Algorithms and Incentives for Robust Ranking

Rajat Bhattacharjee *
Stanford University

Ashish Goel †
Stanford University

Abstract

Spam in the form of link spam and click spam has become a major obstacle in the effective functioning of ranking and reputation systems. Even in the absence of spam, difficulty in eliciting feedback and self-reinforcing nature of ranking systems are known problems. In this paper, we make a case for sharing with users the revenue generated by such systems as incentive to provide useful feedback and present an incentive based ranking scheme in a realistic model of user behavior which addresses the above problems. We give an explicit ranking algorithm based on user feedback. Our incentive structure and ranking algorithm ensure that there is a profitable arbitrage opportunity for the users of the system in correcting the inaccuracies of the ranking. The system is oblivious to the source of inaccuracies (benign or malicious), thus making it robust to spam as well as the problems of eliciting feedback and self-reinforcement.

1 Introduction.

Before the advent of the Internet, content generation was channeled through a limited number of publishers, such as book publishers, movie production companies, music companies, newspapers, and magazines. In order to regulate and also advertise the quality of content, a system of content evaluation had evolved. Evaluation in traditional publishing is done primarily by professional reviewers and editors who are paid for their opinions. In contrast to self-publishing, the editors decide which content gets published in accordance with the quality of the content.

Content generation is no longer channeled through a limited number of publishers. Individuals self-publish their views, or articles, or creative pieces using websites, blogs, photograph hosting services, podcasts, etc. The scale and decentralization of the content in the Internet makes the old centralized mode of content evaluation impractical. At the same time this decentralization and the corresponding lack of editorial control at the source makes content evaluation all the more important. This need has played a strong role in the success of search engines [19, 20], which not only search but also rank content, thus playing the role of reviewers. Similarly, recommendation systems, which use similarities in the feedback profiles of users and entities, have been designed and deployed [1].

PageRank [29] uses the link structure of the Internet to rank web pages. The philosophy of this approach is that the quality of a web page is indicated by the quality of the web pages pointing to it. However, interested parties have used it to promote the ranking of their own web pages, for example, by creating complex link structures pointing to a chosen web page. As heuristics have been proposed and implemented to detect these malicious web pages, the techniques used by the search engine optimizers have also gotten better [14] [15]. Detecting these PageRank amplifying structures is equivalent to the sparsest cut problem [32], which is NP-hard [22].

An alternative to link-based methods such as PageRank [29] and Hits [21] is to use the feedback from users (e.g. clicks). This approach is already used in recommendation systems [1]. The difficulty of using feedback/clicks stems from detecting whether the feedback/clicks are coming from genuine users who found the web page useful or are being fraudulenty generated by a single source, a phenomenon known as click spam. Various solutions have been proposed for this problem. However, these in turn have resulted in smarter techniques being used by spammers [2][24][31].

In this paper, we make a case for sharing with users the revenue generated by ranking and recommendation systems as incentive to provide useful feedback. The main contribution of this paper is an incentive based ranking scheme in a realistic model of user behavior. We give an explicit ranking algorithm based on user feedback. The incentive structure ensures that there is a profitable arbitrage opportunity for the users of the system in correcting the inaccuracies of the ranking. The

*Department of Computer Science. Research supported by NSF ITR grant 0428868 and NSF award 0339262. Email: rajatb@stanford.edu.

†Department of Management Science and Engineering and (by courtesy) Computer Science. Research supported by NSF ITR grant 0428868, an NSF CAREER award 0339262, and an Alfred P. Sloan Faculty Fellowship. Email: ashishg@stanford.edu.

system is oblivious to the source of inaccuracies (benign or malicious), thus making it robust to spam. The mechanisms are designed to provide a higher incentive for discovering high quality entities rather than for providing more positive feedback for already established entities.

1.1 Related work. The task of ranking a set of entities can be viewed as an information aggregation problem. Information markets have been successfully used for information aggregation in various settings. An information market approach can be implemented by floating shares of an entity and allowing users to trade them (this would require a separate system for trading, like market scoring rules [16] and explicit participation of the users).

The two main features that distinguish the ranking problem from other information aggregation problems are:

- The lack of a clear objective outcome. In the ranking problem, the market doesn't merely play the role of predicting an event but actually *affects* the event. For example, a higher ranked web-page is likely to get more clicks, all else being equal.
- Moreover, the ranking algorithm, which is part of the ranking scheme (thus can be controlled by the designer), has an influence on the outcome. The users provide the ratings but the ranking algorithm determines how these ratings are used to compute rankings.

While the first difference complicates the problem, the second one provides additional flexibility for solving the problem.

Miller et al. [25] give a framework (similar to information markets) to incentivize honest feedback. They counter the lack of objective outcomes by comparing a user's reviews to that of its peers. However, their approach doesn't address malicious users who don't care about their profits/losses from the information market, or the discovery of good entities which haven't yet attracted much feedback. Also, their mechanism precludes taking advantage of the extra control that a ranking algorithm has in translating ratings into rankings.

Another important aspect of our approach which distinguishes it from prior work is the idea of using the utility generated by a particular entity as an indicator of its quality (more details are in section 2). This in itself doesn't counter the lack of objective outcome, as the utility generated is still tied to the machinations of the market. But it allows us to design an explicit ranking algorithm which complements the incentive structure and results in a robust ranking scheme.

We first presented the idea of using incentives to create arbitrage opportunities in a position paper[4]. The work didn't assume any control over the ranking algorithm, instead the argument for robustness depended on certain assumptions about the relation between the feedback scores and the utility generated by entities. In this paper, we design a complete system including an explicit ranking algorithm and an incentive structure.

1.2 The I-U model. We assume an idealized user behavior that we call the Inspect-Utilize (or I-U) model. The I-U model is inspired by Null [28]. Here we give a broad overview of the model (formal details are in section 2). We say that the user u has *inspected* an entity e when u views e as presented by the ranking system. For example, reading the snippet appended to a search result is an act of inspection. We say a user u *utilizes* an entity when u performs a task like reading, registering or downloading the entity. We assume that the probability of utilization of an entity e, conditioned on e being inspected, is only dependent on the inherent quality of e. We refer to this event as a *utility generation event.* We view each utility generation event as having a revenue equivalent. For example, in the case of ranking systems like Google[19] and Yahoo[20], the revenue u generates for the system from the ads on the right hand side can be seen as being the aggregate result of the various web pages utilized by u on the left hand side. Thus the revenue equivalent of a utility generation event is the ratio of the estimated rate of revenue generated by u and the estimated rate of web page utilizations. In case of online retailers like Amazon.com [18], inspection is the viewing of a book recommended by the system and utilization is the purchase of the book. The revenue equivalent of the event is the profit made by the system from the purchase of the book. Other interpretations of the revenue equivalent are also possible.

We believe that although majority of the users follow the rankings/recommendations in the inspection of entities, some users do inspect entities independent of the rankings. We refer to the users in the first group as *sheep* and those in the latter as *connoisseurs.* In the context of web search, a connoisseur would be a user who wouldn't merely depend upon search engine ranking but would use more specific keywords or otherwise targeted searches to find the best source of the information she is seeking. In the context of a news article recommendation system, a connoisseur would be a user who is interested and informed in a particular topic, and actively seeks out interesting news articles on this topic from diverse sources. Note that the same user can be a connoisseur for a certain topic and a sheep for another.

1.3 Our results.

The major problems faced by ranking and recommendation system are spam, the difficulty in eliciting feedback and the self-reinforcing nature of ranking systems. The goal of this work is to address these problems. In section 2.1, we list the reasons for our belief that incentives for feedback are necessary for a robust ranking system. We present a ranking algorithm (section 3) and an incentive structure (section 4) which together ensure a robust ranking system. The ranking algorithm takes as input feedback scores of entities and outputs a probabilistic distribution over the rankings of entities. Given a distribution over rankings $\mathcal{D}$, let the random variable X_e give the position of entity e in a ranking picked at random from $\mathcal{D}$. We say that entity e_1 *dominates* e_2 if X_{e_1} stochastically dominates X_{e_2}, that is, $\forall t,\ Pr[X_{e_1} \leq t] \geq Pr[X_{e_2} \leq t]$[1]. We say that a distribution over rankings is a *proper ranking* if for every pair of entities e_i and e_j, the feedback score of e_i being greater than or equal to the feedback score of e_j implies that e_i dominates e_j. Intuitively, if the user chooses to inspect any of the top l slots (for any l), then the probability of e_i being inspected is higher than the probability that e_j is inspected. The result on the ranking algorithm is stated in theorem 3.5. We present a ranking algorithm which runs in polynomial time and outputs a proper ranking. Moreover, in the common case when the normalized feedback score is majorized by the normalized expected inspection rates of slots (see section 3.1), the rankings ensure that the normalized number of inspections of an entity is equal to the normalized feedback score. More importantly, the ranking algorithm facilitates the design of a robust incentive structure, as described below.

The main result of this paper is stated in theorem 4.1. We give an incentive structure, which in conjunction with the ranking algorithm provides a profitable arbitrage opportunity for the users to correct inaccuracies in the system. In particular, the system *ranks by quality*, that is, either the feedback scores (thus the actual ranking) of all the entities are ranked by quality, or the users can profit by leaving negative feedback for a high ranked, low quality entity and positive feedback for a low ranked, high quality entity. Again if the majorization condition holds, we quantify the exact revenue opportunities for the users. If the cost of gaming the system is less than the profit made from the outcome of the ranking system, then there is an economic incentive for spam. Our system is *resistant to gaming* – inaccuracies produced by gaming are merely more arbitrage opportunities for users. These and other properties are discussed in section 4.3.

In section 2, we present the I-U model and the case for incentives. In section 3, we give the ranking algorithm and in section 4, the incentive structure and a discussion of the properties of the system.

2 The I-U model.

Consider the following interaction of a user u with a ranking system. A query from a user results in k entities (*e.g.* web pages) being shown to the user in the form of a ranking (ordered list). From these k entities, u may choose to inspect the top j entities. Among these j entities, u utilizes a subset S. We say that a utility generation event has occurred for the entities in S (more than one entity can be utilized in a single query). We now present a model with the purpose of capturing the interaction just described. The model has the following specifics.

1. **Entities.** The set of entities $\mathcal{E}$ is the universe from which the ranking system returns the result to a query. We denote the ith entity by e_i. Let the number of entities be n. Each entity has an inherent quality q_i which is *not* known. The quality q_i is the probability that e_i is utilized by a user u, conditioned on the fact that u inspects e_i.
2. **Users.** Let $\mathcal{U}$ be the set of users in the system, with the ith user denoted by u_i. We assume that the users are registered with the system. The users provide feedback (positive and negative) on the entities. We classify users as sheep and connoisseurs. This classification is inspired by the well known difficulty of eliciting useful feedback from users [11][30].
   - (a) *Sheep.* The label sheep is associated with a user who provides feedback on entities among those shown to her (placed at the top of the ranking). In particular, a high quality entity which is not shown to a sheep would not get any feedback from her.
   - (b) *Connoisseurs.* In contrast, a connoisseur is a user who would find a good quality entity even when it is not shown to her. We assume the ratio of the number of connoisseurs to the number of sheep in the system be ϵ. Typically, we expect ϵ to be small.
3. **Feedback.** The notion of feedback is captured by tokens. When a user leaves a positive feedback on entity e_i, the number of tokens placed on e_i (denoted by τ_i) is incremented by 1. Similarly, a

[1] Our definition of stochastic dominance is inverted (compared to the traditional definition) because here the numerically lower slots are more important.

negative feedback results in τ_i being decremented by 1. Users are not allowed to place negative tokens on an entity with 0 net (positive - negative) tokens.

4. **Slots.** Slots are the placeholders for the results of a query. Let the number of slots be k. If $i < j$, then slot i is more important than slot j (this will be made precise next). Since it doesn't make sense to have more slots than entities, we assume that $k \leq n$.

5. **Utility.** A query results in entities being placed in the k slots. The user u chooses to inspect the top j slots. If u utilizes an entity e_i placed in one of the top j slots, we say a utility generation event has occurred for e_i. Let G_i be the indicator variable of this event. The random variable G_i is given by $I_{r(i)}U_i$ (described next). The random variable U_i is a Bernoulli random variable with mean q_i, the quality of e_i. The function $r(i)$ gives the slot in which e_i is placed. If e_i is not placed in any slot, then $r(i)$ is ∞. For each query result, a random number j is chosen from $\{1, 2, \ldots, k\}$. The random number j is independent across trials. The underlying distribution for this process is *not* known but is independent of the entities placed in the slots. If $i \leq j$, $I_i = 1$, else $I_i = 0$. The random variables I_i's are Bernoulli random variables but are not independent of each other (for a given query). We assume that $E[I_1], E[I_2], \ldots, E[I_k]$ are known and denote these quantities by $p_1, p_2, \ldots, p_k$, respectively. In simple words, we assume that the expected number of inspections a slot attracts is known. Note that the random process which sets the variables, $I_1, I_2, \ldots, I_k$, implies that $p_1 \geq p_2 \geq \cdots \geq p_k$. As discussed in the introduction, we assume that there exists a procedure to determine the revenue equivalent of each utility generation event.

2.1 Case for incentives. In this section, we present the reasons for our belief that incentives are necessary for the proper functioning of a ranking system based on user feedback. There are three main reasons for our position. (1) The difficulty of eliciting useful feedback from users is well known [11][30]. In a similar vein, it has been shown that search engine results influence the popularity of web pages [8][9]. (2) The feedback profile of an entity plays an important role in attracting future users. This gives a strong incentive for groups associated with the entity to leave fraudulent positive feedback for it. In the context of reputation systems, this phenomenon is known as ballot stuffing and bad mouthing [3][10]. In the context of web page ranking, this phenomenon is known as click spam [2][31]. We believe that solutions proposed to solve this problem would lead to a heuristic race in the lines of PageRank. (3) The problem of new users in a system has been studied in the literature on reputation systems [13]. Similar phenomenon may occur in ranking systems as well $-$ new entities can be added (new web pages are created all the time), or, there might be a sudden change in the relevance of an entity. For example, articles on a certain individual might suddenly become very relevant when he/she is nominated for some important post. Even if a system could be made spam-free, it can be shown that for small values of ϵ (the fraction of connoisseurs in the system), an entity would take an impractically large amount of time to attain a position in the ranking which is in accordance with its quality [4].

3 Ranking algorithm.

The ranking algorithm takes as input the feedback scores of the entities and returns a distribution on rankings. In response to a query, the system picks a ranking at random from this distribution and shows it to the user. Remember that in our model, the feedback score of an entity e_i is the net (positive $-$ negative) number of tokens (τ_i) that have been placed on e_i. The system has a parameter $s > 1$. The role of s will become clear when we discuss the incentive structure. In this paper, all matrices and vectors are over $\mathcal{R}^+$. We add dummy slots $k+1, k+2, \ldots, n$ to the system. The values $p_{k+1}, p_{k+2}, \ldots, p_n$ (the expected number of inspections of these slots) are set to 0. We assume that the entities and slots are arranged such that $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_n$ and $p_1 \geq p_2 \geq \cdots \geq p_n$.

3.1 Majorization. We say that $x = \{x_1, \ldots, x_n\}$ is a *tapering vector* if $x_1 \geq x_2 \geq \cdots \geq x_n$. The tapering vector x is said to be majorized by the tapering vector y if for $j = 1, \ldots, n$, $\sum_{i=1}^{j} x_i \leq \sum_{i=1}^{j} y_i$. Let $T = \tau_1 + \tau_2 + \cdots + \tau_n$ and $S = p_1 + p_2 + \cdots + p_n$. We use τ/T and p/S to denote the vectors $\{\tau_1/T, \tau_2/T, \ldots, \tau_n/T\}$ and $\{p_1/S, p_2/S, \ldots, p_n/S\}$, respectively. In typical applications, the inspections are concentrated on the top slots. Also, the number of entities is order of magnitude more than the number of "significant" slots. Thus we believe that for most applications τ/T is majorized by p/S. Although our algorithm (presented in the next section) doesn't make the assumption that τ/T is majorized by p/S but we get stronger results if this condition holds.

The following famous theorem by Hardy, Littlewood and Pólya [17][23] gives the connection between majorization and doubly stochastic matrices.

Theorem 3.1. (Hardy, Littlewood, Polya) *The following two statements are equivalent: (1) The vector x is majorized by the vector y, (2) There exists a doubly stochastic matrix, D, such that $x = Dy$.*

We need a stronger version of the above theorem for our results. Let 1_i denote the vector with n components, first i of which are 1 and the rest are 0. We say that a doubly stochastic matrix $D = \{d_{ij}\}$ is *prioritized* if for all $l \leq m$ and $j = 1, \ldots, n$, $\sum_{i=1}^{j} d_{li} \geq \sum_{i=1}^{j} d_{mi}$. In other words, D is prioritized if for $i = 1, \ldots, n$, $D1_i$ is a tapering vector. Observe that if we interpret d_{ij} as the fraction of time entity i is placed in slot j, then the notion of a prioritized doubly stochastic matrix corresponds to the notion of proper ranking (in a fractional sense). Using this notation, a matrix is doubly stochastic if $D1_n = D^T 1_n = 1_n$.

Lemma 3.1. *If D_1 and D_2 are two $n \times n$ prioritized doubly stochastic matrices, then $D_1 D_2$ is also a prioritized doubly stochastic matrix.*

Proof. We first show that $D_1 D_2$ is a doubly stochastic matrix. This follows from $D_1 D_2 1_n = D_1 1_n = 1_n$ and $(D_1 D_2)^T 1_n = D_2^T D_1^T 1_n = D_2^T 1_n = 1_n$. Since D_2 is prioritized, the vector $D_2 1_i$ is a tapering vector and can be written as $\sum_{i=1}^{n} \alpha_i 1_i$, where $\alpha_i \geq 0$ (for $i = 1, \ldots, n$). Therefore $D_1 D_2 1_i$ can be written as $\sum_{i=1}^{n} \alpha_i v_i$, where v_i's are tapering vectors themselves. This implies that $\sum_{i=1}^{n} \alpha_i v_i$ is also a tapering vector. Thus $D_1 D_2$ is a prioritized doubly stochastic matrix. □

Theorem 3.2. *If the tapering vector x is majorized by the tapering vector y and $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i = 1$, then there exists a prioritized doubly stochastic matrix D such that $x = Dy$.*

Proof. The proof idea is to transform $y \rightsquigarrow u_1 \rightsquigarrow u_2 \rightsquigarrow \cdots \rightsquigarrow u_r \rightsquigarrow x$, where $u_1, \ldots, u_r$ are tapering vectors and $u_1 = D_1 y, u_2 = D_2 u_1, \ldots, u_r = D_r u_{r-1}, x = D_{r+1} u_r$ (the matrices $D_1, D_2, \ldots, D_{r+1}$ are prioritized and doubly stochastic). Since $\sum_{i=1}^{n} y_i = 1$ and all the D_is are doubly stochastic, the components of each of the intermediate vector sum to 1. Suppose $D = D_{r+1} D_r \ldots D_1$. Observe that $Dy = x$ and by lemma 3.1, D is a prioritized and doubly stochastic matrix. Now we only need to give a procedure which given a tapering vector z (which majorizes x) outputs a prioritized doubly stochastic matrix D' such that $z' = D'z$ majorizes x and z' is a tapering vector. Also, we need to prove that by repeatedly applying this procedure we will get x in finite number of steps.

We denote the ith component of z by z_i. If $z_1 \neq x_1$, set $l = 0$, else set $l = \sup\ \{i | \forall j \leq i, z_j = x_j\}$. Set $m = \sup\ \{i | \forall j = 1, \ldots, i, z_{l+j} = z_{l+1}\}$. Suppose $(z_{l+1} + z_{l+2} + \cdots + z_{l+m} + z_{l+m+1})/(m+1) \geq x_{l+1}$. Then we set $D' = \{d_{ij}\}$ as follows. If $i \leq l$ or $i > l+m+1$, set $d_{ii} = 1$. For $l+1 \leq i \leq l+m+1$ and $l+1 \leq j \leq l+m+1$, $d_{ij} = 1/(m+1)$. For all other positions $d_{ij} = 0$ (the matrix looks like the first matrix in figure 1). It is easy to verify that D' is a prioritized doubly stochastic matrix. Suppose $z' = D'z$ and l', m' be analogously defined for z'. Since $z'_{l+m+1} = z'_{l+1}$, $l' + m' \geq l + m + 1$. We need to verify that z' is a tapering vector which majorizes x. Since $z'_1 = z_1, \ldots, z'_l = z_l$ and $z'_{l+1} + \cdots + z'_{l+m+1} = z_{l+1} + \ldots z_{l+m+1}$, for $i \leq l$ or $i \geq l+m+1$, $\sum_{j=1}^{i} z'_j = \sum_{j=1}^{i} z_j \geq \sum_{j=1}^{i} x_j$. For $i = 1, \ldots, m+1$, $\sum_{j=1}^{i} z'_{l+j} = i z'_{l+1}$. By the above assumption ($z'_{l+1} \geq x_{l+1}$) and the fact that x is a tapering vector, we have $\sum_{j=1}^{i} z'_{l+j} \geq \sum_{j=1}^{i} x_{l+j}$. To verify that z' is a tapering vector, we only need to worry about the boundary cases: $z'_l = z_l \geq z_{l+1} \geq z'_{l+1}$ and $z'_{l+m+1} \geq z_{l+m+1} \geq z_{l+m+2} = z'_{l+m+2}$.

$$\begin{bmatrix} 1 & & & & & & & 0 \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & \frac{1}{m+1} & \cdots & \frac{1}{m+1} & & \\ & & & \vdots & & \vdots & & \\ & & & \frac{1}{m+1} & \cdots & \frac{1}{m+1} & & \\ & & & & & & 1 & \\ & & & & & & & \ddots \\ 0 & & & & & & & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & & & & & & & & 0 \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \alpha & \cdots & \alpha & 1 - m\alpha & & \\ & & & \vdots & & \vdots & \vdots & & \\ & & & \alpha & \cdots & \alpha & 1 - m\alpha & & \\ & & & 1 - m\alpha & \cdots & 1 - m\alpha & 1 - m + m^2\alpha & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ 0 & & & & & & & & 1 \end{bmatrix}$$

Figure 1: The two transformation matrices.

Now suppose $(z_{l+1} + z_{l+2} + \cdots + z_{l+m} + z_{l+m+1})/(m+1) < x_{l+1}$. Then there exists an $\alpha \in [1/(m+1), 1/m]$ such that $\alpha(z_{l+1} + z_{l+2} + \ldots z_{l+m}) + (1 - \alpha m) z_{l+m+1} = x_{l+1}$, because for $\alpha = 1/(m+1)$ the left hand side is less than x_{l+1} and for $\alpha = 1/m$ the left hand side is equal to $z_{l+1} \geq x_{l+1}$ (z majorizes x and $z_i = x_i$, for $i = 1, \ldots, l$). We set $D' = \{d_{ij}\}$ such that $d_{ii} = 1$ if $i \leq l$ or $i > l+m+1$. For $l+1 \leq i \leq l+m$ and $l+1 \leq j \leq l+m$, $d_{ij} = \alpha$. For $l+1 \leq i \leq l+m$, $j = l+m+1$, $d_{ij} = d_{ji} = 1 - m\alpha$. For $i = l+m+1$, $d_{ii} = 1 - m + m^2\alpha$. For all other positions $d_{ij} = 0$ (the matrix looks like the second matrix in figure 1). For the same reasons as in the previous case $z' = D'z$ is a tapering vector. Using the facts $z'_{l+1} = x_{l+1}$, $z'_{l+m+1} \geq z_{l+m+1} \geq x_{l+m+1}$

and similar arguments as above, we can show that z' majorizes x. The rows and columns of D' sum to 1, so D' is doubly stochastic. To verify that D' is prioritized, observe that $\alpha \geq 1/(m+1) \geq 1-m\alpha$. Since $z'_{l+1} = x_{l+1}$, l' corresponding to the vector z' is greater than or equal to $l+1$. This proves the existence of the procedure. We are left with proving that repeated applications of this procedure would give us x.

Obviously, if $l = n$ then $z = x$. Suppose $l + m = n$. Since $z_1 + \cdots + z_n = x_1 + \cdots + x_n = 1$ and $z_1 + \cdots + z_l = x_1 + \cdots + x_l$, we get $z_{l+1} + \cdots + z_n = x_{l+1} + \cdots + x_n$. However, $z_{l+1} = z_{l+2} = \cdots = z_n$ and x is a tapering vector. Therefore, $z_i = x_i$ for $i = l+1, \ldots, n$. With each transformation we either increment l or $l + m$, thus after at most $2n$ transformations z will be equal to x. □

We first give a program to obtain the matrix D. Then we discuss how the distribution over the rankings can be extracted from D.

3.2 Obtaining D. The following program takes as input $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_n$ and $p_1 \geq p_2 \geq \ldots p_n$ (with at least one non-zero component). It outputs the matrix $D = \{d_{ij}\}$. Interpret d_{ij} is the fraction of time entity i is placed in slot j.

minimize $\quad f(D, \tau)$
subject to:

$$\sum_{i=1}^{n} d_{ij} = 1, \quad j = 1, \ldots, n,$$

$$\sum_{j=1}^{n} d_{ij} = 1, \quad i = 1, \ldots, n,$$

$$\sum_{j=1}^{n'} d_{lj} \geq \sum_{j=1}^{n'} d_{mj}, \quad \text{for all } l \leq m \text{ and } n' = 1, \ldots, n,$$

$$\frac{1}{\tau_l^s} \sum_{j=1}^{n} d_{lj} p_j \leq \frac{1}{\tau_m^s} \sum_{j=1}^{n} d_{mj} p_j, \quad \text{for all } l \leq m,$$

$$d_{ij} \geq 0, \quad i = 1, \ldots, n, \;\; j = 1, \ldots, n.$$

The first constraint arises from the fact that only one entity can be placed in a slot. The second constraint says that an entity cannot be placed twice in the ranking. The third constraint captures the fact that the matrix is prioritized. Note that the constraints capture these conditions in a fractional sense, however, we will show in the next subsection that these constraints do imply that the conditions are true for each ranking in the distribution. The importance of the fourth constraint will be clear when we discuss the incentive structure. Since s, τ and p are given, all the constraints are linear.

Let $\eta_i = \sum_{j=1}^{n} d_{ij} p_j$. In other words η_i is the expected number of inspections that entity i receives. Observe that the fourth constraint is equivalent to $\eta_l/\tau_l^s \leq \eta_m/\tau_m^s$ (when $l \leq m$). Ideally, we wish η/S to be equal to τ/T, that is, the normalized number of inspections an entity should be equal to the normalized feedback score. We will show that this is indeed true when p/S majorizes τ/T. However, in the general case we can only hope to get something close and that is the purpose of function f. The function f is chosen such that $f(D, \tau) \geq 0$ for all D and τ, and $f(D, \tau) = 0$ if and only if $\eta/S = \tau/T$. Possible functions that can be used are $\|\eta/S - \tau/T\|_2$ and $\|\eta/S - \tau/T\|_\infty$. For other such functions and their relative merits, see [7].

Theorem 3.3. *If p has at least one non-zero component, then the above program is feasible. Moreover, if p/S is majorized by τ/T, it returns $\{d_{ij}\}$ such that $\eta/S = \tau/T$.*

Proof. The matrix $D := \{d_{ij} = 1/n\}$ is feasible. Since $D1_i = (i/n)1_n$, $D1_n = D^T 1_n = 1_n$. Hence, D is a prioritized doubly stochastic matrix and the first three constraints are satisfied. For the fourth constraint, notice that $\eta_l = \eta_m$ for all pairs (l, m). Since $l \leq m$, $\tau_l \geq \tau_m$, $\eta_l/\tau_l^s \leq \eta_m/\tau_m^s$, thus the fourth constraint holds as well.

Suppose τ/T is majorized by p/S. By theorem 3.2, there exists a prioritized doubly stochastic matrix D such that $Dp/S = \tau/T$. Since $\eta = Dp$, we get $\eta = \tau S/T$. Now $f(D, \tau) = 0$ if and only if $\eta = \tau S/T$, thus the optimal D is such that $\eta = \tau S/T$. Since D is a prioritized doubly stochastic matrix, we only need to show that the fourth constraint is satisfied by D. Suppose $l \leq m$. Now $\eta_l/\tau_l^s = \tau_l^{1-s} S/T$ and $\eta_m/\tau_m^s = \tau_m^{1-s} S/T$. Since $\tau_l \geq \tau_m$ and $s > 1$, $\eta_l/\tau_l^s \leq \eta_m/\tau_m^s$, thus D satisfies the fourth constraint as well. □

3.3 Extracting the distribution. Given D we wish to find a distribution on the rankings (which is the same as convex combination of permutations). The following famous theorem by Birkhoff [6] and von Neumann [27] shows the existence of such a convex combination of permutations.

Theorem 3.4. (Birkhoff, von Neumann) *An $n \times n$ matrix is doubly stochastic if and only if it is a convex combination of permutation matrices.*

In other words, there exist positive numbers $\theta_1, \theta_2, \ldots, \theta_m$ and permutation matrices $P_1, P_2, \ldots, P_m$ such that $D = \sum_{i=1}^{m} \theta_i P_i$ and $\sum_{i=1}^{m} \theta_i = 1$. Dumalge and Halperin [12] give an $O(n^{4.5})$ algorithm for obtaining this distribution. It is further known that the number of permutation matrices in the distribution is bounded by $O(n^2)$. Note that we don't need to compute the distribution online. When a query arrives, we merely need to pick a random permutation from this distribution (which can be done efficiently by maintaining indices). The following theorem is now immediate.

THEOREM 3.5. *There exists a polynomial time ranking algorithm which takes as input feedback scores (τ_i's) and outputs a proper ranking. Moreover, if τ/T is majorized by p/S, then η_i/S (the normalized number of inspections of entity e_i) is equal to τ_i/T (the normalized feedback score).*

4 Incentive structure.

Users are allowed to place positive and negative tokens on various entities subject to the constraint that the absolute value of the net (positive - negative) number of tokens a user can place (across entities) is bounded. The feedback scores of the entities (the vector τ) is used to obtain the distribution on rankings using the ranking algorithm described in the previous section. The rankings when shown to the users may result in utility generation events. At each such event, a part of the revenue (equivalent to the utility generated) is shared with the users. Central to the incentive structure are the notions of tokens and revenue distribution which are formalized next.

4.1 Tokens. A token T_i is a three-tuple: $\{p(T_i), u(T_i), e(T_i)\}$. The value $p \in \{+1, -1\}$ specifies whether the token is a positive/negative token. A value of $+1$ indicates that T_i is a positive token and a value of -1 indicates that the token is a negative one. The user who placed the token is determined by $u \in \mathcal{U}$ and $e \in \mathcal{E}$ determines the entity on which the token is placed. The order of arrival of tokens is given by the subscript i. Since in our model time is continuous, we assume that no two tokens arrive at the same time. The only constraint is that at any given time absolute value of the net positive tokens of a user is bounded by γ, which is a system parameter. Note that a user can obtain more positive tokens by placing negative tokens.

4.2 Revenue distribution. The revenue distribution mechanism has two parameters, β and s. The fraction of revenue to be distributed as incentive among the users is determined by $\beta \leq 1$. The parameter $s > 1$ (same as in the ranking algorithm) controls the relative importance of tokens placed earlier on an entity. Suppose a utility generation event occurs for an entity e at time t, and results in R amount of equivalent revenue being generated for the system. Let $\mathcal{T}$ be the set of all the tokens in the system. For a given token T_i, such that $e(T_i) = e$, we define $\alpha(i)$ as the the net number of tokens placed on e just after T_i is placed on e. More formally, $\alpha(i) = \sum_{T_j \in \mathcal{T}: j \leq i, e(T_j)=e} p(T_j)$. The revenue share of user $u(T_i)$ due to token T_i is given by $\frac{p_i \beta R}{\zeta(s)\alpha(i)^s}$, where $\zeta(s)$ is the Riemann zeta-function and is equal to $\sum_{i=1}^{\infty} \frac{1}{i^s}$. Note that the above quantity is positive or negative depending on p_i. For each user u_i, we maintain an account acc_i. Depending on the sign of the revenue share, the amount is added or subtracted from acc_i. The user can cash all or part of this amount at any point (acc_i gets reduced by the amount cashed). However, the user cannot pay the system to get a larger acc_i. In other words, the user cannot pay to gain greater control of the system. The situation arising from the bankruptcy of a user (acc_i falling below 0) is discussed in the Appendix.

LEMMA 4.1. *If R is the revenue equivalent of a utility generation event for entity e, then the revenue shared with the users is bounded by βR.*

Proof. The total revenue shared with the users is given by $\sum_{T_i \in \mathcal{T}: e(T_i)=e} \frac{p_i \beta R}{\zeta(s)\alpha(i)^s}$. The users who placed two tokens i and j, such that $\alpha(i) = \alpha(j)$, will have the same absolute value of revenue share added or deducted. Moreover for a fixed α, the net number (positive $-$ negative) tokens having this value is less than equal to 1. Hence, the total revenue shared with the users is upper bounded by $\sum_{i=1}^{\infty} \frac{\beta R}{\zeta(s) i^s} = \beta R$. $\square$

The expected rate at which revenue is generated for an entity e_i is given by

$$E[I_{r(i)} U_i] = E[I_{r(i)}] E[U_i] = q_i (Dp)_i = q_i \eta_i,$$

where $(Dp)_i$ is the ith component of the vector Dp. We use the fact that the random process which determines the value of $I_1, I_2, \ldots, I_k$ is independent of the quality of the entities placed in the slots. We say the act of putting a negative token on entity e_i and a positive token on entity e_j presents a *profitable arbitrage opportunity* if the expected rate at which revenue is generated by placing a token on e_i is strictly less than that for e_j, that is, $q_i \eta_i / \tau_i^s < q_j \eta_j / \tau_j^s$. We say that pair (i, j) has an *inverted ranking* if $q_i < q_j$ and the expected number of inspections received by e_i is more than that received by e_j (which is equivalent to $\tau_i \geq \tau_j$ due to the third constraint in the program).

THEOREM 4.1. *If the pair (i, j) has inverted ranking, then placing a negative token on entity e_i and a positive token on entity e_j presents a profitable arbitrage opportunity. Moreover, if τ/T is majorized by p/S the profitability of placing a positive token on entity e_i is directly proportional to it's quality (q_i) and inversely proportional to the net number of tokens on it (τ_i).*

Proof. From the discussion above, in order to prove the first claim, we need to show that $\frac{q_i \eta_i}{\tau_i^s} < \frac{q_j \eta_j}{\tau_j^s}$. Since the pair (i, j) has inverted ranking, $q_i < q_j$ and $\tau_i \geq \tau_j$. The fourth constraint of the program ensures that $\tau_i \geq \tau_j$ implies $\frac{\eta_i}{\tau_i^s} \leq \frac{\eta_j}{\tau_j^s}$. Therefore, we get the required inequality. When τ/T is majorized by p/S, $\eta_i = \tau_i S/T$ (theorem 3.3). Therefore, the expected rate of return is given by $q_i \eta_i / \tau_i^s = q_i \tau_i^{1-s} S/T$. The second claim now follows the fact that s is greater than 1. □

4.3 Properties. The ranking algorithm and the incentive structure ensure the following properties of the system.

1. *Ranking by quality.* The system either ranks entities by qualities or if there exists a pair (i, j) which has inverted ranking, then it provides a profitable arbitrage opportunity to users in placing a positive token on e_j and a negative token on e_i, thus correcting the ranking.

2. *Resistance to gaming (spam).* Suppose a malicious user u makes revenue from outside the system by inaccuracies in the system. The gaming of system by u would only lead to more arbitrage opportunities for others, who would have an incentive in correcting the system.

3. *Resistance to racing.* The system is said to be resistant to racing if two users A and B cannot indefinitely repeat actions a_A and a_B, respectively, where a_B undoes the effects of a_A and vice versa. Due to the bound on the number of positive tokens, two users cannot keep adding positive tokens to their chosen entities ad infinitum. Also they cannot continuously keep placing positive token on their chosen entity and negative token on their rival's entity, as one of these actions would have a net negative value and eventually one of them would get bankrupt[2].

4. *Guarantee of position.* The system is designed such that the share of a user u who places a token on entity e is fixed for all future utility generation events on e, irrespective of the actions of other users in the system.

[2] A more explicit approach to avoid "racing" is to multiply any negative revenue by $(1 + \delta)$ where δ is an arbitrarily small positive number. Now even if two players are "racing" on pages which have the same quality and the same number of tokens, one of them will go bankrupt quite quickly. And the property of resistance to gaming will be affected only marginally.

References

[1] G. Adomavicius, A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, June 2005.*

[2] V. Anupam, A. Mayer, K. Nissin, B. Pinkas, M. Reiter. On the security of pay-per-click and other web advertising schemes. In *Proceedings of the 8th International Conference on World Wide Web, 1091-1100, 1999.*

[3] R. Bhattacharjee, A. Goel. Avoiding ballot stuffing in eBay-like reputation systems. *Third workshop on economics of peer-to-peer systems, 2005.*

[4] R. Bhattacharjee, A. Goel. Incentive based ranking mechanisms. *First workshop on the economics of networked systems, June 2006.*

[5] G. Bianconi, A-L. Barbasi. Competition and multiscaling in evolving networks. Europhysics letters, 54(4), 436-442, 2001

[6] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucumán Rev. Ser. A, vol. 5 pp. 147-151, 1946.*

[7] S. Boyd, L. Vandenberghe. *Convex Optimization, Chapter 6. Cambridge University Press 2004.*

[8] J. Cho, S. Roy. Impact of Web search engines on page popularity. In *Proceedings of the Thirteenth International WWW Conference, 2004.*

[9] J. Cho, S. Roy, R. E. Adams. Page quality: In search of an unbiased web ranking. *SIGMOD, 2005.*

[10] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the second ACM Conference on Electronic Commerce, October 2000.*

[11] C. Dellarocas. The digitization of word-of-mouth: promise and challenges of online reputation systems. *Management Science, Oct 2003.*

[12] L. Dumalge, I. Halperin. On a theorem of Frobenius-König and J. von Neumann's game of hide and seek. *Trans. Roy. Soc. Canada III(3), vol. 49, pp. 23-29, 1955.*

[13] E. Friedman, P. Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy, 10(2):173-199. 2001.*

[14] Z. Gyongyi, H. Garcia-Molina. Link Spam Alliances. *31st International Conference on Very Large Data Bases (VLDB).*

[15] Z. Gyongyi, H. Garcia-Molina. Spam: It's not just for inboxes anymore. *IEEE Computer Magazine, 38:10, 28-34.*

[16] R. Hanson Combinatorial Information Market Design. *Information Systems Frontiers, 5:1, 107-119, 2003.*

[17] G. H. Hardy, J. E. Littlewood, G. Pólya. *Inequalities, 2nd edition, 1952, Cambridge University Press, London.*

[18] http://www.amazon.com

[19] http://www.google.com

[20] http://www.yahoo.com

[21] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM, 46(5):604-632, 1999.*

[22] F. T. Leighton, S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM, 46(6):787-832, 1999.*

[23] A. W. Marshall, I. Olkin. *Inequalities: Theory of Majorization and Its Applications, 1979, Academic Press, New York.*

[24] A. Metwally, D. Agrawal, A. El Abbadi. Duplicate detection in click streams. In *Proceeding of the 14th International Conference on World Wide Web, 12-21, 2005.*

[25] N. Miller, P. Resnick, R. Zeckhauser. Eliciting informative feedback: the peer-prediction method. *Management Science 51(9), 2005.*

[26] R. Motwani, Y. Xu. Evolution of page popularity under random web graph models. *Principles of Databases Systems, 2006*

[27] J. von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. In *Contributions to the optimal assignment problem to the Theory of Games. Princeton Univ. Press, 1953, vol. 2, pp.5-12.*

[28] B. Null. A discussion of click-through algorithms for web page ranking. *Available at http://www.stanford.edu/~null/ClickThroughAlg_Tutorial.pdf.*

[29] L. Page, S. Brin, R. Motwani, T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. *Stanford Digital Library Technologies Project, 1998.*

[30] P. Resnick, R. Zeckhauser, E. Friedman, K. Kuwabara. Reputation systems. *Communications of the ACM, 43(12):45-48, December 2000.*

[31] D. Vise. Clicking to steal. *Washington Post Magazine, F01, April 17 2005.*

[32] H. Zhang, A. Goel, R. Govindan, K. Mason, B. Van Roy. Making eigenvector-based reputation systems robust to collusion. *Workshop on Algorithms and Models for the Web Graph (WAW) 2004.*

Appendix: Bankruptcy.

We say that a user has gone bankrupt when his account falls below 0. A bankrupt user, u, is stopped from participating in the system for a pre-defined interval. However, the tokens placed by u are not removed from the system. The tokens are not removed because doing so would interfere with the expectations/understanding of other users who placed tokens on some entity e after u placed a token on e (guarantee of position would fail to hold). This choice has the disadvantage that the system has to pay for the net loss that the bankrupt users would have made. Various heuristics can be suggested for fixing this problem, for example, the system can increase the deduction in case of a negative token by a multiplicative factor of $(1 + \delta)$, where δ is a very small number. This would create a large pool to cushion bankruptcies. Also, the revenue share due to old tokens can be decayed with time, limiting the impact of bankruptcy. We note here that in case of an exception, the system or part of it can be shut down and started afresh (since the system doesn't accept any money from the users, there are no contractual obligations). Each time such an exception occurs, a number of users who jeopardize the system would be weeded out. This does assume that we have some handle on ids of users but the recent methods used in the industry shows that this is feasible.