Incentive Based Ranking Mechanisms

Rajat Bhattacharjee *
Stanford University

Ashish Goel †
Stanford University

Position Paper

Abstract

We consider ranking and recommendation systems based on user feedback. We make a case for sharing the revenue generated by such systems with users as incentive to provide useful feedback. Our main contribution are mechanisms for ranking/recommendation which gives incentive for the users to provide useful feedback and is resistant to selfish/malicious behavior (click spam). The mechanisms are designed to give higher incentives for discovering high quality entities rather than for merely providing additional positive feedback for already established entities. A page whose rating/ranking is at variance with its real quality represents an arbitrage opportunity. The mechanisms are simple enough to be used with existing technology in ranking and recommendation systems, requiring little or no extra effort by the users.

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 like Google [14], Yahoo [16] and many others, which not only search but also rank content, thus playing the role of reviewers. In addition, recommendation systems use similarities in the feedback profiles of users and entities to recommend new items [1].

PageRank [22] uses the link structure of the Internet to rank webpages. The philosophy of this approach is that the quality of a webpage is indicated by the quality of the webpages pointing to it. However, interested parties have used it to promote the ranking of their own webpages, for example, by creating dummy webpages pointing to their own. As heuristics have been proposed and implemented to detect these malicious webpages, the techniques used by the search engine optimizers have also gotten better [10] [11]. Detecting these PageRank amplifying structures is equivalent to the sparsest cut problem [25], which is NP-hard [18].

An alternative to link-based methods such as PageRank [22] and Hits [17] is to use the feedback from users (e.g. clicks). This approach is already used in many recommendation systems [1]. The difficulty of using feedback/clicks stems from detecting whether the feedback/clicks are coming from genuine users who found the webpage useful or are coming from a single source, a phenomenon known as click spam. Various solutions have been

*Department of Computer Science. Email: rajatb@stanford.edu.

†Department of Management Science and Engineering and (by courtesy) Computer Science. Email: ashishg@stanford.edu.

proposed for this problem. However, these in turn have resulted in smarter techniques being used by the spammers [2][19][24].

The main contribution of this paper is two fold: (1) we make a case for sharing with users the revenue generated by such systems as incentive to provide useful feedback. (2) we present a preliminary design of mechanisms for ranking/recommendation systems which give incentive to the users to provide useful feedback. 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. In section 3, we motivate and list desirable properties of a ranking system. The proposed ranking mechanisms (section 4) are shown to possess these properties, in particular they are resistant to click spam. The mechanisms are simple enough to be used with existing technology in ranking and recommendation systems. We begin by giving a generic model for ranking and recommendation systems.

2 A Generic Model for Ranking Systems

In the introduction, we mention the problems of using PageRank for ranking. Here we consider ranking systems which are based on user feedback. A typical ranking system has the following features (similar ideas apply to recommendation systems based on user feedback as well):

1. **Entities.** The set of entities which we wish to rank is denoted by $\mathcal{E}$. We denote the ith entity by e_i. Each entity has an inherent quality denoted by q_i which is *not* known. However, if two entities with qualities q_i and q_j are presented to users with $q_i > q_j$, then more users would find entity i better than entity j.

2. **Users.** The set of users in the system is denoted by $\mathcal{U}$. These users provide feedback on the entities. We denote the ith user in the system by u_i. Note that we implicitly assume that these users are registered with the system. The users are further classified as (this classification is inspired by the well known difficulty of eliciting useful feedback from users [8][23]):

   (a) **Sheep.** The label sheep corresponds to the user who leaves feedback for an entity when the entity is shown to him/her (recommended or ranked highly). A high quality entity which is not shown to a sheep would not get any feedback from that sheep.

   (b) **Connoisseur.** A connoisseur is a user who would find a good quality entity even when it is not shown to him/her. We assume that the ratio of connoisseurs to sheep in the system is ϵ. Typically, we expect ϵ to be small. In the context of web search, a connoisseur would be a user who wouldn't merely depend upon search engine ranking and would use more specific keywords or otherwise targeted searches to find the information he is looking for. In the context of news articles, a connoisseur would be a user who is really interested in a particular topic and would look out for any interesting news article on this topic. Note that the same user can be a connoisseur for a certain topic and a sheep for another.

3. **Feedback.** The notion of feedback is captured by tokens. When a user gives positive feedback for an entity i, the number of tokens placed on the entity, denoted by τ_i, is incremented by 1. We represent the relative number of tokens an entity attracts, $\frac{\tau_i}{\sum_j \tau_j}$, by r_i.

4. **Revenue/Utility.** We identify the revenue/utility generated by an entity i with the rate at which the sheep leave positive feedback for that entity. The rate at which revenue is generated by entity i is given by the revenue function, $f(r_i, q_i)$. In general, the revenue function is *not* known but we do know the revenue generation event for each entity. The function is assumed to be non-decreasing in r_i and increasing in q_i. In other words, if two entities have the same share of tokens but the quality of first entity is better than the second, then the first entity generates more revenue than the second. Similarly, if two entities have the same quality but one has greater share of tokens, then the revenue generated by the first is more than the second. Implicit in these assumptions is the fact that the revenue function is a good indicator of the utility that an entity generates for the system. For example, this might not be the case in the pay-per-click model for ad-funded ranking systems, however, it would be a good indicator in the pay-per-acquisition

model. The revenue/utility can be of three kinds: (1) recommendations can be directly related to the purchase of goods, e.g. in the case of e-merchants like Amazon.com [13], thus better recommendations would lead to better revenues, (2) rankings and recommendations would lead to increased consumer satisfaction, thus attracting more users and the number of users in the system is directly related to the revenue generates, e.g. in the case of service providers who charge users for membership such as Netflix [15], and (3) the relationship with actual revenue generation can be more abstract like in the case of ad-funded search engines such as Google [14], where consumer satisfaction increases the number of users, which is then translated into revenue through ads.

In most cases we have some knowledge of function f. For example, an interesting case is when $f(r,q) = rq$, which arises when the probability that a webpage gets viewed is given by r and the conditional probability that the page gets clicked is given by q (this function also arises in related settings [21][4]). We define a general class of functions which includes the above function. A function f is said to be a *separable function* if $f(r,q) = qr^{\alpha}$, for some $\alpha > 0$.

3 Desiderata

In this section, we motivate and list properties that a ranking system should have. In section 4.1.3, we formalize these properties.

1. A ranking system should result in a ranking which is in accordance with the quality of the entities. More precisely, if for two entities i and j, $q_i > q_j$, then the ranking of the entities should be such that entity i is positioned before entity j. We call this property *ranking by quality.*

2. Groups associated with a particular entity might have interest in promoting its rank irrespective of its quality. For example, the owner of a hotel in Bali would like the webpage of his hotel to be one of the first few webpages which show up when a user searches for Bali. Also some groups might be interested in demoting the rank of a certain entity. For example, the owner of a rival hotel might try to lower the ranking of the webpage of the other hotel. Thus, a ranking system should be resistant to such selfish/malicious behavior. We call this property *resistance to gaming.*

3. Imagine two entities of similar quality (one can think of two news providing webpages) with huge resources. The number of users these two entities attract would then depend upon their relative ranking. An item which is slightly lower in ranking might succeed in improving its ranking by using the knowledge of how the ranking system works. For example, in case of PageRank, the entity would try to make sure that more webpages point to it. In case the ranking system uses click through analysis, the entity might try to fraudulently generate more clicks. In response the rival entity might indulge in similar practices to restore the relative ranking. This cycle can repeat endlessly making the ranking system unstable. A good ranking system should not foster such behavior. We call this property *resistance to racing.*

3.1 Case for Incentives

We believe 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 [8][23]. In a similar vein, it has been shown that search engine results influence the popularity of webpages [5][6]. (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] [7]. In the context of webpage ranking, this phenomenon has been studied in the literature under the name of click spam [2][24]. 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 reputation systems [9]. Similar phenomenon may occur in ranking systems as well. New entities can be added (new webpages 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, or articles on a certain stock might suddenly be sought after a surprise declaration of strong earnings. Even if technology could be developed for combating click spam, 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. Please see Appendix for a more detailed analysis.

4 Incentive Based Mechanisms for Ranking

We first present the mechanisms for ranking systems based on incentives. We then show that the ranking system has the properties we listed in section 3 (and which we also formalize in section 4.1.3), in particular, it is resistant to click spam.

4.1 Ranking Mechanisms

Users are allowed to place positive and negative tokens on various entities subject to some constraints. The ranking of entities is updated based on the knowledge of the number of tokens placed on the entities. The ranking results in revenue generation events. At each such event, a part of the revenue is shared with the users. Central to the ranking mechanism are the notions of tokens and incentives. We first formalize the notion of tokens and then describe the incentives. Finally, we describe the mechanisms for placing tokens.

4.1.1 Tokens

A token T_i is a five tuple

$$\{p_i, u_i, e_i, w_i, t_i\}.$$

The value $p_i \in \{+1, -1\}$ 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 $u_i \in \mathcal{U}$ and $e_i \in \mathcal{E}$ determines the entity on which the token is placed. The weight of a token (chosen by the user while placing the token) is given by $w_i \in \mathcal{R}^+$ and the time at which the token is placed is given by $t_i \in \mathcal{R}^+$. The order of arrival of tokens is given by the subscript i. We assume that no two tokens arrive at the same time. The only constraint is that at any given time 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 (for potental future placement) by placing negative tokens.

4.1.2 Incentives

In this section, we describe how revenue is shared among the users. Suppose a revenue generation event occurs for an entity e at time t, and results in R amount of revenue being generated for the system. The mechanism has two parameters pertaining to incentives, β and s. The fraction of revenue to be distributed as incentive among the users is determined by $\beta \leq 1$. The parameter $s > 1$ controls the relative importance of tokens placed on an item depending on the order in which they were placed.

Let $\mathcal{T}$ be the set of all the tokens in the system. For a given token T_i, such that $e_i = e$, and a time period t we define $a_i(t)$ and $b_i(t)$ as follows (informally $a_i(t)$ is the weight of tokens on entity e_i which were placed before T_i and $b_i(t) - a_i(t)$ is the weight of token T_i at time t, note that $a_0(t) = 1$):

$$a_i(t) = \sum_{T_j \in \mathcal{T}: j<i, e_j=e} p_j w_j + 1,$$

$$b_i(t) = p_i w_i + a_i(t). \tag{1}$$

In case the above values fall below 1 for an entity, it is removed from the system for some pre-defined time. The revenue share of user u_i during time period t due to token T_i is given by

$$s\beta R \int_{a_i(t)}^{b_i(t)} \frac{1}{\tau^s} d\tau. \tag{2}$$

Note that the above quantity is positive or negative depending on p_i. We emphasize the following two properties: (1) the relative importance of the tokens placed earlier (discoveries of high quality entities) can be controlled by s, (2) the tokens placed after token T_i have no bearing on the incentives generated by T_i (contrast it with the case where s is allowed to be equal to 1). We note that the proposed mechanisms can be implemented in the existing systems where there are ways of giving explicit or implicit positive and negative feedback.

4.1.3 Properties of the System

In the following, we assume users behave rationally. In particular, we assume that if users see an arbitrage opportunity then the opportunity will be availed. Under this assumption, in our setting the desired properties listed in section 3 can be formalized as follows.

1. **Ranking by quality.** For every pair of entities (i, j) such that $q_i < q_j$ and $\tau_i > \tau_j$, there should exist a profitable arbitrage opportunity in the form of removing a token from entity i and placing it on entity j. We will now demonstrate that our mechanism satisfies the properties listed in the deseridata, when the revenue function $f(r, q)$ is a separable function (see section 2 for definition). Recall that this class contains the important function, $f(r, q) = rq$.

   Suppose there exists a pair of entities (i, j) such that $q_i < q_j$ and $\tau_i > \tau_j$, where τ_i and τ_j are the respective number of tokens taking into account the weights in equation 1. Let $f(r, q) = qr^\alpha$. Since $f(r, q)$ is an increasing function of q, $f(r_i, q_i)/\tau_i^s < f(r_i, q_j)/\tau_i^s$. We set the parameter s to an arbitrary value greater than α. Now $f(r_i, q_j)/\tau_i^s = q_j r_i^\alpha/\tau_i^s = \frac{q_j \tau_i^\alpha}{\tau_i^s (\sum_k \tau_k)^\alpha} < \frac{q_j \tau_j^\alpha}{\tau_j^s (\sum_k \tau_k)^\alpha} = f(r_j, q_j)/\tau_j^s$. The last inequality follows from the fact that $\tau_i > \tau_j$ and $s > \alpha$. Note that if $f(r_i, q_i)/\tau_i^s < f(r_j, q_j)/\tau_j^s$, then users can perform arbitrage by placing a negative token on entity i and a positive token on entity j. Hence the system ranks entities according to quality.

2. **Resistance to gaming.** In the setting of our mechanisms, the definition of resistance to gaming is identical to the definition of ranking by quality (the above mentioned inconsistency in the ranking can be a result of malicious behavior).

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. Let acc_i be the current account of user i. This value acc_i is the amount that the user i has generated as incentives from the past. 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 case the value of the account of a user goes negative, the feedback of the user is not taken into consideration for a predefined time (the older tokens placed by the user are removed by setting the w_i's of corresponding tokens to 0). Note because of 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 theireventually one of them would get bankrupt[1].

The system allows for addition of other features, for example, the tokens can be made to decay at a rate $d(t)$. The decay function ensures that a misstep of a user (that is, placing an erroneous negative token) is not recurrently penalized. Also, in many ranking and recommendation systems, we have greater leverage in controlling f. Suppose there are n entities that need to be recommended and we knew the number of eyeballs that an entity in the ith position would attract. Then the scheme we use to convert the r_i's to a (possibly probabilistic) ranking would decide the revenue function f. Since the choice of the scheme is in the hands of the designer, so is the revenue function f. In general, an appropriate model for users' response to a ranking system would help one to design better ranking schemes for that system.

4.2 Comparison to Information Markets

An alternative way of thinking about the problem is to characterize it as an information aggregation problem. Information markets have been successfully used for this purpose. So can we use information markets for webpage ranking? An information market approach can be implemented by floating shares of an entity and allowing users to trade them. (Note that this would require a separate system for trading and explicit participation of the users.) It is reasonable to assume that the part of revenue that one would share with the information market would only be a fraction of the actual revenue generated by the webpage. Hence, the owner of the webpage would

[1] A more explicit approach to avoid "racing" is to multiply any negative revenue (i.e. when p_i is negative in equation 1, resulting in a negative share from equation 2) 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.

have an incentive to buy all the shares, thus creating a thin market in which the owner by the act of hoarding the shares succeeds in taking away the arbitrage opportunities of other users, thus artificially increasing the price of the webpage. Market scoring rules [12] solve the thin market problem. Our scheme might be seen as an adaptation of market scoring rules. However, there are important features in our setting which makes the direct use of market scoring rules infeasible. Unlike other information aggregation problems, the outcome of a ranking system is not divorced from the machinations of the market. In fact here the market is not merely a predictor of an event but plays an indispensable role in content distribution. Miller et al. [20] counter the lack of objective outcomes by comparing a user's reviews to that of its peers. Their scheme gives users an incentive to provide honest feedback. However, their approach doesn't address malicious users and the discovery of good entities which haven't attracted much feedback yet. Also, in their model, the impact of reviews on the outcome (for example, on the revenue generated by the system) is not explicit.

5 Future Directions

As pointed out in section 4.1.3, an appropriate model for users' response to a ranking system would help in designing better systems. In our view, modeling users' response to ranking/recommendation systems in specific domains is an important direction for future work. Another direction is designing ranking schemes with r_i's as input. While the appropriateness of the models would have a strong bearing on the design and success of these schemes, it is also possible that there are ranking algorithms and revenue sharing schemes which can be shown to work for a generic class of models of user behavior. One example would be the class of models where the number of eyeballs that a position attracts is fixed but unknown and the probability that an eyeball is converted to a useful event is a function of the quality of the entity present there. The separable revenue functions studied in this paper are a step in that direction.

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

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

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

[7] 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.*

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

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

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

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

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

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

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

[15] http://www.netflix.com

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

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

[18] 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.*

[19] 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.*

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

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

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

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

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

[25] 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 A

In this appendix, we demonstrate in a simple setting, the need of incentives for a ranking/recommendation system to work properly. We emphasize that the problems we point to are not due to fraudulent clicks and hence cannot be fixed by better technology for detecting click spam. We show that an initial imbalance in the feedback would take exponential time to be corrected. Suppose there are two entities, e_1 and e_2, of the same quality q with e_1 having 1 token and e_2 having $\gamma > 1$ tokens. This difference can be due to various reasons. It can be a result of some targeted feedback by an interested party. 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, or articles on a certain stock might suddenly be sought after a surprise declaration of strong earnings. Also, the difference may be a result of the fact that the first entity is a new one in the system.

Since the quality of the two entities are the same, we can ignore the dependence of f on q_i. For the purpose of exposition, we assume that f depends linearly on r_i, that is, $f(r_i) = r_i = \frac{\tau_i}{\sum_j \tau_j}$. Similar results can be shown for other functions. Let the number of users in the system be $(1+\epsilon)n$ where there are n sheep and ϵn connoisseurs. Let τ_1 be the number of tokens on e_1 and τ_2 be the number of tokens on e_2. We normalize the number of users in the system and assume the ratio of sheep to connoisseur is $1 : \epsilon$. Now the rate at which sheep would put tokens on e_1 is given by $\frac{\tau_1}{\tau_1+\tau_2}$. Similarly the rate at which tokens are put on e_2 by sheep is given by $\frac{\tau_2}{\tau_1+\tau_2}$. Initially, $\tau_1 = 1$ and $\tau_2 = \gamma$. Since we are interested in proving a negative result, it doesn't harm us to assume that all the connoisseur weight ϵ is assigned to e_1. Hence,

$$
\begin{aligned}
\frac{d\tau_1}{dt} &= \epsilon + \frac{\tau_1}{\tau_1+\tau_2}, \frac{d\tau_2}{dt} = \frac{\tau_2}{\tau_1+\tau_2} \\
\frac{d\tau_1}{dt} + \frac{d\tau_2}{dt} &= 1+\epsilon \qquad \text{[summing]} \\
\tau_1+\tau_2 = 1+\gamma+(1+\epsilon)t & \qquad \text{[integrating]} \\
\frac{d\tau_2}{dt} &= \frac{\tau_2}{(1+\gamma)+(1+\epsilon)t} \\
\log\frac{\tau_2}{\gamma} &= \frac{1}{1+\epsilon}\log\frac{(1+\gamma)+(1+\epsilon)t}{1+\gamma} \\
\tau_2 &= \gamma\left(1+\frac{1+\epsilon}{1+\gamma}t\right)^{\frac{1}{1+\epsilon}}
\end{aligned}
$$

We are interested in the time when $\tau_1 = \tau_2$. Let the time when this equality is reached be T.

$$
\begin{aligned}
2\gamma\left(1+\frac{1+\epsilon}{1+\gamma}T\right)^{\frac{1}{1+\epsilon}} &= 1+\gamma+(1+\epsilon)T \\
T &= \left[\left(\frac{2\gamma}{1+\gamma}\right)^{1+1/\epsilon} - 1\right]\left(\frac{1+\gamma}{1+\epsilon}\right)
\end{aligned}
$$

For a large value of γ and $\epsilon = .01$, the time taken would be of the order of 2^{100} which is impractical. Note that even if we assume that we don't normalize and at every step the total change is of the order of n, this number would still be large for appropriate values of γ. It is easy to see that merely allowing negative feedback would make no qualitative difference in the above analysis.