i}\right)\left(I - (1-\alpha)W_{ii}\right)\mathbf{1}^T$. Since for all $v \in S, \mathrm{pr}(v) \leq \mathrm{pr}'(v)$, it follows that $\mathrm{pr}'_{S_i} - \mathrm{pr}_{S_i}$ and c_2 are both nonnegative. Thus, we have

$$\frac{\mathrm{pr}'_{S_o} w_0^T}{\mathrm{pr}'_{S_i} w_i^T} = \frac{\mathrm{pr}'_{S_i} w_i^T - c_1}{\mathrm{pr}'_{S_i} w_i^T} = \frac{\mathrm{pr}_{S_i} w_i^T + c_2 - c_1}{\mathrm{pr}_{S_i} w_i^T + c_2}.$$

Since for every $v \in S_o$, the degree of v in G_S is strictly smaller that the degree of v in G, we have $W''_{v,u} \geq W_{v,u}(1+\frac{1}{d})$, where d is the maximum degree among vertices in S_o in G_S. For $v \in S_i$, there is no change in degree from G to G_S, so $W_{ii} = W''_{ii}$. Hence, we have

$$\begin{aligned}\mathrm{pr}''_{S_i}\left(I-(1-\alpha)W_{ii}\right) &= \alpha s_{S_i} + \mathrm{pr}''_{S_o}\left((1-\alpha)W''_{oi}\right)\\ &\geq \alpha s_{S_i} + \mathrm{pr}''_{S_o}\left((1-\alpha)W_{oi}\right)(1+\frac{1}{d}).\end{aligned}$$

Therefore,

$$\frac{\mathrm{pr}''_{S_o} w_o^T}{\mathrm{pr}''_{S_i} w_i^T} \leq \frac{\mathrm{pr}_{S_i} w_i^T - c_1}{(1+\frac{1}{d})\mathrm{pr}_{S_i} w_i^T},$$

and the lemma follows from

$$\frac{\mathrm{pr}_{S_i} w_i^T + c_2 - c_1}{\mathrm{pr}_{S_i} w_i^T + c_2} \geq \frac{\mathrm{pr}_{S_i} w_i^T - c_1}{\mathrm{pr}_{S_i} w_i^T} \geq \frac{\mathrm{pr}_{S_i} w_i^T - c_1}{(1+\frac{1}{d})\mathrm{pr}_{S_i} w_i^T}.$$

□

We remark that pr'' tends to underestimate the ranking since it ignores all the boundary edges. In the other direction, the influence of boundary nodes should be taken into consideration but not so much as to overestimate the PageRank on S_o in comparison with pr'_{S_o}. Lemma 1 shows that pr_{S_o} is bounded between pr''_{S_o} and pr'_{S_o} as desired.

4 Dirichlet PageRank with Given Boundary Conditions

In this section, we generalize Dirichlet PageRank to use arbitrary boundary conditions given by $\sigma : \delta(S) \to \mathbb{R}$. Note that σ can have negative values.

The Dirichlet PageRank vector with given boundary conditions σ is defined by the equations:

$$\mathrm{pr}(v) = \begin{cases} \alpha s(v) + (1-\alpha)\sum_{u\in V} \mathrm{pr}(u) W_{uv} & \text{if } v \in S,\\ \sigma(v) & \text{if } v \in \delta(S). \end{cases} \tag{6}$$

Here, we use the convention that $|\sigma| \leq 1$. Let $W_{\delta(S)}$ denote W restricted to $\delta(S) \times S$.

Theorem 3. *For a connected graph G, vector s, $\alpha > 0$ and given boundary conditions σ, the above PageRank equation (6) has one and only one solution. With $\beta = \frac{2\alpha}{1-\alpha}$, it is given by*

$$\mathrm{pr}_S = (\beta s_S + 2\sigma_{\delta(S)} W_{\delta(S)}) D_S^{-1/2} \mathcal{G}_{S,\beta} D_S^{1/2}.$$

Proof. Note that

$$\begin{aligned}\mathrm{pr}_S &= \alpha s_S + (1-\alpha)\left(\mathrm{pr}_S W_S + \sigma_{\delta(S)} W_{\delta(S)}\right) \qquad (7)\\ &= \frac{1-\alpha}{2}(\beta s_S + 2\sigma_{\delta(S)} W_{\delta(S)}) + (1-\alpha)\mathrm{pr}_S W_S.\end{aligned}$$

The theorem follows by expressing W_S in terms of $\mathcal{L}_S$ and solving for pr_S, as in the proof of Theorem 3. □

To solve Problem 2, one can adjust the ranking by setting boundary conditions σ and solving the Dirichlet PageRank equation. We can use σ to specify known quantitive distrust which will then propagate to the rest of vertices in S.

5 Algorithms and Analysis

Solving the PageRank equations [(2) or (6)] with boundary conditions requires both matrix-vector multiplication and solving a linear system of the form:

$$x(\beta I_S + \mathcal{L}_S) = y.$$

The running time of solving the PageRank equation is dominated by the complexity of solving the linear system. Since the matrix $\beta I_S + \mathcal{L}_S$ is diagonally dominant, it can be solved approximately in nearly-linear time with a Spielman-Teng Solver [22], but we will also give a simpler algorithm ApproxDirichPR to compute approximate Dirichlet PageRank vectors. This approximation algorithm is faster and has a better approximation ratio if the constant α is not too small.

The algorithm ApproxDirichPR is outlined as follows: we initialize pr_S as $\mathbf{0}$ and maintain a residue r, which is the difference between the right side and left side of equation (7). Then we gradually move the 'mass' from r to pr_S while maintaining the following invariant:

$$\mathrm{pr}_S + r = \alpha s_S + (1-\alpha)\left(\mathrm{pr}_S W_S + \sigma_{\delta(S)} W_{\delta(S)}\right)$$

until we have $r(v) \leq \epsilon' d_v$ for every $v \in S$. At the start, we set $\epsilon' = 1$. After each iteration, we decrease ϵ' by half until $\epsilon' \leq \epsilon$ which is the given desired approximation ratio.

```
Input: $G$, $S$, $\alpha$, $s$, $\sigma$, $\epsilon$
Output: $\mathrm{pr}_S$
  $\mathrm{pr}_S \Leftarrow \mathbf{0}$, $\epsilon' \Leftarrow 1$, $r \Leftarrow \alpha s_S + (1-\alpha)\sigma_{\delta(S)} W_{\delta(S)}$
  while $\epsilon' > \epsilon$ do
    while $|r(v)| \geq \epsilon' d_v$ for some $v$ do
      $\mathrm{pr}_S(v) \Leftarrow \mathrm{pr}_S(v) + r(v)$
      For each neighbor $u$ of $v$, $r(u) \Leftarrow r(u) + (1-\alpha)r(v)/2d_v$
      $r(v) \Leftarrow (1-\alpha)r(v)/2$
    end while
    $\epsilon' \Leftarrow \epsilon'/2$
  end while
```

Algorithm 1: ApproxDirichPR

To analyze the above algorithm, the proof of Theorem 1 is given as follows.

Proof. (of Theorem 1) To bound the running time, we first show that in each iteration of the inner loop, $|r|_1$ will decrease by at least $\alpha\epsilon' d_v$. Let r^b be r before the iteration and r^a be r after the iteration. We have:

$$\begin{aligned}
|r^a|_1 &= |r^a(v)| + \sum_{u \neq v} |r^a(u)| \\
&= \frac{1-\alpha}{2}|r^b(v)| + \sum_{u \neq v} |r^b(u) + \frac{1-\alpha}{2d_v} r^b(v)| \\
&\leq \frac{1-\alpha}{2}|r^b(v)| + \sum_{u \neq v} \left(|r^b(u)| + \frac{1-\alpha}{2d_v}|r^b(v)| \right) \\
&= (1-\alpha)|r^b(v)| + \sum_{u \neq v} |r^b(u)| \\
&= |r^b|_1 - \alpha |r^b(v)|.
\end{aligned}$$

We note that the above equations hold for both positive and negative values of $r^b(v)$. Since v is chosen satisfying $|r(v)| \geq \epsilon' d_v$, it follows that $|r^a|_1 \leq |r^b|_1 - \alpha\epsilon' d_v$.

Note that at the beginning of each iteration of the outer loop, $|r| \leq 2\epsilon' \mathrm{vol}(S)$. Let T be the number of iterations of the inner loop and v_i be the vertex selected at the i-th iteration for $1 \leq i \leq T$. We have:

$$\sum_{i=1}^{T} \alpha\epsilon' d_{v_i} \leq 2\epsilon' \mathrm{vol}(S),$$

which implies

$$\sum_{i=1}^{T} d_{v_i} \leq \frac{2\mathrm{vol}(S)}{\alpha}.$$

Since a FIFO queue can be used to store every vertex v such that $|r(v)| \geq \epsilon' d_v$, each iteration of the inner loop can be completed in $O(d_v)$ time. Therefore, the running time of one outer iteration is $\sum_{i=1}^{T} d_{v_i}$, which is bounded from above by $\frac{2\mathrm{vol}(S)}{\alpha}$.

There are $\log \frac{1}{\epsilon}$ iterations of the outer loop; therefore, the overall running time is $O\left(\frac{\mathrm{vol}(S)\log\frac{1}{\epsilon}}{\alpha}\right)$.

To prove the correctness of the approximation ratio, we will first show that the following invariant is maintained during the entire algorithm:

$$\mathrm{pr}_S + r = \alpha s_S + (1-\alpha)\left(\mathrm{pr}_S W_S + \sigma_{\delta(S)} W_{\delta(S)}\right).$$

This equation holds trivially in the beginning where $\mathrm{pr}_S = \mathbf{0}$ and $r = \alpha s_S + (1-\alpha)\sigma_{\delta(S)} W_{\delta(S)}$. For each inner iteration, let r^b, pr_S^b be r, pr_S before the iteration and r^a, pr^a_S be r, pr_S after the iteration. We have

$$\begin{aligned}
&\mathrm{pr}^a_S(v) + r^a(v)\\
&= \mathrm{pr}^b_S(v) + r^b(v) + \frac{1-\alpha}{2} r^b(v)\\
&= \alpha s_S(v) + (1-\alpha)\left([\mathrm{pr}^b_S W_S](v) + [\sigma_{\delta(S)} W_{\delta(S)}](v)\right) + \frac{1-\alpha}{2} r^b(v)\\
&= \alpha s_S(v) + (1-\alpha)\left(\frac{1}{2}\mathrm{pr}^b_S(v) + \sum_{w \neq v} \frac{1}{2d_w}\mathrm{pr}^b_S(w) + [\sigma_{\delta(S)} W_{\delta(S)}](v)\right) + \frac{1-\alpha}{2} r^b(v)\\
&= \alpha s_S(v) + (1-\alpha)\left(\frac{1}{2}\left(\mathrm{pr}^b_S(v) + r^b(v)\right) + \sum_{w \neq v} \frac{1}{2d_w}\mathrm{pr}^b_S(w) + [\sigma_{\delta(S)} W_{\delta(S)}](v)\right)\\
&= \alpha s_S(v) + (1-\alpha)\left(\frac{1}{2}\mathrm{pr}^a_S(v) + \sum_{w \neq v} \frac{1}{2d_w}\mathrm{pr}^a_S(w) + [\sigma_{\delta(S)} W_{\delta(S)}](v)\right)\\
&= \alpha s_S(v) + (1-\alpha)\left([\mathrm{pr}^a_S W_S](v) + [\sigma_{\delta(S)} W_{\delta(S)}](v)\right),
\end{aligned}$$

and for $u \neq v$,

$$\begin{aligned}
&\mathrm{pr}^a_S(u) + r^a(u)\\
&= \mathrm{pr}^b_S(u) + r^b(u) + \frac{1-\alpha}{2d_v} r^b(v)\\
&= \alpha s_S(u) + (1-\alpha)\left([\mathrm{pr}^b_S W_S](u) + [\sigma_{\delta(S)} W_{\delta(S)}](u)\right) + \frac{1-\alpha}{2d_v} r^b(v)\\
&= \alpha s_S(u) + (1-\alpha)\left(\frac{1}{2}\mathrm{pr}^b_S(u) + \sum_{w \neq u} \frac{1}{2d_w}\mathrm{pr}^b_S(w) + [\sigma_{\delta(S)} W_{\delta(S)}](u)\right) + \frac{1-\alpha}{2d_v} r^b(v)\\
&= \alpha s_S(u) + (1-\alpha)\left(\frac{1}{2}\mathrm{pr}^a_S(u) + \sum_{w \neq u} \frac{1}{2d_w}\mathrm{pr}^a_S(w) - \frac{1}{2d_v} r^b_S(v) + [\sigma_{\delta(S)} W_{\delta(S)}](u)\right)\\
&\quad + \frac{1-\alpha}{2d_v} r^b(v)\\
&= \alpha s_S(u) + (1-\alpha)\left(\frac{1}{2}\mathrm{pr}^a_S(u) + \sum_{w \neq u} \frac{1}{2d_w}\mathrm{pr}^a_S(w) + [\sigma_{\delta(S)} W_{\delta(S)}](u)\right)\\
&= \alpha s_S(u) + (1-\alpha)\left([\mathrm{pr}^a_S W_S](u) + [\sigma_{\delta(S)} W_{\delta(S)}](u)\right).
\end{aligned}$$

Thus, the invariant is maintained during the entire algorithm. Note that the equations hold for both positive and negative values of r. As a result, the output $\widetilde{\mathrm{pr}}_S$ satisfies:

$$\widetilde{\mathrm{pr}}_S + r = \alpha s_S + (1-\alpha)\left(\widetilde{\mathrm{pr}}_S W_S + \sigma_{\delta(S)} W_{\delta(S)}\right),$$

where $|r(v)| < \epsilon d_v$ for all vertices $v \in S$, and the exact solution pr_S satisfies

$$\mathrm{pr}_S = \alpha s_S + (1-\alpha)\left(\mathrm{pr}_S W_S + \sigma_{\delta(S)} W_{\delta(S)}\right).$$

Taking the difference of these two equations, we get

$$\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S = r + (1-\alpha)\left((\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S) W_S\right).$$

Since

$$|(\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S) W_S|_1 \le |\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S|_1,$$

we have

$$|\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S|_1 \le |r|_1 + (1-\alpha)|(\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S) W_S|_1 \le |r|_1 + (1-\alpha)|\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S|_1,$$

which implies

$$|\mathrm{pr}_S - \widetilde{\mathrm{pr}}_S|_1 \le \frac{1}{\alpha}|r|_1 < \frac{\epsilon \mathrm{vol}(S)}{\alpha}.$$

□

6 Applications of Dirichlet PageRank

6.1 Allowing negative edges in the graph

Negative edges arise in many network problems, making PageRank less suitable (see [18]) for finding a desirable ranking. There have been numerous attempts to address this problem; one way is to ignore the entries corresponding to the negative links, as seen as in [17, 19]. By treating a negative link between two vertices the same as no link [17, 19], the PageRank vector can be computed and used as a ranking. Unfortunately, the information contained in those negative links is lost. To remedy this, in [13], a trust ranking is computed based on only positive links and one single step of propagation of distrust. Namely, the distrust is only propagated to immediate neighbors without influencing the rest of the vertices. A more sophisticated algorithm PageTrust is proposed in [18], which uses a fairly complicated update rule relying on a relatively large number of iterations until convergence. However, the running time for each iteration is quite large: $O(\overline{d}nn^-)$, where $\overline{d}$ is the average degree and n^- is the number of vertices receiving negative links. Thus, the worst case complexity is $O(n^3)$.

Using Dirichlet PageRank, we develop a simple and fast algorithm to propagate distrust in graphs with negative edges. The key idea can be outlined as follows: We first compute the usual PageRank based on positive edges, then based on the ranking result, we convert negative links to boundary conditions and then compute Dirichlet PageRank.

Let E^+ be the set of positive edges, E^- be the set of negative edges, and V^- be the set of vertices incident to negative edges. Let $d^-(u)$ and $d^+(u)$ denote the numbers of negative and positive edges incident to u, respectively. For each vertex $u \in V^-$, we create a *shadow vertex* u^s. Let V^s denote the set of all shadow vertices, and define

$$E^s = \big\{\{u^s, v\}, \{u, v^s\} | \{u, v\} \in E^-\big\} .$$

We then form a new graph $\widehat{G} = (\widehat{V}, \widehat{E})$, where $\widehat{V} = V \bigcup V^s$, and $\widehat{E} = E^+ \bigcup E^s$. In the following algorithm, we will set boundary conditions on V^s in $\widehat{G}$ to propagate distrust.

```
Input: $G = (V, E), v, \alpha, \epsilon$
Output: pr
  Determine $\widehat{G}$, $V^s$ and $E^+$ as described above.
  $\mathrm{pr}^+ \Leftarrow$ SharpApproximatePR$(v, \alpha, \epsilon)$ using the graph $(V, E^+)$
  $\sigma(u) \Leftarrow \frac{d^-}{d^+}\mathrm{pr}^+(u)$ for each vertex $u$ in $V^s$
  $\mathrm{pr} \Leftarrow$ ApproxDirichPR$(\widehat{G}, V, \alpha, v, \sigma, \epsilon)$
```

Algorithm 2: NegLinkPageRank

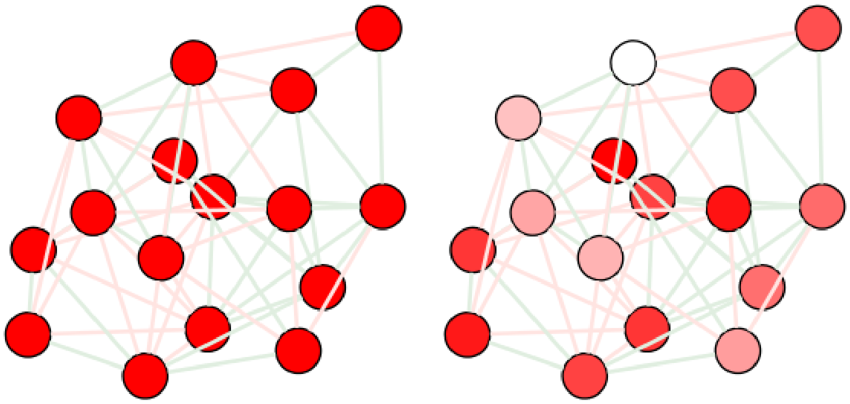
The intuition behind setting the values of the boundary condition $\sigma(u)$ is as follows: We let $\sigma(u)/d^- = \mathrm{pr}^+(u)/d^+$. Namely, for a vertex, the amount of distrust propagated via the negative edge is equal to the amount of trust propagated via the positive edge. The running time of this algorithm is nearly linear time, using our ApproxDirichPR algorithm.

As an example of using Dirichlet PageRank on a graph with positive and negative edges, we examine a network of tribes in New Guinea, studied in the mid-twentieth century [12, 21]. A positive edge indicates a tribal alliance, and negative edges represent enemy relationships. As illustrated in Fig. 5a, the positive edges are in light green and the negative edges are in light red. One way to calculate a vertex ranking is to ignore the negative edges as shown in Fig. 5a. The PageRank vector computed in this manner is actually the uniform distribution. In Fig. 5b, we use the Dirichlet PageRank to compute a vertex ranking taking the negative edges into account. It is apparent that vertices are appropriately ranked by taking advantages of their trusting and distrusting relationships.

6.2 Adjusting Spammers' Influence

One disadvantage of pure link-based ranking systems such as PageRank is that they interpret all nodes as honest agents and all links as votes or validation between nodes. However, real-world networks such as the World Wide Web often contain malicious nodes or spammers. Of interest is to find ranking systems that can better represent the true ranking of nodes in the graph.

There are many schemes developed to tackle such problems [1, 4, 5, 7, 13, 14, 17, 18]. It turns out that many of these schemes can be modeled using the Dirichlet PageRank with different boundary conditions. For example, [4] outlines an algorithm SpamRank which penalizes spam nodes. This algorithm uses sampling to find nodes whose PageRank vectors are significantly different from their neighbors', and gives a heuristic penalty score for each node. It then uses these penalty



(a) PageRank, computed by ignoring negative links.

(b) Dirichlet PageRank, taking negative links into account

Fig. 5: Comparison of vertex rankings on a network [12, 21].

scores as a seed vector for personalized PageRank computation. However, if the penalties is represented by a probability distribution, the sampling techniques can be problematic for vertices with low degree. Using the Dirichlet PageRank, we can penalize known spammers v by enforcing the condition $\mathrm{pr}(v) = 0$. This is done in the example given in Fig. 4. Furthermore, one can adjust the ranking even further by enforcing $\mathrm{pr}(v) = -1$.

In [5], the trust and distrust are propagated within a network through a weighted random walk W with a trusted seed vertex s by assigning at the start, the rank $\mathrm{pr}(s) = 1$. This can be regarded as the Dirichlet PageRank with the boundary condition $\sigma(s) = 1$. There is a subtle difference in the way distrust is handled (the algorithm in [5] does not allow for the propagation of trust scores less than 0). We note that the Dirichlet PageRank allows us to efficiently consider these and many other models.

6.3 Adjusting Rank Based on Trust

While it is important to devise ranking systems that take known spammers into account, it is also crucial to be able to calculate a ranking based on various notions of trust in a network. There are numerous scenarios for which the Dirichlet PageRank with boundary conditions can serve as a useful algorithmic tool.

We consider the following problem: In a network G, the node v wants to compute a personalized ranking of the nodes, but v trusts its own friends and wants its ranking on the top ρ fraction of nodes to be similar to its friends'. This quantifies the assumptions that one's friends' actions carry a great deal of weight in one's own decisions. Vertex v can efficiently compute a personalized PageRank vector as its ranking function using algorithms from [11]. However, PageRank alone will not take into account the implied trust between v and its friends. But using Dirichlet PageRank with boundary conditions, we can take v's trusted friends into account. We illustrate this in the algorithm PRTrustedFriends.

```
Input: $G = (V, E)$, $v$, $\alpha$, $\rho$, $\epsilon$
Output: $\boldsymbol{p}$
  Compute $v$'s ranking: $\boldsymbol{p} \Leftarrow$ SharpApproximatePR($v, \alpha, \epsilon$) [11]
  Compute $v$'s neighbors' rankings: for each neighbor $u$,
  $\boldsymbol{p}_u \Leftarrow$ SharpApproximatePR($u, \alpha, \epsilon$)
  Take a weighted average of $v$'s neighbors' rankings:
  $\boldsymbol{p'} \Leftarrow \frac{1}{\sum_{u \sim v} p(u)} \sum_{u \sim v} p(u) \boldsymbol{p}_u$.
  Take a set $S$ of nodes that $v$ ranks highly:
  $S \Leftarrow \arg\max_{S \subseteq V, |S| \leq \rho |V|} \sum_{s \in S} p(s)$
  Use $v$'s friends' ranking of $S$ to adjust $\boldsymbol{p}$:
  $\boldsymbol{p} \Leftarrow$ ApproxDirichPR($G, V \setminus S, \alpha, v, \boldsymbol{p'}, \epsilon$)
```

Algorithm 3: PRTrustedFriends

A natural extension of PRTrustedFriends can be described as the case that v is a newcomer to a network and is therefore unsure about what other nodes are trustworthy. In such a scenario, the only available information to v is the network itself. For ranking purposes, v can select a small number of nodes to compare with its own ranking. If these nodes are well distributed, this provides some control to ensure that v's own ranking function is not too distorted by the presence of nearby spam or malicious nodes. We give the algorithm PRValidation.

```
Input: $G = (V, E)$, $v$, $k$, $\alpha$, $\rho$, $\epsilon$
Output: $\boldsymbol{p}$
  Compute $v$'s ranking: $\boldsymbol{p} \Leftarrow$ SharpApproximatePR($v, \alpha, \epsilon$) [11]
  $v_1, \ldots, v_k \Leftarrow$ i.i.d. samples from $V$ according to $\boldsymbol{p}$
  Compute rankings for the sampled nodes:
  $\boldsymbol{p}_k \Leftarrow$ SharpApproximatePR($v_k, \alpha, \epsilon$)
  Take a weighted average of these sampled rankings:
  $\boldsymbol{p'} \Leftarrow \frac{1}{\sum_{i=1}^{k} p(v_k)} \sum_{i=1}^{k} p(v_k) \boldsymbol{p}_k$
  Take a set $S$ of nodes that $v$ ranked highly:
  $S \Leftarrow \arg\max_{S \subseteq V, |S| \leq \rho |V|} \sum_{s \in S} p(s)$
  Use the sampled rankings to adjust $\boldsymbol{p}$:
  $\boldsymbol{p} \Leftarrow$ ApproxDirichPR($G, V \setminus S, \alpha, v, \boldsymbol{p'}, \epsilon$)
```

Algorithm 4: PRValidation

A third, more complex situation arises in the context of different types of social networks. Although the setup here appears somewhat complicated, it is a natural model for a common social phenomenon, addressing distinctions among different types of social networks.

Suppose that we have two networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $V_1 \subseteq V_2$. We interpret G_1 as a closely-knit social network where edges represent a deep mutual trust for one another. G_2 is a larger network where edges are formed by relatively weak reasons, such as acquaintance or curiosity. We assume that a vertex v does not know much about the many sources of information acquired through in G_2. An important question for v is to determine which nodes in G_2 are trustworthy? How should the vertices of G_2 be ranked by taking advantage of G_1?

One effective way of finding such a ranking for vertices v in G_2 is to compute the ranking on G_1 first and then compute Dirichlet PageRank on G_2 using G_1's ranking as the boundary condition. This is outlined in the algorithm PRTrustNetwork.

```
Input: $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, $v \in V_1 \cap V_2$, $\alpha$, $\epsilon$
Output: $\boldsymbol{q}$
  $\boldsymbol{p} \Leftarrow$ SharpApproximatePR$(v, \alpha, \epsilon)$ [11] for $G_1$
  $\boldsymbol{q} \Leftarrow$ ApproxDirichPR$(G_2, V_2 \setminus V_1, \alpha, v, \boldsymbol{p}, \epsilon)$
```

Algorithm 5: PRTrustNetwork

The Dirichlet PageRank can also be used to solve a related problem when a global ranking for G_2 is already known or pre-computed. Suppose that such a ranking for G_2 exists, and a vertex $v \in G_1$ wishes to be able to rank G_1 by taking this into account. One way to do this is to compute a Dirichlet PageRank vector for G_1, but using the adjacent nodes in G_1 as a boundary with ranking given by the global ranking on G_2. This procedure is given in the algorithm PRInferRanking as follows.

```
Input: $G_2 = (V_2, E_2)$, $G_1 = (V_1, E_1) \subseteq G_2$, $v \in V_1$, $\boldsymbol{p}$, $\alpha$, $\epsilon$
Output: $\boldsymbol{q}$
  $\partial E_1 \Leftarrow \{(w, x) \in E_2 | w \in V_1, x \notin V_1\}$
  $\partial V_1 \Leftarrow \{w \in V_2 \setminus V_1 | w \text{ is an endpoint of an } e \in \partial E_1\}$
  $\boldsymbol{q} \Leftarrow$ ApproxDirichPR$((V_1 \cup \partial V_1, E_1 \cup \partial E_1), V_1, \alpha, v, \boldsymbol{p}, \epsilon)$
```

Algorithm 6: PRInferRanking

In this section, we have examined several examples as applications of the Dirichlet PageRank vectors. The list is by no means complete. These examples offer a glimpse of the applicability and flexibility of the Dirichlet PageRank. Further applications and research directions remain to be explored.

References

1. R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *WWW* 2008.
2. R. Andersen, F. Chung and K. Lang. Local graph partitioning using PageRank vectors. In *FOCS* 2006.
3. R. Baeza-Yates, C. Castillo and V. López. PageRank increase under different collusion topologies. In Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, 2005.
4. A. Benczur, K. Csalogany, T. Sarlos and M. Uher. SpamRank — fully automatic link spam detection. In Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web, 2005.
5. C. Borgs, J. Chayes, A.T. Kalai, A. Malekian and M. Tennenholtz. A novel approach to propagating distrust. WINE 2010.
6. S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems*, **30 (1-7)**, (1998), 107-117.
7. A. Cheng and E. Friedman, Sybilproof reputation mechanisms. In Proceedings of Third Workshop on Economics of Peer-to-Peer Systems, 2005.
8. F. Chung, PageRank as a discrete Green's function, *Geometry and Analysis* I, ALM 17 (2010), 285–302.
9. F. Chung, *Spectral Graph Theory*, AMS Publications, 1997.
10. F. Chung and S.-T. Yau, Discrete Green's functions, *J. Combinatorial Theory (A)* **91** (2000), 191–214.
11. F. Chung and W. Zhao, A sharp PageRank algorithm with applications to edge ranking and graph sparsification. *Proceedings of Workshop on Algorithms and Models for the Web Graph* (WAW 2010), *Lecture Notes in Computer Science* **6516**, 2–14.
12. P. Hage and F. Harary, *Structural Models in Anthropology*, Cambridge University Press, 1983.
13. R. Guha, R. Kumar, P. Raghavan and A. Tomkins, Propagation of trust and distrust. In *WWW* 2004.
14. Z. Gyöngyi, H. Garcia-Molina and J. Pedersen, Combating Web spam with TrustRank. In *VLDB* 2004.
15. T. Haveliwala, Topic-sensitive PageRank: A context-sensitive ranking algorithm for Web search, *IEEE Transactions on Knowledge and Data Engineering* **15** (2004), 784–796.
16. G. Jeh and J. Widom, Scaling personalized Web search. In *WWW* 2003.
17. S. Kamvar, M. Schlosser and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. WWW 2003.
18. C. de Kerchove and P. Dooren. The PageTrust algorithm: how to rank Web pages when negative links are allowed? In Proceedings of the SIAM International Conference on Data Mining (2008).
19. A. Langville and C. Meyer. Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press, 2006.
20. D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten and S.M. Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology* **54**:4 (2003), 396–405.

21. K. Read, Cultures of the central highlands, New Guinea, *Southwestern Journal of Anthropology* **10** (1954), 1–43.
22. D. A. Spielman and S. -H. Teng, Nearly-Linear Time Algorithms for Preconditioning and Solving Symmetric, Diagonally Dominant Linear Systems, 2008. Available at `http://arxiv.org/abs/cs.NA/0607105`.