DiffusionRank: A Possible Penicillin for Web Spamming

Haixuan Yang, Irwin King, and Michael R. Lyu
Dept. of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
{hxyang,king,lyu}@cse.cuhk.edu.hk

ABSTRACT

While the *PageRank* algorithm has proven to be very effective for ranking Web pages, the rank scores of Web pages can be manipulated. To handle the manipulation problem and to cast a new insight on the Web structure, we propose a ranking algorithm called *DiffusionRank*. *DiffusionRank* is motivated by the heat diffusion phenomena, which can be connected to Web ranking because the activities flow on the Web can be imagined as heat flow, the link from a page to another can be treated as the pipe of an air-conditioner, and heat flow can embody the structure of the underlying Web graph. Theoretically we show that *DiffusionRank* can serve as a generalization of *PageRank* when the heat diffusion coefficient γ tends to infinity. In such a case $1/\gamma = 0$, *DiffusionRank* (*PageRank*) has low ability of anti-manipulation. When $\gamma = 0$, *DiffusionRank* obtains the highest ability of anti-manipulation, but in such a case, the web structure is completely ignored. Consequently, γ is an interesting factor that can control the balance between the ability of preserving the original Web and the ability of reducing the effect of manipulation. It is found empirically that, when $\gamma = 1$, *DiffusionRank* has a Penicillin-like effect on the link manipulation. Moreover, *DiffusionRank* can be employed to find group-to-group relations on the Web, to divide the Web graph into several parts, and to find link communities. Experimental results show that the *DiffusionRank* algorithm achieves the above mentioned advantages as expected.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Search and Retrieval; G2.2 [**Discrete Mathematics**]: Graph Theory

General Terms: Algorithms.

Keywords: Random Graph, PageRank, DiffusionRank

1. INTRODUCTION

While the *PageRank* algorithm [13] has proven to be very effective for ranking Web pages, inaccurate *PageRank* results are induced because of web page manipulations by people for commercial interests. The manipulation problem is also called the Web spam, which refers to hyperlinked pages on the World Wide Web that are created with the intention of misleading search engines [7]. It is reported that approximately 70% of all pages in the .biz domain and about 35% of the pages in the .us domain belong to the spam category [12]. The reason for the increasing amount of Web spam is explained in [12]: some web site operators try to influence the positioning of their pages within search results because of the large fraction of web traffic originating from searches and the high potential monetary value of this traffic.

From the viewpoint of the Web site operators who want to increase the ranking value of a particular page for search engines, Keyword Stuffing and Link Stuffing are being used widely [7, 12]. From the viewpoint of the search engine managers, the Web spam is very harmful to the users' evaluations and thus their preference to choosing search engines because people believe that a good search engine should not return irrelevant or low-quality results. There are two methods being employed to combat the Web spam problem. Machine learning methods are employed to handle the keyword stuffing. To successfully apply machine learning methods, we need to dig out some useful textual features for Web pages, to mark part of the Web pages as either spam or non-spam, then to apply supervised learning techniques to mark other pages. For example, see [5, 12]. Link analysis methods are also employed to handle the link stuffing problem. One example is the *TrustRank* [7], a link-based method, in which the link structure is utilized so that human labelled trusted pages can propagate their trust scores trough their links. This paper focuses on the link-based method.

The rest of the materials are organized as follows. In the next section, we give a brief literature review on various related ranking techniques. We establish the Heat Diffusion Model (HDM) on various cases in Section 3, and propose *DiffusionRank* in Section 4. In Section 5, we describe the data sets that we worked on and the experimental results. Finally, we draw conclusions in Section 6.

2. LITERATURE REVIEW

The importance of a Web page is determined by either the textual content of pages or the hyperlink structure or both. As in previous work [7, 13], we focus on ranking methods solely determined by hyperlink structure of the Web graph. All the mentioned ranking algorithms are established on a graph. For our convenience, we first give some notations. Denote a static graph by $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, $E = \{(v_i, v_j) \mid \text{there is an edge from } v_i \text{ to}$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...$5.00.

$v_j\}$. I_i and d_i denote the in-degree and the out-degree of page i respectively.

2.1 PageRank

The importance of a Web page is an inherently subjective matter, which depends on the reader's interests, knowledge and attitudes [13]. However, the average importance of all readers can be considered as an objective matter. *PageRank* tries to find such average importance based on the Web link structure, which is considered to contain a large amount of statistical data. The Web is modelled by a directed graph G in the *PageRank* algorithms, and the rank or "importance" x_i for page $v_i \in V$ is defined recursively in terms of pages which point to it: $x_i = \sum_{(j,i)\in E} a_{ij}x_j$, where a_{ij} is assumed to be $1/d_j$ if there is a link from j to i, and 0 otherwise. Or in matrix terms, $\mathbf{x} = \mathbf{A}\mathbf{x}$. When the concept of "random jump" is introduced, the matrix form is changed to

$$\mathbf{x} = [(1-\alpha)\mathbf{g}\mathbf{1}^T + \alpha\mathbf{A}]\mathbf{x}, \tag{1}$$

where α is the probability of following the actual link from a page, $(1-\alpha)$ is the probability of taking a "random jump", and $\mathbf{g}$ is a stochastic vector, i.e., $\mathbf{1}^T\mathbf{g} = \mathbf{1}$. Typically, $\alpha = 0.85$, and $\mathbf{g} = \frac{1}{n}\mathbf{1}$ is one of the standard settings, where $\mathbf{1}$ is the vector of all ones [6, 13].

2.2 TrustRank

TrustRank [7] is composed of two parts. The first part is the seed selection algorithm, in which the inverse *PageRank* was proposed to help an expert of determining a good node. The second part is to utilize the biased *PageRank*, in which the stochastic distribution $\mathbf{g}$ is set to be shared by all the trusted pages found in the first part. Moreover, the initial input of $\mathbf{x}$ is also set to be $\mathbf{g}$. The justification for the inverse *PageRank* and the solid experiments support its advantage in combating the Web spam. Although there are many variations of *PageRank*, e.g., a family of link-based ranking algorithms in [2], *TrustRank* is especially chosen for comparisons for three reasonss: (1) it is designed for combatting spamming; (2) its fixed parameters make a comparison easy; and (3) it has a strong theoretical relations with *PageRank* and *DiffusionRank*.

2.3 Manifold Ranking

In [17], the idea of ranking on the data manifolds was proposed. The data points represented as vectors in Euclidean space are considered to be drawn from a manifold. From the data points on such a manifold, an undirected weighted graph is created, then the weight matrix is given by the Gaussian Kernel smoothing. While the manifold ranking algorithm achieves an impressive result on ranking images, the biased vector $\mathbf{g}$ and the parameter k in the general personalized *PageRank* in [17] are unknown in the Web graph setting; therefore we do not include it in the comparisons.

2.4 Heat Diffusion

Heat diffusion is a physical phenomena. In a medium, heat always flow from position with high temperature to position with low temperature. Heat kernel is used to describe the amount of heat that one point receives from another point. Recently, the idea of heat kernel on a manifold is borrowed in applications such as dimension reduction [3] and classification [9, 10, 14]. In these work, the input data is considered to lie in a special structure.

All the above topics are related to our work. The readers can find that our model is a generalization of *PageRank* in order to resist Web manipulation, that we inherit the first part of *TrustRank*, that we borrow the concept of ranking on the manifold to introduce our model, and that heat diffusion is a main scheme in this paper.

3. HEAT DIFFUSION MODEL

Heat diffusion provides us with another perspective about how we can view the Web and also a way to calculate ranking values. In this paper, the Web pages are considered to be drawn from an unknown manifold, and the link structure forms a directed graph, which is considered as an approximation to the unknown manifold. The heat kernel established on the Web graph is considered as the representation of the relationship between Web pages. The temperature distribution after a fixed time period, induced by a special initial temperature distribution, is considered as the rank scores on the Web pages. Before establishing the proposed models, we first show our motivations.

3.1 Motivations

There are two points to explain that *PageRank* is susceptible to web spam.

- **Over-democratic.** There is a belief behind *PageRank*—all pages are born equal. This can be seen from the equal voting ability of one page: the sum of each column is equal to one. This equal voting ability of all pages gives the chance for a Web site operator to increase a manipulated page by creating a large number of new pages pointing to this page since all the newly created pages can obtain an equal voting right.
- **Input-independent.** For any given non-zero initial input, the iteration will converge to the same stable distribution corresponding to the maximum eigenvalue 1 of the transition matrix. This input-independent property makes it impossible to set a special initial input (larger values for trusted pages and less values even negative values for spam pages) to avoid web spam.

The input-independent feature of *PageRank* can be further explained as follows. $\mathbf{P} = [(1-\alpha)\mathbf{g}\mathbf{1}^T + \alpha\mathbf{A}]$ is a positive stochastic matrix if $\mathbf{g}$ is set to be a positive stochastic vector (the uniform distribution is one of such settings), and so the largest eigenvalue is 1 and no other eigenvalue whose absolute value is equal to 1, which is guaranteed by the Perron Theorem [11]. Let $\mathbf{y}$ be the eigenvector corresponding to 1, then we have $\mathbf{P}\mathbf{y} = \mathbf{y}$. Let $\{\mathbf{x}_k\}$ be the sequence generated from the iterations $\mathbf{x}_{k+1} = \mathbf{P}\mathbf{x}_k$, and $\mathbf{x}_0$ is the initial input. If $\{\mathbf{x}_k\}$ converges to $\mathbf{x}$, then $\mathbf{x}_{k+1} = \mathbf{P}\mathbf{x}_k$ implies that $\mathbf{x}$ must satisfy $\mathbf{P}\mathbf{x} = \mathbf{x}$. Since the only maximum eigenvalue is 1, we have $\mathbf{x} = c\mathbf{y}$ where c is a constant, and if both $\mathbf{x}$ and $\mathbf{y}$ are normalized by their sums, then $c = 1$. The above discussions show that *PageRank* is independent of the initial input $\mathbf{x}_0$.

In our opinion, $\mathbf{g}$ and α are objective parameters determined by the users' behaviors and preferences. $\mathbf{A}$, α and $\mathbf{g}$ are the "true" web structure. While $\mathbf{A}$ is obtained by a crawler and the setting $\alpha = 0.85$ is accepted by the people, we think that $\mathbf{g}$ should be determined by a user behavior investigation, something like [1]. Without any prior knowledge, $\mathbf{g}$ has to be set as $\mathbf{g} = \frac{1}{n}\mathbf{1}$.

TrustRank model does not follow the "true" web structure by setting a biased $\mathbf{g}$, but the effects of combatting spamming are achieved in [7]; *PageRank* is on the contrary in some ways. We expect a ranking algorithm that has an effect of anti-manipulation as *TrustRank* while respecting the "true" web structure as *PageRank*.

We observe that the heat diffusion model is a natural way to avoid the over-democratic and input-independent feature of *PageRank*. Since heat always flows from a position with higher temperatures to one with lower temperatures, points are not equal as some points are born with high temperatures while others are born with low temperatures. On the other hand, different initial temperature distributions will give rise to different temperature distributions after a fixed time period. Based on these considerations, we propose the novel *DiffusionRank*. This ranking algorithm is also motivated by the viewpoint for the Web structure. We view all the Web pages as points drawn from a highly complex geometric structure, like a manifold in a high dimensional space. On a manifold, heat can flow from one point to another through the underlying geometric structure in a given time period. Different geometric structures determine different heat diffusion behaviors, and conversely the diffusion behavior can reflect the geometric structure. More specifically, on the manifold, the heat flows from one point to another point, and in a given time period, if one point x receives a large amount of heat from another point y, we can say x and y are well connected, and thus x and y have a high similarity in the sense of a high mutual connection.

We note that on a point with unit mass, the temperature and the heat of this point are equivalent, and these two terms are interchangeable in this paper. In the following, we first show the HDM on a manifold, which is the origin of HDM, but cannot be employed to the World Wide Web directly, and so is considered as the ideal case. To connect the ideal case and the practical case, we then establish HDM on a graph as an intermediate case. To model the real world problem, we further build HDM on a random graph as a practical case. Finally we demonstrate the *DiffusionRank* which is derived from the HDM on a random graph.

3.2 Heat Flow On a Known Manifold

If the underlying manifold is known, the heat flow throughout a geometric manifold with initial conditions can be described by the following second order differential equation: $\frac{\partial \mathbf{f}(\mathbf{x},t)}{\partial t} - \Delta \mathbf{f}(\mathbf{x},t) = 0$, where $\mathbf{f}(\mathbf{x},t)$ is the heat at location $\mathbf{x}$ at time t, and $\Delta \mathbf{f}$ is the *Laplace-Beltrami operator* on a function $\mathbf{f}$. The heat diffusion kernel $\mathbf{K}_t(\mathbf{x},\mathbf{y})$ is a special solution to the heat equation with a special initial condition—a unit heat source at position $\mathbf{y}$ when there is no heat in other positions. Based on this, the heat kernel $\mathbf{K}_t(\mathbf{x},\mathbf{y})$ describes the heat distribution at time t diffusing from the initial unit heat source at position $\mathbf{y}$, and thus describes the connectivity (which is considered as a kind of similarity) between $\mathbf{x}$ and $\mathbf{y}$. However, it is very difficult to represent the World Wide Web as a regular geometry with a known dimension; even the underlying is known, it is very difficult to find the heat kernel $\mathbf{K}_t(\mathbf{x},\mathbf{y})$, which involves solving the heat equation with the delta function as the initial condition. This motivates us to investigate the heat flow on a graph. The graph is considered as an approximation to the underlying manifold, and so the heat flow on the graph is considered as an approximation to the heat flow on the manifold.

3.3 On an Undirected Graph

On an undirected graph G, the edge (v_i, v_j) is considered as a pipe that connects nodes v_i and v_j. The value $f_i(t)$ describes the heat at node v_i at time t, beginning from an initial distribution of heat given by $f_i(0)$ at time zero. $\mathbf{f}(t)$ ($\mathbf{f}(0)$) denotes the vector consisting of $f_i(t)$ ($f_i(0)$).

We construct our model as follows. Suppose, at time t, each node i receives $M(i,j,t,\Delta t)$ amount of heat from its neighbor j during a period of Δt. The heat $M(i,j,t,\Delta t)$ should be proportional to the time period Δt and the heat difference $f_j(t) - f_i(t)$. Moreover, the heat flows from node j to node i through the pipe that connects nodes i and j. Based on this consideration, we assume that $M(i,j,t,\Delta t) = \gamma(f_j(t) - f_i(t))\Delta t$. As a result, the heat difference at node i between time $t+\Delta t$ and time t will be equal to the sum of the heat that it receives from all its neighbors. This is formulated as

$$f_i(t+\Delta t) - f_i(t) = \sum_{j:(j,i)\in E} \gamma(f_j(t) - f_i(t))\Delta t, \tag{2}$$

where E is the set of edges. To find a closed form solution to Eq. (2), we express it in a matrix form: $(\mathbf{f}(t+\Delta t) - \mathbf{f}(t))/\Delta t = \gamma \mathbf{H}\mathbf{f}(t)$, where $d(v)$ denotes the degree of the node v. In the limit $\Delta t \to 0$, it becomes $\frac{d}{dt}\mathbf{f}(t) = \gamma \mathbf{H}\mathbf{f}(t)$. Solving it, we obtain $\mathbf{f}(t) = e^{\gamma t \mathbf{H}}\mathbf{f}(0)$, especially we have

$$\mathbf{f}(1) = e^{\gamma \mathbf{H}}\mathbf{f}(0), H_{ij} = \begin{cases} -d(v_j), & j = i, \\ 1, & (v_j, v_i) \in E, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $e^{\gamma H}$ is defined as $e^{\gamma \mathbf{H}} = \mathbf{I} + \gamma \mathbf{H} + \frac{\gamma^2}{2!}\mathbf{H}^2 + \frac{\gamma^3}{3!}\mathbf{H}^3 + \cdots$.

3.4 On a Directed Graph

The above heat diffusion model must be modified to fit the situation where the links between Web pages are directed. On one Web page, when the page-maker creates a link (a,b) to another page b, he actually forces the energy flow, for example, people's click-through activities, to that page, and so there is added energy imposed on the link. As a result, heat flows in a one-way manner, only from a to b, not from b to a. Based on such consideration, we modified the heat diffusion model on an undirected graph as follows.

On a directed graph G, the pipe (v_i, v_j) is forced by added energy such that heat flows only from v_i to v_j. Suppose, at time t, each node v_i receives $RH = RH(i,j,t,\Delta t)$ amount of heat from v_j during a period of Δt. We have three assumptions: (1) RH should be proportional to the time period Δt; (2) RH should be proportional to the the heat at node v_j; and (3) RH is zero if there is no link from v_j to v_i. As a result, v_i will receive $\sum_{j:(v_j,v_i)\in E} \sigma_j f_j(t)\Delta t$ amount of heat from all its neighbors that points to it.

On the other hand, node v_i diffuses $DH(i,t,\Delta t)$ amount of heat to its subsequent nodes. We assume that: (1) The heat $DH(i,t,\Delta t)$ should be proportional to the time period Δt. (2) The heat $DH(i,t,\Delta t)$ should be proportional to the the heat at node v_i. (3) Each node has the same ability of diffusing heat. This fits the intuition that a Web surfer only has one choice to find the next page that he wants to browse. (4) The heat $DH(i,t,\Delta t)$ should be uniformly distributed to its subsequent nodes. The real situation is more complex than what we assume, but we have to make this simple assumption in order to make our model concise. As a result, node v_i will diffuse $\gamma f_i(t)\Delta t/d_i$ amount of heat to any of its subsequent nodes, and each of its subsequent node should receive $\gamma f_i(t)\Delta t/d_i$ amount of heat. Therefore $\sigma_j = \gamma/d_j$. To sum up, the heat difference at node v_i between time $t+\Delta t$ and time t will be equal to the sum of the heat that it receives, deducted by what it diffuses. This is formulated as $f_i(t+\Delta t) - f_i(t) = -\gamma f_i(t)\Delta t + \sum_{j:(v_j,v_i)\in E} \gamma/d_j f_j(t)\Delta t$. Similarly, we obtain

$$\mathbf{f}(1) = e^{\gamma \mathbf{H}}\mathbf{f}(0), H_{ij} = \begin{cases} -1, & j = i, \\ 1/d_j, & (v_j, v_i) \in E, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

3.5 On a Random Directed Graph

For real world applications, we have to consider random edges. This can be seen in two viewpoints. The first one is that in Eq. (1), the Web graph is actually modelled as a random graph, there is an edge from node v_i to node v_j with a probability of $(1-\alpha)g_j$ (see the item $(1-\alpha)\mathbf{g}\mathbf{1}^T$), and that the Web graph is predicted by a random graph [15, 16]. The second one is that the Web structure is a random graph in essence if we consider the content similarity between two pages, though this is not done in this paper. For these reasons, the model would become more flexible if we extend it to random graphs. The definition of a random graph is given below.

Definition 1. *A random graph $RG = (V, \mathbf{P} = (p_{ij}))$ is defined as a graph with a vertex set V in which the edges are chosen independently, and for $1 \le i, j \le |V|$ the probability of (v_i, v_j) being an edge is exactly p_{ij}.*

The original definition of random graphs in [4], is changed slightly to consider the situation of directed graphs. Note that every static graph can be considered as a special random graph in the sense that p_{ij} can only be 0 or 1.

On a random graph $RG = (V, \mathbf{P})$, where $\mathbf{P} = (p_{ij})$ is the probability of the edge (v_i, v_j) exists. In such a random graph, the expected heat difference at node i between time $t + \Delta t$ and time t will be equal to the sum of the expected heat that it receives from all its antecedents, deducted by the expected heat that it diffuses.

Since the probability of the link (v_j, v_i) is p_{ji}, the expected heat flow from node j to node i should be multiplied by p_{ji}, and so we have $f_i(t+\Delta t) - f_i(t) = -\gamma\ f_i(t)\,\Delta t + \sum_{j:(v_j,v_i)\in E} \gamma p_{ji} f_j(t)\Delta t/RD^+(v_j)$, where $RD^+(v_i)$ is the expected out-degree of node v_i, it is defined as $\sum_k p_{ik}$. Similarly we have

$$\mathbf{f}(1) = e^{\gamma \mathbf{R}}\mathbf{f}(0), R_{ij} = \begin{cases} -1, & j = i; \\ p_{ji}/RD^+(v_j), & j \ne i. \end{cases} \quad (5)$$

When the graph is large, a direct computation of $e^{\gamma \mathbf{R}}$ is time-consuming, and we adopt its discrete approximation:

$$\mathbf{f}(1) = (\mathbf{I} + \frac{\gamma}{N}\mathbf{R})^N \mathbf{f}(0). \quad (6)$$

The matrix $(\mathbf{I} + \frac{\gamma}{N}\mathbf{R})^N$ in Eq. (6) and matrix $e^{\gamma \mathbf{R}}$ in Eq. (5) are called *Discrete Diffusion Kernel* and the *Continuous Diffusion Kernel* respectively. Based on the Heat Diffusion Models and their solutions, *DiffusionRank* can be established on undirected graphs, directed graphs, and random graphs. In the next section, we mainly focus on *DiffusionRank* in the random graph setting.

4. DIFFUSIONRANK

For a random graph, the matrix $(\mathbf{I} + \frac{\gamma}{N}\mathbf{R})^N$ or $e^{\gamma \mathbf{R}}$ can measure the similarity relationship between nodes. Let $f_i(0)$= $1, f_j(0) = 0$ if $j \ne i$, then the vector $\mathbf{f}(0)$ represent the unit heat at node v_i while all other nodes has zero heat. For such $\mathbf{f}(0)$ in a random graph, we can find the heat distribution at time 1 by using Eq. (5) or Eq. (6). The heat distribution is exactly the $i-$th row of the matrix of $(\mathbf{I} + \frac{\gamma}{N}\mathbf{R})^N$ or $e^{\gamma \mathbf{R}}$. So the ith-row jth-column element h_{ij} in the matrix $(\mathbf{I} + \gamma\Delta t\mathbf{R})^N$ or $e^{\gamma \mathbf{R}}$ means the amount of heat that v_i can receive from v_j from time 0 to 1. Thus the value h_{ij} can be used to measure the similarity from v_j to v_i. For a static graph, similarly the matrix $(I + \frac{\gamma}{N}\mathbf{H})^N$ or $e^{\gamma \mathbf{H}}$ can measure the similarity relationship between nodes.

The intuition behind is that the amount $h(i, j)$ of heat that a page v_i receives from a unit heat in a page v_j in a unit time embodies the extent of the link connections from page v_j to page v_i. Roughly speaking, when there are more uncrossed paths from v_j to v_i, v_i will receive more heat from v_j; when the path length from v_j to v_i is shorter, v_i will receive more heat from v_j; and when the pipe connecting v_j and v_i is wide, the heat will flow quickly. The final heat that v_i receives will depend on various paths from v_j to v_i, their length, and the width of the pipes.

Algorithm 1 DiffusionRank Function

Input: The transition matrix $\mathbf{A}$; the inverse transition matrix $\mathbf{U}$; the decay factor α_I for the inverse PageRank; the decay factor α_B for PageRank; number of iterations M_I for the inverse PageRank; the number of trusted pages L; the thermal conductivity coefficient γ.

Output: DiffusionRank score vector $\mathbf{h}$.

```
$\mathbf{s} = \mathbf{1}$
for $i = 1$ TO $M_I$ do
    $\mathbf{s} = \alpha_I \cdot \mathbf{U} \cdot \mathbf{s} + (1 - \alpha_I) \cdot \frac{1}{n} \cdot \mathbf{1}$
end for
Sort $\mathbf{s}$ in a decreasing order: $\pi = Rank(\{1, \dots, n\}, \mathbf{s})$
$\mathbf{d} = \mathbf{0}$, $Count = 0$, $i = 0$
while $Count \le L$ do
    if $\pi(i)$ is evaluated as a trusted page then
        $\mathbf{d}(\pi(i)) = 1, Count + +$
    end if
    $i + +$
end while
$\mathbf{d} = \mathbf{d}/|\mathbf{d}|$
$\mathbf{h} = \mathbf{d}$
Find the iteration number $M_B$ according to $\lambda$
for $i = 1$ TO $M_B$ do
    $\mathbf{h} = (1 - \frac{\gamma}{M_B})\mathbf{h} + \frac{\gamma}{M_B}(\alpha_B \cdot \mathbf{A} \cdot \mathbf{h} + (1 - \alpha_B) \cdot \frac{1}{n} \cdot \mathbf{1})$
end for
RETURN $\mathbf{h}$
```

4.1 Algorithm

For the ranking task, we adopt the heat kernel on a random graph. Formally the *DiffusionRank* is described in Algorithm 1, in which,the element U_{ij} in the inverse transition matrix $\mathbf{U}$ is defined to be $1/I_j$ if there is a link from i to j, and 0 otherwise. This trusted pages selection procedure by inverse *PageRank* is completely borrowed from *TrustRank* [7] except for a fix number of the size of the trusted set. Although the inverse *PageRank* is not perfect in its ability of determining the maximum coverage, it is appealing because of its polynomial execution time and its reasonable intuition—we actually inverse the original link when we try to build the seed set from those pages that point to many pages that in turn point to many pages and so on. In the algorithm, the underlying random graph is set as $\mathbf{P} = \alpha_B \cdot \mathbf{A} + (1 - \alpha_B) \cdot \frac{1}{n} \cdot \mathbf{1}_{n\times n}$, which is induced by the Web graph. As a result, $\mathbf{R} = -\mathbf{I} + \mathbf{P}$.

In fact, the more general setting for *DiffusionRank* is $\mathbf{P} = \alpha_B \cdot \mathbf{A} + (1-\alpha_B) \cdot \frac{1}{n} \cdot \mathbf{g} \cdot \mathbf{1}^T$. By such a setting, *DiffusionRank* is a generalization of *TrustRank* when γ tends to infinity and when g is set in the same way as *TrustRank*. However, the second part of *TrustRank* is not adopted by us. In our model, $\mathbf{g}$ should be the true "teleportation" determined by the user's browse habits, popularity distribution over all the Web pages, and so on; $\mathbf{P}$ should be the true model of the random nature of the World Wide Web. Setting $\mathbf{g}$ according to the trusted pages will not be consistent with the basic idea of Heat Diffusion on a random graph. We simply set $\mathbf{g} = \mathbf{1}$ only because we cannot find it without any priori knowledge.

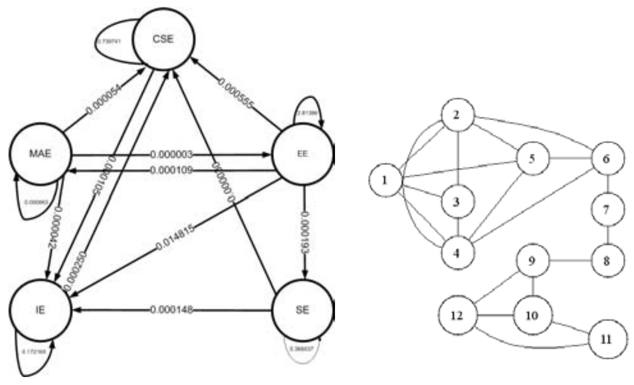
Remark. In a social network interpretation, *DiffusionRank* first recognizes a group of trusted people, who may not be highly ranked, but they know many other people. The initially trusted people are endowed with the power to decide who can be further trusted, but cannot decide the final voting results, and so they are not dictators.

4.2 Advantages

Next we show the four advantages for *DiffusionRank*.

4.2.1 Two closed forms

First, its solutions have two forms, both of which are closed form. One takes the discrete form, and has the advantage of fast computing while the other takes the continuous form, and has the advantage of being easily analyzed in theoretical aspects. The theoretical advantage has been shown in the proof of theorem in the next section.



(a) Group to Group Relations (b) An undirected graph

Figure 1: Two graphs

4.2.2 Group-group relations

Second, it can be naturally employed to detect the group-group relation. For example, let $G2$ and $G1$ denote two groups, containing pages $(j_1, j_2, \ldots, j_s)$ and $(i_1, i_2, \ldots, i_t)$, respectively. Then $\sum_{u,v} h_{i_u,j_v}$ is the total amounts of heat that $G1$ receives from $G2$, where h_{i_u,j_v} is the i_u-th row j_v-th column element of the heat kernel. More specifically, we need to first set $f(0)$ for such an application as follows. In $\mathbf{f}(0) = (f_1(0), f_2(0), \ldots, f_n(0))^T$, if $i \in \{j_1, j_2, \ldots, j_s\}$, then $f_i(0) = 1$, and 0 otherwise. Then we employ Eq. (5) to calculate $\mathbf{f}(1) = (f_1(1), f_2(1), \ldots, f_n(1))^T$, finally we sum those $f_j(1)$ where $j \in \{i_1, i_2, \ldots, i_t\}$. Fig. 1 (a) shows the results generated by the *DiffusionRank*. We consider five groups—five departments in our Engineering Faculty: CSE, MAE, EE, IE, and SE. γ is set to be 1, the numbers in Fig. 1 (a) are the amount of heat that they diffuse to each other. These results are normalized by the total number of each group, and the edges are ignored if the values are less than 0.000001. The group-to-group relations are therefore detected, for example, we can see that the most strong overall tie is from EE to IE. While it is a natural application for *DiffusionRank* because of the easy interpretation by the amount heat from one group to another group, it is difficult to apply other ranking techniques to such an application because they lack such a physical meaning.

4.2.3 Graph cut

Third, it can be used to partition the Web graph into several parts. A quick example is shown below. The graph in Fig. 1 (b) is an undirected graph, and so we employ the Eq. (3). If we know that node 1 belongs to one community and that node 12 belongs to another community, then we can put one unit positive heat source on node 1 and one unit negative heat source on node 12. After time 1, if we set $\gamma = 0.5$, the heat distribution is [0.25, 0.16, 0.17, 0.16, 0.15, 0.09, 0.01, -0.04, -0.18 -0.21, -0.21, -0.34], and if we set $\gamma = 1$, it will be [0.17, 0.16, 0.17, 0.16, 0.16, 0.12, 0.02, -0.07, -0.18, -0.22, -0.24, -0.24]. In both settings, we can easily divide the graph into two parts: $\{1, 2, 3, 4, 5, 6, 7\}$ with positive temperatures and $\{8, 9, 10, 11, 12\}$ with negative temperatures. For directed graphs and random graphs, similarly we can cut them by employing corresponding heat solution.

4.2.4 Anti-manipulation

Fourth, it can be used to combat manipulation. Let $G2$ contain trusted Web pages $(j_1, j_2, \ldots, j_s)$, then for each page i, $\sum_v h_{i,j_v}$ is the heat that page i receives from $G2$, and can be computed by the discrete approximation of Eq. (4) in the case of a static graph or Eq. (6) in the case of a random graph, in which $f(0)$ is set to be a special initial heat distribution so that the trusted Web pages have unit heat while all the others have zero heat. In doing so, manipulated Web page will get a lower rank unless it has strong in-links from the trusted Web pages directly or indirectly. The situation is quite different for *PageRank* because *PageRank* is input-independent as we have shown in Section 3.1. Based on the fact that the connection from a trusted page to a "bad" page should be weak–less uncross paths, longer distance and narrower pipe, we can say *DiffusionRank* can resist web spam if we can select trusted pages. It is fortunate that the trusted pages selection method in [7]–the first part of *TrustRank* can help us to fulfill this task. For such an application of *DiffusionRank*, the computation complexity for *Discrete Diffusion Kernel* is the same as that for *PageRank* in cases of both a static graph and a random graph. This can be seen in Eq. (6), by which we need N iterations and for each iteration we need a multiplication operation between a matrix and a vector, while in Eq. (1) we also need a multiplication operation between a matrix and a vector for each iteration.

4.3 The Physical Meaning of γ

γ plays an important role in the anti-manipulation effect of *DiffusionRank*. γ is the thermal conductivity–the heat diffusion coefficient. If it has a high value, heat will diffuse very quickly. Conversely, if it is small, heat will diffuse slowly. In the extreme case, if it is infinitely large, then heat will diffuse from one node to other nodes immediately, and this is exactly the case corresponding to *PageRank*. Next, we will interpret it mathematically.

THEOREM 1. *When γ tends to infinity and $\mathbf{f}(0)$ is not the zero vector, $e^{\gamma\mathbf{R}}\mathbf{f}(0)$ is proportional to the stable distribution produced by PageRank.*

Let $\mathbf{g} = \frac{1}{n}\mathbf{1}$. By the Perron Theorem [11], we have shown that 1 is the largest eigenvalue of $\mathbf{P} = [(1-\alpha)\mathbf{g}\mathbf{1}^T + \alpha\mathbf{A}]$, and that no other eigenvalue whose absolute value is equal to 1. Let $\mathbf{x}$ be the stable distribution, and so $\mathbf{Px} = \mathbf{x}$. $\mathbf{x}$ is the eigenvector corresponding to the eigenvalue 1. Assume the $n-1$ other eigenvalues of $\mathbf{P}$ are $|\lambda_2| < 1, \ldots, |\lambda_n| < 1$, we can find an invertible matrix $\mathbf{S} = (\ \mathbf{x} \quad \mathbf{S}_1\ )$ such that

$$\mathbf{S}^{-1}\mathbf{PS} = \begin{pmatrix} 1 & * & * & * \\ 0 & \lambda_2 & * & * \\ & & \ddots & \\ 0 & 0 & & * \\ 0 & 0 & 0 & \lambda_n \end{pmatrix}. \tag{7}$$

Since $e^{\gamma\mathbf{R}} = e^{\gamma(-\mathbf{I}+\mathbf{P})} =$

$$S^{-1}\begin{pmatrix} 1 & * & * & * \\ 0 & e^{\gamma(\lambda_2-1)} & * & * \\ & & \ddots & \\ 0 & 0 & & * \\ 0 & 0 & 0 & e^{\gamma(\lambda_n-1)} \end{pmatrix}S, \tag{8}$$

all eigenvalues of the matrix $e^{\gamma\mathbf{R}}$ are $1, e^{\gamma(\lambda_2-1)}, \ldots, e^{\gamma(\lambda_n-1)}$. When $\gamma \to \infty$, they become $1, 0, \ldots, 0$, which means that 1 is the only nonzero eigenvalue of $e^{\gamma\mathbf{R}}$ when $\gamma \to \infty$. We can see that when $\gamma \to \infty$, $e^{\gamma\mathbf{R}}e^{\gamma\mathbf{R}}\mathbf{f}(0) = e^{\gamma\mathbf{R}}\mathbf{f}(0)$, and so $e^{\gamma\mathbf{R}}\mathbf{f}(0)$ is an eigenvector of $e^{\gamma\mathbf{R}}$ when $\gamma \to \infty$. On the other hand, $e^{\gamma\mathbf{R}}x = (\mathbf{I}+\gamma\mathbf{R}+\frac{\gamma^2}{2!}\mathbf{R}^2+\frac{\gamma^3}{3!}\mathbf{R}^3+\ldots)\mathbf{x} = \mathbf{Ix}+\gamma\mathbf{Rx}+\frac{\gamma^2}{2!}\mathbf{R}^2\mathbf{x}+\frac{\gamma^3}{3!}\mathbf{R}^3\mathbf{x}+\ldots = \mathbf{x}$ since $\mathbf{Rx} = (-\mathbf{I}+\mathbf{P})\mathbf{x} = -\mathbf{x}+\mathbf{x} = \mathbf{0}$, and hence $\mathbf{x}$ is the eigenvector of $e^{\gamma\mathbf{R}}$ for any γ. Therefore both $\mathbf{x}$ and $e^{\gamma\mathbf{R}}\mathbf{f}(0)$ are the eigenvectors corresponding the unique eigenvalue 1 of $e^{\gamma\mathbf{R}}$ when $\gamma \to \infty$, and consequently $\mathbf{x} = ce^{\gamma\mathbf{R}}\mathbf{f}(0)$.

By this theorem, we see that *DiffusionRank* is a generalization of *PageRank*. When $\gamma = 0$, the ranking value is most robust to manipulation since no heat is diffused and the system is unchangeable, but the Web structure is completely ignored since $e^{\gamma\mathbf{R}}\mathbf{f}(0) = e^{0\mathbf{R}}\mathbf{f}(0) = \mathbf{If}(0) = \mathbf{f}(0)$; when $\gamma = \infty$, *DiffusionRank* becomes *PageRank*, it can be manipulated easily. We expect an appropriate setting of γ that can balance both. For this, we have no theoretical result, but in practice we find that $\gamma = 1$ works well in Section 5. Next we discuss how to determine the number of iterations if we employ the discrete heat kernel.

4.4 The Number of Iterations

While we enjoy the advantage of the concise form of the exponential heat kernel, it is better for us to calculate *DiffusionRank* by employing Eq. (6) in an iterative way. Then the problem about determining N–the number of iterations arises:

For a given threshold ϵ, find N such that $||((\mathbf{I}+\frac{\gamma}{N}\mathbf{R})^N - e^{\gamma\mathbf{R}})\mathbf{f}(0)|| < \epsilon$ for any $\mathbf{f}(0)$ whose sum is one.

Since it is difficult to solve this problem, we propose a heuristic motivated by the following observations. When $\mathbf{R} = -\mathbf{I}+\mathbf{P}$, by Eq. (7), we have $(\mathbf{I}+\frac{\gamma}{N}\mathbf{R})^N = (\mathbf{I}+\frac{\gamma}{N}(-\mathbf{I}+\mathbf{P}))^N =$

$$\mathbf{S}^{-1}\begin{pmatrix} 1 & * & * & * \\ 0 & (1+\frac{\gamma(\lambda_2-1)}{N})^N & * & * \\ & & \ddots & \\ 0 & 0 & & * \\ 0 & 0 & 0 & (1+\frac{\gamma(\lambda_n-1)}{N})^N \end{pmatrix}\mathbf{S}. \tag{9}$$

Comparing Eq. (8) and Eq. (9), we observe that the eigenvalues of $(\mathbf{I}+\frac{\gamma}{N}\mathbf{R})^N - e^{\gamma\mathbf{R}}$ are $(1+\frac{\gamma(\lambda_n-1)}{N})^N - e^{\gamma(\lambda_n-1)}$. We propose a heuristic method to determine N so that the difference between the eigenvalues are less than a threshold for only positive λs.

We also observe that if $\gamma = 1, \lambda < 1$, then $|(1+\frac{\gamma(\lambda-1)}{N})^N - e^{\gamma(\lambda-1)}| < 0.005$ if $N \geq 100$, and $|(1+\frac{\gamma(\lambda-1)}{N})^N - e^{\gamma(\lambda-1)}| < 0.01$ if $N \geq 30$. So we can set $N = 30$, or $N = 100$, or others according to different accuracy requirements. In this paper, we use the relatively accurate setting $N = 100$ to make the real eigenvalues in $(\mathbf{I}+\frac{\gamma}{N}\mathbf{R})^N - e^{\gamma\mathbf{R}}$ less than 0.005.

5. EXPERIMENTS

In this section, we show the experimental data, the methodology, the setting, and the results.

5.1 Data Preparation

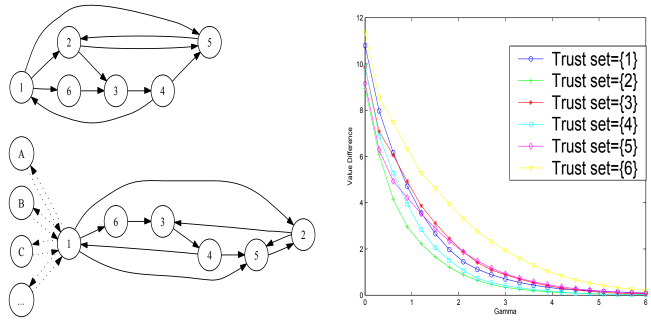
Our input data consist of a toy graph, a middle-size real-world graph, and a large-size real-world graph. The toy graph is shown in Fig. 2 (a). The graph below it shows node 1 is being manipulated by adding new nodes $A, B, C, \ldots$ such that they all point to node 1, and node 1 points to them all. The data of two real Web graph were obtained from the domain in our institute in October, 2004. The total number of pages found are 18,542 in the middle-size graph, and 607,170 in the large-size graph respectively. The middle-size graph is a subgraph of the large-size graph, and they were obtained by the same crawler: one is recorded by the crawler in its earlier time, and the other is obtained when the crawler stopped.

5.2 Methodology

The algorithms we run include *PageRank*, *TrustRank* and *DiffusionRank*. All the rank values are multiplied by the number of nodes so that the sum of the rank values is equal to the number of nodes. By this normalization, we can compare the results on graphs with different sizes since the average rank value is one for any graph after such normalization. We will need value difference and pairwise order difference as comparison measures. Their definitions are listed as follows.

Value Difference. The value difference between $\mathbf{A} = \{A_i\}_{i=1}^n$ and $\mathbf{B} = \{B_i\}_{i=1}^n$ is measured as $\sum_{i=1}^n |A_i - B_i|$.

Pairwise Order Difference. The order difference between $\mathbf{A}$ and $\mathbf{B}$ is measured as the number of significant order differences between $\mathbf{A}$ and $\mathbf{B}$. The pair $(A[i], A[j])$ and $(B[i], B[j])$ is considered as a significant order difference if one of the following cases happens: both $A[i] > [<]A[j]+0.1$ and $B[i] \leq [\geq]A[j]$; both $A[i] \leq [\geq]A[j]$ and $B[i] > [<]A[j]+0.1$.



(a) (b)

Figure 2: (a) The toy graph consisting of six nodes, and node 1 is being manipulated by adding new nodes $A, B, C, \ldots$ (b) The approximation tendency to PageRank by DiffusionRank

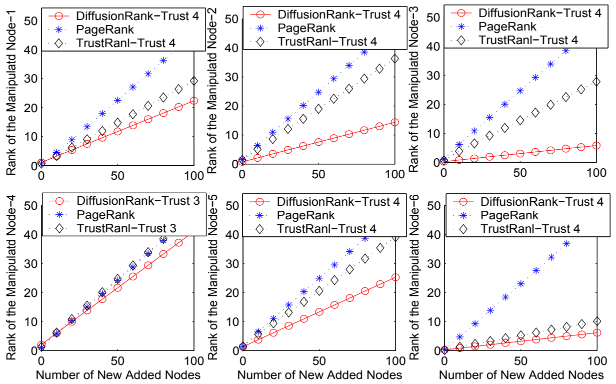
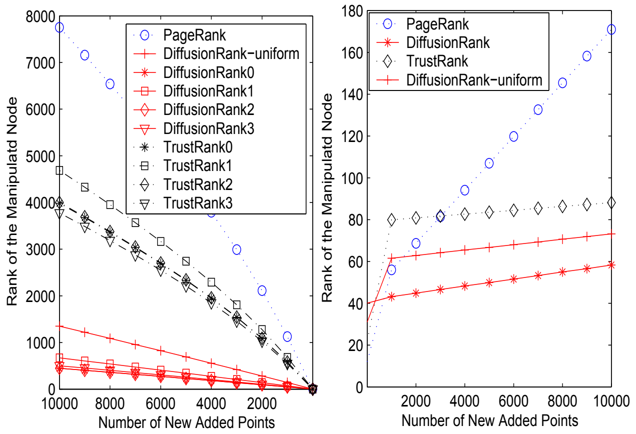


Figure 3: The rank values of the manipulated nodes on the toy graph



(a) (b)

Figure 4: (a) The rank values of the manipulated nodes on the middle-size graph; (b) The rank values of the manipulated nodes on the large-size graph

5.3 Experimental Set-up

The experiments on the middle-size graph and the large-size graphs are conducted on the workstation, whose hardware model is Nix Dual Intel Xeon 2.2GHz with 1GB RAM and a Linux Kernel 2.4.18-27smp (RedHat7.3). In calculating *DiffusionRank*, we employ Eq. (6) and the discrete approximation of Eq. (4) for such graphs. The related tasks are implemented using C language. While in the toy graph, we employ the continuous diffusion kernel in Eq. (4) and Eq. (5), and implement related tasks using Matlab.

For nodes that have zero out-degree (dangling nodes), we employ the method in the modified *PageRank* algorithm [8], in which dangling nodes of are considered to have random links uniformly to each node. We set $\alpha = \alpha_I = \alpha_B = 0.85$ in all algorithms. We also set $\mathbf{g}$ to be the uniform distribution in both *PageRank* and *DiffusionRank*. For *DiffusionRank*, we set $\gamma = 1$. According to the discussions in Section 4.3 and Section 4.4, we set the iteration number to be $M_B = 100$ in *DiffusionRank*, and for accuracy consideration, the iteration number in all the algorithms is set to be 100.

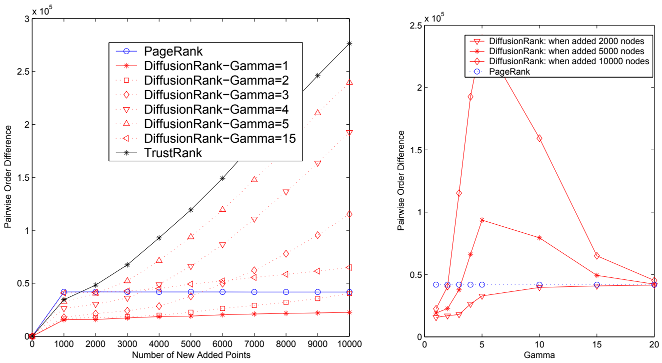
5.4 Approximation of PageRank

We show that when γ tends to infinity, the value differences between *DiffusionRank* and *PageRank* tend to zero. Fig. 2 (b) shows the approximation property of *DiffusionRank*, as proved in Theorem 1, on the toy graph. The horizontal axis of Fig. 2 (b) marks the γ value, and vertical axis corresponds to the value difference between *DiffusionRank* and *PageRank*. All the possible trusted sets with $L = 1$ are considered. For $L > 1$, the results should be the linear combination of some of these curves because of the linearity of the solutions to heat equations. On other graphs, the situations are similar.

5.5 Results of Anti-manipulation

In this section, we show how the rank values change as the intensity of manipulation increases. We measure the intensity of manipulation by the number of newly added points that point to the manipulated point. The horizontal axes of Fig. 3 stand for the numbers of newly added points, and vertical axes show the corresponding rank values of the manipulated nodes. To be clear, we consider all six situations. Every node in Fig. 2 (a) is manipulated respectively, and its corresponding values for *PageRank*, *TrustRank* (TR), *DiffusionRank* (DR) are shown in the one of six sub-figures in Fig. 3. The vertical axes show which node is being manipulated. In each sub-figure, the trusted sets are computed below. Since the inverse *PageRank* yields the results $[1.26, 0.85, 1.31, 1.36, 0.51, 0.71]$. Let $L = 1$. If the manipulated node is not 4, then the trusted set is $\{4\}$, and otherwise $\{3\}$. We observe that in all the cases, rank values of the manipulated node for DiffusionRank grow slowest as the number of the newly added nodes increases. On the middle-size graph and the large-size graph, this conclusion is also true, see Fig. 4. Note that, in Fig. 4 (a), we choose four trusted sets ($L = 1$), on which we test *DiffusionRank* and *TrustRank*, the results are denoted by DiffusionRank$i$ and TrustRank$i$ ($i = 0, 1, 2, 3$ denotes the four trusted set); in Fig. 4 (b), we choose one trusted set ($L = 1$). Moreover, in both Fig. 4 (a) and Fig. 4 (b), we show the results for *DiffusionRank* when we have no trusted set, and we trust all the pages before some of them are manipulated.

We also test the order difference between the ranking order A before the page is manipulated and the ranking order PA after the page is manipulated. Because after manipulation, the number of pages changes, we only compare the common part of A and PA. This experiment is used to test the stability of all these algorithms. The less the order difference, the stabler the algorithm, in the sense that only a smaller part of the order relations is affected by the manipulation. Figure 5 (a) shows that the order difference values change when we add new nodes that point to the manipulated node. We give several γ settings. We find that when $\gamma = 1$, the least order difference is achieved by DiffusionRank. It is interesting to point out that as γ increases, the order difference will increase first; after reaching a maximum value, it will decrease, and finally it tends to the *PageRank* results. We show this tendency in Fig. 5 (b), in which we choose three different settings—the number of manipulated nodes are 2,000, 5,000, and 10,000 respectively. From these figures, we can see that when $\gamma < 2$, the values are less than those for *PageRank*, and that when $\gamma > 20$, the difference between *PageRank* and *DiffusionRank* is very small. After these investigations, we find that in all the graphs we tested, *DiffusionRank* (when $\gamma = 1$) is most robust to manipulation both in value difference and order difference. The trust set selection algorithm proposed in [7] is effective for both *TrustRank* and *DiffusionRank*.



(a) (b)

Figure 5: (a) Pairwise order difference on the middle-size graph, the least it is, the more stable the algorithm; (b) The tendency of varying γ

6. CONCLUSIONS

We conclude that *DiffusionRank* is a generalization of *PageRank*, which is interesting in that the heat diffusion coefficient γ can balance the extent that we want to model the original Web graph and the extent that we want to reduce the effect of link manipulations. The experimental results show that we can actually achieve such a balance by setting $\gamma = 1$, although the best setting including varying γ_i is still under further investigation. This anti-manipulation feature enables *DiffusionRank* to be a candidate as a penicillin for Web spamming. Moreover, *DiffusionRank* can be employed to find group-group relations and to partition Web graph into small communities. All these advantages can be achieved in the same computational complexity as *PageRank*. For the special application of anti-manipulation, *DiffusionRank* performs best both in reduction effects and in its stability among all the three algorithms.

7. ACKNOWLEDGMENTS

We thank Patrick Lau, Zhenjiang Lin and Zenglin Xu for their help. This work is fully supported by two grants from the Research Grants Council of the Hong Kong Special administrative Region, China (Project No. CUHK4205/04E and Project No. CUHK4235/04E).

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. T. Dumais. Improving web search ranking by incorporating user behavior information. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 19–26, 2006.

[2] R. A. Baeza-Yates, P. Boldi, and C. Castillo. Generalizing pagerank: damping functions for link-based ranking algorithms. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 308–315, 2006.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, Jun 2003.

[4] B. Bollobás. *Random Graphs*. Academic Press Inc. (London), 1985.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning (ICML)*, pages 89–96, 2005.

[6] N. Eiron, K. S. McCurley, and J. A. Tomlin. Ranking the web frontier. In *Proceeding of the 13th World Wide Web Conference (WWW)*, pages 309–318, 2004.

[7] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, editors, *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB)*, pages 576–587, 2004.

[8] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Exploiting the block structure of the web for computing pagerank. Technical report, Stanford University, 2003.

[9] R. I. Kondor and J. D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In C. Sammut and A. G. Hoffmann, editors, *Proceedings of the Nineteenth International Conference on Machine Learning (ICML)*, pages 315–322, 2002.

[10] J. Lafferty and G. Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163, Jan 2005.

[11] C. R. MacCluer. The many proofs and applications of perron's theorem. *SIAM Review*, 42(3):487–498, 2000.

[12] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web (WWW)*, pages 83–92, 2006.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report Paper SIDL-WP-1999-0120 (version of 11/11/1999), Stanford Digital Library Technologies Project, 1999.

[14] H. Yang, I. King, and M. R. Lyu. NHDC and PHDC: Non-propagating and propagating heat diffusion classifiers. In *Proceedings of the 12th International Conference on Neural Information Processing (ICONIP)*, pages 394–399, 2005.

[15] H. Yang, I. King, and M. R. Lyu. Predictive ranking: a novel page ranking approach by estimating the web structure. In *Proceedings of the 14th international conference on World Wide Web (WWW) - Special interest tracks and posters*, pages 944–945, 2005.

[16] H. Yang, I. King, and M. R. Lyu. Predictive random graph ranking on the web. In *Proceedings of the IEEE World Congress on Computational Intelligence (WCCI)*, pages 3491–3498, 2006.

[17] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16 (NIPS 2003)*, 2004.