Ranking and Suggesting Popular Items

Milan Vojnović, James Cruise, Dinan Gunawardena, and Peter Marbach

Abstract—We consider the problem of ranking the popularity of items and suggesting popular items based on user feedback. User feedback is obtained by iteratively presenting a set of suggested items, and users selecting items based on their own preferences either from this suggestion set or from the set of all possible items. The goal is to quickly learn the true popularity ranking of items (unbiased by the made suggestions), and suggest true popular items. The difficulty is that making suggestions to users can reinforce popularity of some items and distort the resulting item ranking. The described problem of ranking and suggesting items arises in diverse applications including search query suggestions and tag suggestions for social tagging systems. We propose and study several algorithms for ranking and suggesting popular items, provide analytical results on their performance, and present numerical results obtained using the inferred popularity of tags from a month-long crawl of a popular social bookmarking service. Our results suggest that lightweight, randomized update rules that require no special configuration parameters provide good performance.

Index Terms—Popularity ranking, recommendation, suggestion, implicit user feedback, search query, social tagging.

1 Introduction

We consider the problem of learning the popularity of items that is assumed to be a priori unknown but has to be learned from the observed user's selection of items. In particular, we consider systems where each user is presented with a list of items ("suggested items") and the user selects a set of preferred items that can contain either suggested items or any other items preferred by this user. The list of suggested items would typically contain only a (small) subset of popular items. The goal of the system is to efficiently learn the popularity of items and suggest popular items to users.

Items are suggested to users to facilitate tasks such as browsing or tagging of the content. Items could be search query keywords, files, documents, and any items selected by users from short lists of popular items. A specific application is that of tagging of the content where items are "tags" applied by users to content such as photos (e.g., Flickr), videos (e.g., YouTube), or Web pages (e.g., del.icio.us) for their later retrieval or personal information management. The basic premise of social tagging is that the user can choose any set of tags for an information object according to her *preference*. In most existing social tagging applications, users are presented with tag suggestions that are made based on the history of tag selections.[1] Fig. 1 shows an example user interface to enter tags for a Web page.

The learning of item popularity is complicated by the suggesting of items to users. Indeed, we expect that users would tend to select suggested items more frequently. This could be for various reasons, for example, (least effort) where users select suggested items as it is easier than thinking of alternatives that are not suggested or (bandwagon) where humans may tend to conform to choices of other users that are reflected in the suggestion set showing a few popular items. In practice, we find indications that such *popularity bias* may well happen; see, for example, Sen et al. [18] and Suchanek et al. [20]. In Fig. 3, we provide results of our own user study that indicate users' tendency to imitate.[2] One may ask, if suggesting popular items seems problematic due to potential popularity disorder, why make suggestions in the first place? This is for several reasons; for example, suggestions may help recall what candidate items are. A fix to avoid popularity skew would be to suggest all candidate items and not restrict to a short list of few popular items. This is often impractical for reasons such as limited user interface space, user ability to process smaller sets easier, and the irrelevance of less popular items. So, the number of suggestion items is limited to a small number (e.g., seven for the suggested tags in del.icio.us).

In this paper, our goal is to propose algorithms and analyze their performance for suggesting popular items to users in a way that enables learning of the users' *true preference* over items. The true preference refers to the preference over items that would be observed from the users' selections over items without exposure to any suggestions. A simple scheme for ranking and suggesting popular items (that appears in common use in practice)

- *M. Vojnović and D. Gunawardena are with Microsoft Research Ltd., 7 JJ Thomson Avenue, CB3 0FB Cambridge, UK. E-mail: {milanv, dinang}@microsoft.com.*
- *J. Cruise is with the Department of Mathematics, Bristol University, University Walk, Royal Fort Annex, Bristol BS8 1TW, UK. E-mail: marjrc@bristol.ac.uk.*
- *P. Marbach is with the Bahen Center for Information Technology (BCIT), University of Toronto, Room BA5232, Toronto, ON M5S 3G4, Canada. E-mail: marbach@cs.toronto.edu.*

Manuscript received 16 July 2008; revised 29 Nov. 2008; accepted 8 Dec. 2008; published online 16 Jan. 2009.
Recommended for acceptance by Y. Chen.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-07-0366.
Digital Object Identifier no. 10.1109/TKDE.2009.34.

1. We focus on the ranking of items where the only available information is the observed selection of items. In learning of the user's preference over items, one may leverage some side information about items, but this is out of the scope of this paper.

2. The user study was conducted with about 400 participants that tagged a set of six Web pages. For a Web page, the user was either presented or not presented tag suggestions which enabled us to measure the frequencies of tag selections in the presence and absence of suggestions.

Fig. 1. An example tag entry user interface. The user is presented with recommended tags and selects any preferred tags by typing them in the provided text box.

presents a fixed number of the most popular items as observed from the past item selections. We show analysis that suggests such a simple scheme can *lock down* to a set of items that are *not* the true most popular items if the popularity bias is sufficiently large, and may obscure learning the true preference over items. In this paper, we propose alternative algorithms designed to avoid such reinforcements and provide formal performance analysis of the ranking limit points and popularity of the suggested items. While convergence speed of the algorithms is of interest, its formal analysis is out of the scope of this paper.

In the remainder of the paper, we first define more precisely the problem that we consider (Section 2), provide a summary of our results (Section 3), and discuss related work (Section 4). We then introduce the algorithms that we study (Section 5) and present our main analytical results (Section 6) and numerical results (Section 7). The proofs of our main results are given in the Appendix.

2 Problem Formulation

In this section, we present a more precise description of the problem of ranking and suggesting items that we consider, and define a user's choice model that we use for our analysis.

2.1 Ranking and Suggesting Items

We consider the situation where users select items from a given set $C := \{1, 2, \dots, c\}$, where $c > 1$. We let

$$r = (r_1, r_2, \dots, r_c)$$

be the users' true preference over the set of items C and call r *the true popularity* rank scores. For an item i, we interpret r_i as the portion of users that would select item i if suggestions were not made. We assume that the true popularity rank scores r are such that: 1) r_i is *strictly* positive for each item i, 2) items are enumerated such that $r_1 \geq r_2 \geq \cdots \geq r_c$, and 3) r is normalized such that it is a probability distribution, i.e., $r_1 + r_2 + \cdots + r_c = 1$.

An algorithm is specified by: 1) *ranking rule*: the rule that specifies how to update the ranking scores of items that are denoted by $\rho = (\rho_1, \dots, \rho_c)$ and 2) *suggestion rule*: the rule that specifies what subset of items to suggest to a user. We assume that the size of the suggestion set is fixed to s, a positive integer that is a system configuration parameter. A notable difference with respect to ranking problems such as that of web search results is that our ranking algorithms do not account for the order in which items are suggested to the users—this is because we are interested in small suggestion sets and assume that we can allow for randomized presentation order.

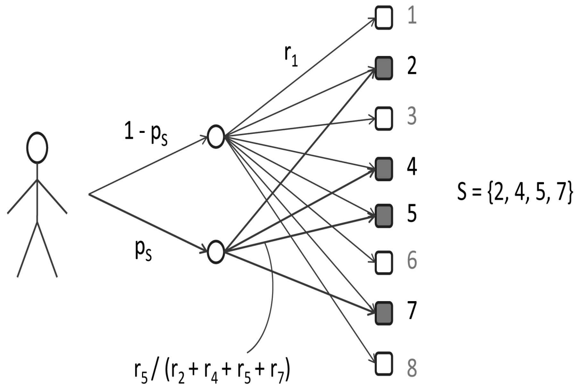

Fig. 2. User's choice model: an item is selected over a set of eight items with the set $S = \{2, 4, 5, 7\}$ of suggested items. With probability $1 - p_S$, item is selected by sampling from the distribution r over the entire set of items, else, the same but confined to the items in the set S.

The design objective to learn the true popularity ranking of items means that the ranking order induced by the ranking scores $\rho(t)$ at time t is the same as that induced by the true popularity ranking scores r, as the number of users t tends to be large. In other words, we want that for any two items i and j, $r_i \geq r_j$ implies $\rho_i(t) \geq \rho_j(t)$, for sufficiently large t. The design objectives are also to suggest true popular items and identify quickly the true popular items. Ideally, we would like the ranking order induced by $\rho(t)$ to conform to that induced by r after a small number of selections. We characterize the precision of a set of items S using the following metric. For a set S of size s,

$$\mathrm{prec}(S) = \frac{|\{i \in S : \ r_i \geq r_s\}|}{s}. \tag{1}$$

This is a standard information-retrieval measure of *precision* [17], [21] defining the relevant items to be $1, 2, \dots, v$, where $v = \max\{i \in C : \ r_i \leq r_s\}$. Note that (1) values the same any item i such that $r_i \geq r_s$. Alternatively, one may consider weighted precision that would value more items with larger frequencies, which we do not consider in this paper.

We derive some of our analytical results assuming a model for the user's choice of items described in the following section.

2.2 User's Choice Model

We use the following model for how users choose items. Suppose a user is presented a set S of suggested items. The user selects an item from the entire set of items by sampling, using the true item popularity distribution r, with probability $1 - p_S$. Otherwise, the user does the same but confines her choice to items in the suggestion set S. (See Fig. 2.) In other words, we have

$$\begin{aligned} &\Pr\{\text{selected item } = i | \text{ suggestion set} = S\} \\ &\quad = (1 - p_S) r_i + p_S \frac{r_i 1_{i \in S}}{\sum_{j \in S} r_j}. \end{aligned} \tag{2}$$

We admit this simple and intuitive model in order to facilitate analysis under a model of user's choice that biases

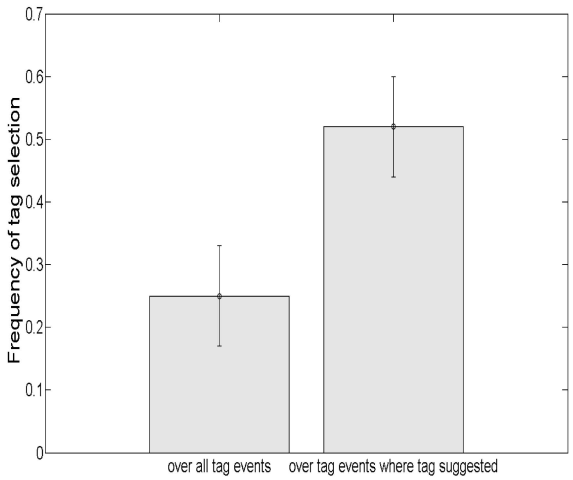

Fig. 3. Tag popularity boost due to suggestions. Frequency of selection for the tag "apollo" over all tag applications and over those at which "apollo" was suggested. The error bars correspond to 95 percent confidence intervals.

to items in the suggestion set. The model turns out to conform to the user's choice axioms introduced by Luce [13]; also related is Luce-Shepard model [14], [2]. The model accommodates two cases of interest:

Case 1. Consider a dichotomous user population where a fraction $1 - p$ of users sample an item from the distribution r over the entire set of items and the remaining fraction of users p *imitate* by sampling an item from their preference distribution r confined to the presented suggestion set. We then have (2) with $p_S \equiv p$.

Case 2. Suppose that suggesting an item boosts its probability of selection in the following way. Each user selects an item i with probability proportional to αr_i, where $\alpha > 1$ if item i is suggested and $\alpha = 1$ if item i is not suggested. The boost of items presented in the suggestion set is thus by a fixed multiplicative factor greater than 1. We then have that (2) holds with the imitation probability $p_S := [(\alpha - 1)r_S]/[1 + (\alpha - 1)r_S]$, where $r_S := \sum_{j \in S} r_j$.

Note that given the suggestion set of items, the user's item selection is stochastically independent of the past item selections. This may not hold if we consider items selected by a given user, but it would hold for items selected by distinct users.

3 Summary of Results

We propose and analyze three randomized update rules for suggesting popular items based on the history of item selection.

First, we consider the naive algorithm which suggests a fixed number of the topmost popular items and show that this algorithm can fail to learn the true popularity ranking of items if the imitation probability in the user's choice model is sufficiently large. In particular, we find that there exists a threshold on the imitation probability below which the algorithm guarantees to learn the true popularity ranking of items, and otherwise, this may not hold. We fully specify this threshold in terms of the suggestion set size and the true popularity ranks of items. This result enables us to estimate the threshold imitation probability for a given true popularity ranking. In particular, we provide estimates for the threshold using our data set of tags applied to popular Web pages in del.icio.us and found threshold to be typically around 0.1 for the suggestion set sizes ranging from 1 to 10 tags. This suggests that in real-world scenarios, using the above simple scheme may result in failing to learn the true popularity of items at small imitation rates.

Our first randomized algorithm suggests to each user a random set of items S sampled with the probability proportional to the *sum* of the item popularity rank scores. We call this algorithm PROP (frequency proportional sampling). Such a sampling appears a natural randomization that one would consider in order to mitigate the popularity ranking skew; biasing to popular items, but still letting each item appear recurrently in the suggestion set. Again, we show that this algorithm guarantees to learn the true popularity ranking only if the imitation probability is smaller than a threshold and fully specify this threshold. Another issue with PROP is that sampling proportional to the sum of rank scores appears computationally nontrivial. For these reasons, we propose two other randomized schemes. One of the randomized schemes is M2S ("move-to-set") that recursively updates the suggestion set based on the items selected by users. The algorithm biases to show recently used items and in the special case of the suggestion size equal to one is equivalent to showing the last used item. It is noteworthy that M2S does not require any counters for updating the suggestion set. We show that the update rule of M2S results in suggesting a set of items proportional to the *product* of the popularity rank scores. We prove that under our user's choice model, for any imitation probability smaller than 1, M2S guarantees that the frequencies of item selections induce a popularity ranking that corresponds to that of the true popularity. One feature of M2S is that any item that is recurrently selected by users appears in the suggestion set with same rate, increasing with the popularity of this item. Our last randomized algorithm, FM2S ("frequency move-to-set"), is designed to restrict suggesting only items that are sufficiently popular. At a high level, FM2S replaces an item in the suggestion set with a new item only if this new item is (likely to be) more popular than at least one item already in the suggestion set. The basic idea is to mitigate the popularity bias by keeping auxiliary scores per item that are incremented for items that are selected, but only if not suggested. We show that FM2S tends to display only a subset of sufficiently popular items with respect to their true popularity and fully determine this set of items in terms of the suggestion set size and the true popularity rank scores of items. The true popularity ranking of these items can be inferred from their frequency of appearance in the suggestion set. The algorithm can be seen as a relaxation of the TOP scheme that avoids locking down to suggesting a set of items that are not the true most popular.

In summary, M2S and FM2S are randomized algorithms for suggesting popular items and learning their true popularity ranking that are *lightweight* with respect to their computational and storage requirements. The algorithms are *self-tuning* in that they do not require any special configuration parameters, except the size of the suggestion set. The main distinguishing feature of FM2S is that it confines to displaying only sufficiently popular items. Another interesting property for some applications is that it allows a larger set of most popular items than the suggestion set to be separated out. For example, this may be of interest for applications where the size of the suggestion set is limited but the display of a larger set of popular is required.

Finally, we present numerical results obtained by evaluating our analytical results using the popularity rank scores of tags for bookmarks inferred from a month-long crawl of the social bookmarking application del.icio.us.

4 Related Work

The problem studied in this paper relates to the broad area of recommendation systems (e.g., Kumar et al. [10] and Sandler and Kleinberg [9]) in which the goal is to learn which items are preferred by users based on the user's selection of items. The important distinction of our problem is in that we consider a system with *feedback*, established through suggestions. This may results in positive reinforcements similar in spirit to those of well-known preferential attachment [3], [5]. Another related area is that of voting systems. Specifically, our system could formally be seen as an instance of *approval voting* [8], [4] in that each user can select any set of candidates offered on a voting ballot. Again, intrinsic to our problem is the continual feedback that indicates popular candidates. In the voting systems literature, it is well recognized that such feedback may bias the voting results; see, e.g., Simon [19] for an analysis of the bandwagon and underdog effects induced by preelection polls. Our work is related to statistical learning problems of the multiarmed bandit type (e.g., Lai and Robbins [11]), described as follows: We consider a finite set of items. Each user is presented with an item that is selected by this user with (unknown) probability specific to this item. The goal is to present items to the users such that the expected cumulative number of item selections is maximized. An asymptotically optimal rule to decide which item to present was found by Lai and Robbins [11] and was further extended by Anantharam et al. [1] to allow presenting more than one item. Further related work includes that of using the click-through data to enhance the Web search results. Pandey et al. [15] and Cho et al. [6] studied the *entrenchment problem* where the search engine result sets lock down to a set of popular URLs and proposed to intervene the results with randomly sampled URLs. An important distinction of our problem to that studied in the aforementioned prior work is that selection of items is not restricted to items presented to the user. Finally, we discuss the research on social tagging as our numerical results are provided specifically for this context. In [7], Golder and Huberman provide various statistical characterization results on the tagging in the social bookmarking application del.icio.us. Sen et al. [18] studied the effect of the tag suggestions on the users' choice of tags in MovieLens systems, which they instrumented for experiments. Their results suggest that tags applied by users are affected by tag suggestions. Xu et al. [23] proposed a system for recommended tags but their work differs from the goal of this paper. Finally, we refer to Suchanek at al. [20] for an estimation procedure of the imitation rate defined in this paper and estimates for tagging of Web pages scenario.

5 Algorithms

We provide a precise definition of the algorithms for ranking and suggesting items that we consider in this section.

5.1 A Naive Algorithm

We first introduce the simple algorithm TOP which consists of a ranking and a suggestion rule as defined below.

TOP (Top popular)
Init: $V_i = 0$ *for each item* i
At the t*-th item selection:*
  If *item* i *selected:*
    $V_i \leftarrow V_i + 1$
  $S \leftarrow$ *a set of* s *items with largest* V *counts*

The ranking rule is to set the rank score of an item equal to the number of selections of this item in the past. For this algorithm and the algorithms introduced later, we initialize $V_i = 0$ for each item i. The implicit assumption is that we assume no prior information about the popularity of items, and hence, initially assume that all items are equally popular.[3] The suggestion rule sets the suggestion set to a set of the top s most popular items with respect to the current popularity rank scores.

We will later identify cases when this simple algorithm can get locked down to a ranking ρ that induces a different ranking than that induced by the true popularity ranking r, and thus, may fail to learn the true popularity of items. To overcome this problem, we consider in the following alternative ranking and suggestion rules.

5.2 Ranking Rules

In this section, we define two ranking rules called rank rule 1 and rank rule 2.

Rank rule 1. A simple ranking rule is the one that we already encountered in the algorithm TOP, where the rank score for an item i is incremented by 1 whenever a user selects this item.

Init: $V_i = 0$ *for each item* i
At the t*-th item selection:*
 If *item* i *selected:*
  $V_i \leftarrow V_i + 1$
  $\rho_i \leftarrow V_i / t$

We will see that this ranking may fail to discover the ranking order of the true popularity when combined with a suggestion rule that reinforces items that were selected early on, as it is the case under TOP.

Rank rule 2. We noted that rank rule 1 may fail to discover the ranking order of the true popularity if used with suggestion rules such as TOP. To overcome this problem, we may redefine the rank scores in the following way.

Init: $T_i = 0$, $V_i = 0$, *for each item* i
At the t*-th item selection:*
 For each item i:
  If *item* i ***not*** *suggested:*
   $T_i \leftarrow T_i + 1$
   If *item* i *selected:*
    $V_i \leftarrow V_i + 1$
   $\rho_i \leftarrow V_i / T_i$.

3. In practice, one may use prior information about item popularity. For example, in social bookmarking applications, information from the keywords metadata and content of Web pages can be used to bias the initial tag popularity.

Now, the rank scores ρ are updated only for an item that was selected and was not suggested to the user. The ranking score ρ_i for an item i can be interpreted as the rate at which item i is selected over selections for which item i was not suggested. We have the following result:

Lemma 1. *Consider any suggestion rule combined with the rank rule 2 under the only assumption that each item exits the suggestion set infinitely often. Then, under the user's choice model:* $\lim_{t\to+\infty} \rho(t) = r$.

The lemma follows by noting that under the user's choice model, an item i that is not suggested is selected with probability proportional to r_i. The result tells us that under the user's choice model, rank rule 2 combined with a suggestion rule from broad set, guarantees to learn the true popularity ranking of items. The only assumption is that suggestion rule is such that each item exits the suggestion set with a probability that is lower bounded by a positive (but possibly very small) constant. However, rank rule 2 might have slow rate of convergence as the rank scores are updated only over a subsequence of item selections that were not suggested. For this reason, we will focus on rank rule 1 combined with different suggestion rules and consider robustness in learning the true popularity ranking.

5.3 Suggestion Rules

We introduce three different suggestion rules: 1) Frequency Proportional, 2) Move-to-Set, and 3) Frequency Move-to-Set.

`PROP` is a randomized algorithm that for each user presents a suggestion set of items, sampled with probability proportional to the sum of the current rank scores of items. The algorithm is described below in more detail.

> `PROP` (FREQUENCY PROPORTIONAL)
> *At the t-th item selection:*
> *Sample a set S of s items with probability*
> $\propto \sum_{j\in S} \rho_j$

We will later show analysis that this suggestion rule combined with rank rule 1 is more robust to imitation than `TOP`, but there still may exist cases when it fails to learn the true popularity of items. Note also that the algorithm is computationally demanding when the number of items c and suggestion set size s are nonsmall; it requires sampling on a set of $\binom{c}{s}$ elements. We do not examine how this sampling could be efficiently done. Our next algorithm is computationally very simple.

> `M2S` (MOVE-TO-SET)
> *At the t-th item selection with item i selected:*
> ***If** item i **not** in the suggestion set S*
> *Remove a random item from S*
> *Add i to S*

`M2S` is a random iterative update rule of the suggestion set of items. The suggestion set is updated only when a user selects an item that is not in the suggestion set presented to the user. Note that for the suggestion set size of one item, `M2S` suggests *the last used item*, a recommendation rule used by many user interface designs. For the suggestion set size greater than one item, `M2S` is different from suggesting the last distinct used items due to the random eviction of items from the suggestion set, but note that the rule does bias to presenting *recently used items*. We will show how exactly this update rule tends to bias the sampling of the suggestion set with respect to true popularity rank scores of items. As an aside, note that `M2S` relates to the self-organized sorting of items known as move-to-front heuristic (e.g., [16]).

It follows from the description of the suggestion rule `M2S` that any item would recurrently appear in the suggestion set, provided only that it is recurrently selected by users with some positive probability (no matter how small). Our next algorithm is similar in spirit to `M2S`, but is designed so that an item appears recurrently in the suggestion set only if sufficiently popular, with respect to its true popularity. This lockdown feature characterizes the simple algorithm `TOP`, but note that our goal is to ensure that the lockdown is to a subset of true top popular items. We call this new algorithm `FM2S` (frequency move to set) for the reasons that we discuss shortly; the algorithm is defined as follows.

> `FM2S` (FREQUENCY MOVE-TO-SET)
> *Init: $W_i = 0$ for each item i*
> *At the t-th item selection with item i selected:*
> ***If** item i **not** in the suggestion set S*
> $W_i \leftarrow W_i + 1$
> $E = \{j \in S : W_j < W_i\}$
> ***If** E is nonempty*
> *Remove a random item from S that is in E*
> *Add i to S*

For each item, the algorithm keeps a counter of how many users selected this item over users that were not suggested this item. The rationale is not to update the counter for items that were suggested and selected by users in order to mitigate the positive reinforcement due to exposure in the suggestion set. Furthermore, a selected item that was not suggested does not immediately qualify for entry in the suggestion set (as with `M2S`), but only if its counter exceeds that of an item that is already in the suggestion set. In addition, specific to `FM2S` is that the eviction of an item from the suggestion set is over a subset of items with smallest counter. We will see how this additional update rules yield higher precision of the suggestion set in our numerical examples (Section 7).

6 ANALYTICAL RESULTS

In this section, we present our main analytical results on the ranking and suggesting algorithms introduced in the preceding section.

6.1 Top Popular

We analyze robustness of the algorithm `TOP` to the user imitation. We will see that there exists a threshold on the imitation probability below which the algorithm guarantees to learn the true popularity rank of items and above or at this threshold, the undesired lockdown can happen where the learned ranking of items does not conform to that of the true popularity ranking. The threshold is a function of the suggestion set size s and the true popularity rank scores r that is made explicit in the result below.

Theorem 1. *Consider the algorithm* `TOP` *under the user's choice model with arbitrarily fixed tie break of item ranks.*

1. ***Top-popular sets***. *Suppose the user imitation probability is $p < 1$. Given r and s, the top popular rankings are induced by the ranking scores given by*

$$\rho_i(S) = \begin{cases} (1-p)r_i + p\frac{r_i}{\sum_{j\in S} r_j}, & i \in S, \\ (1-p)r_i, & i \in C \setminus S, \end{cases} \tag{3}$$

where S is a set of s items that satisfies

$$\left(\sum_{j\in S} r_j\right)\left(\frac{\max_{j\in C\setminus S} r_j}{\min_{j\in S} r_j} - 1\right) \leq \frac{p}{1-p}. \tag{4}$$

2. ***Uniqueness***. *The set $\{1, 2, \ldots, s\}$ is a unique top popular set if and only if the imitation probability p is such that*

$$p < p_{\text{crit}}(r, s),$$

where $p_{\text{crit}}(r, s)$ is the threshold imitation given by

$$p_{\text{crit}}(r, s) = \min_{0\leq i<s<j\leq c} \frac{a(i,j)}{1 + a(i,j)},$$

with

$$a(i,j) = \left(\sum_{k=1}^{i} r_k + \sum_{k=j-s+i+1}^{j} r_k\right)\left(\frac{r_{i+1}}{r_j} - 1\right).$$

Proof. Proof is provided in the Appendix. □

Equations (3) and (4) follow from our user choice model and the definition of the top popular suggestion rule—(4) is equivalent to saying that $\min_{j\in S}\rho_j \geq \max_{j\in C\setminus S}\rho_j$. Item 2 tells us that the true popularity ranking of items (with fixed tie break) is unique if and only if the imitation probability is smaller than the threshold asserted under item 2. This threshold depends on the true popularity ranking scores r and the size of the suggestion set s. This result is obtained by finding the values of the imitation probability p for which the left-hand side in (4) is strictly greater than $p/(1-p)$ over all sets S of items other than the set S^*. We will use the above result in Section 7 to evaluate the threshold imitation probabilities for the true popularity ranks r inferred from our data set.

6.2 Frequency Proportional

We now consider the suggestion rule `PROP` combined with rank rule 1. In particular, we focus on characterizing its robustness to imitation. We derive exact characterization of the threshold imitation probability below which the ranking scores $\rho(t)$ converge to the true popularity ranking scores r. When this condition holds, we obtain a closed-form expression for the frequencies with which items are suggested to users and the average precision of the suggested set.

Theorem 2. *The limit state of the suggestion rule* `PROP` *combined with rank rule 1 is characterized as follows.*

1. ***Limit ranking***. *Under the user's choice model, from any initial ranking scores $\rho(0)$, the ranking scores $\rho(t)$ converge to the true ranking scores r if and only if the imitation probability satisfies*

$$p < \frac{1}{\sigma(A)}, \tag{5}$$

where $\sigma(A)$ is the spectral radius of the matrix $A = (a_{ij})$ defined by

$$a_{ij} = r_i \frac{1}{\binom{c-1}{s-1}} \sum_{S'\in S_s:\ \{i,j\}\subseteq S'} \frac{1}{\sum_{k\in S'} r_k}, \tag{6}$$

for $i, j \in C$, where S_s contains all subsets of s distinct items from C.

2. ***Suggestion set***. *Under condition (5), the limit frequency at which an item i is suggested to users is*

$$s_i = r_i + \frac{s-1}{c-1}(1 - r_i).$$

The average precision of the suggestion set (1) is

$$\mathbb{E}(\text{prec}(S)) = \frac{v}{s}\left(\bar{r}_v + \frac{s-1}{c-1}(1 - \bar{r}_v)\right),$$

where $\bar{r}_v := (\sum_{i=1}^{v} r_i)/v$.

Proof. Proof is provided in the Appendix. □

The result in (5) gives an exact condition under which for given true popularity ranking scores r and the suggestion set size s, the algorithm guarantees to learn the true popularity ranking scores r. For this to hold, the imitation probability must be smaller than the threshold asserted in (5). The following corollary gives a sufficient condition.

Corollary 1. *A sufficient condition for the relation in (5) to hold is*

$$p < \frac{1}{s} + \left(1 - \frac{1}{s}\right)\frac{\left(\sum_{i=c-s+1}^{c} r_i\right)/(s-1)}{r_1}.$$

A stronger but simpler sufficient condition is $p < 1/s$.

Note that for the suggestion set size of one item ($s = 1$), the algorithm learns the true popularity rank r for any imitation probability $p < 1$. Note further that in this case, the limit frequency at which an item i is suggested is equal to r_i. We will see that the same property holds with the next algorithm that we consider.

6.3 Move to Set

We show that under the suggestion set update rule `M2S` starting from any initial suggestion set of items, the probability distribution of the suggestion set converges to a unique limit distribution that is characterized in terms of the true popularity distribution r and the suggestion set size s.

Theorem 3. *The limit state of the suggestion rule* `M2S` *as the number of items tends to be large is characterized as follows:*

1. ***Suggestion set***. *From any initial suggestion set $S(0)$, the probability distribution of the suggestion set $S(t)$ as the number of item selections t tends to infinity converges to*

$$\pi(S') \propto \prod_{i \in S'} r_i, \; S' \in S_s, \tag{7}$$

where the set S_s contains all subsets of s distinct items from the set of all items C.

2. ***Frequencies at which items are suggested.*** *The frequency at which an item i is suggested, s_i, has the following properties: 1) the larger the item true popularity r_i, the larger the frequency s_i and 2) the frequency s_i is sublinear in r_i, i.e., $s_i/r_i \leq s_j/r_j$, for any items i and j such that $r_i \geq r_j$.*
3. ***Combination with rank rule 1.*** *For any imitation probability $p < 1$, the limit ranking scores ρ induce the true popularity ranking, i.e.,*

$$r_i \geq r_j \Rightarrow \rho_i(t) \geq \rho_j(t), \text{for sufficiently large } t.$$

Proof. Proof is available in the Appendix. It relies on Theorem 4, which is of independent interest and shown later in this section. □

Item 1 tells us that in the limit (steady state), the algorithm samples a suggestion set S with the probability proportional to the product of the true popularity of the items in the set S. Hence, at a high level, the `M2S` update rule is similar to `PROP`, one difference being that `PROP` tends to the sampling proportional to the *sum* of the true item popularity rank scores (provided that the stability condition in Theorem 2 holds). Another way to look at the suggestion rule `M2S` is as a construction of a Markov chain that has the stationary distribution such that the probability of the suggestion set is proportional to the product of the true popularity rank scores of the items in the set. The frequency at which an item appears in the suggestion set is given by item 1 upon summing over the suggestion sets that containing the given item. We were unable to obtain a closed-form expression for this frequency in general for any $s \geq 1$. In the absence of a closed-form formula, item 2 provides properties about the frequency at which an item is suggested. Item 3 shows that the following interesting property holds: combining rank rule 1 with `M2S` guarantees learning the true popularity ranking for any imitation probability less than 1. The result follows using the following theorem which is of general interest—it provides a sufficient condition on the stationary distribution of the suggestion set and the imitation parameter for the rank rule 1 to induce the true ranking. We denote with $S_{i,j}$ the set containing all subsets of the set $C \setminus \{i, j\}$ of size $s-1$.

Theorem 4. *Consider a suggestion rule Σ under the user's choice model with parameters r and $(p_{S'}, S' \in S_s)$. Assume that the suggestion set under Σ has the limit distribution π. If for any i and j such that $r_i \geq r_j$, the following monotonicity conditions hold:*

1. *$\pi(A \cup \{i\}) \geq \pi(A \cup \{j\})$, for any $A \in S_{ij}$ and*
2. *$p_{A\cup\{i\}} \geq p_{A\cup\{j\}}$, for any $A \in S_{ij}$,*

then, the limit ranking of the rank rule 1 is the same as that induced by r.

Condition A says that replacing an item from a suggestion set with an item that is equally or more popular does not decrease the probability of the suggestion set. Condition B is an analogous statement for the imitation rate.

The result of the theorem may prove useful for design of new suggestion rules.

Proof. Suppose $r_i \geq r_j$. $\rho_i(+\infty) \geq \rho_j(\infty) \Leftrightarrow$

$$\begin{aligned}(1-\bar{p})r_i &+ \sum_{S' \in S_s} p_{S'} \frac{r_i 1_{i \in S'}}{\sum_{k \in S'} r_k} \pi(S') \\ &\geq (1-\bar{p})r_j + \sum_{S' \in S_s} p_{S'} \frac{r_j 1_{j \in S'}}{\sum_{k \in S} r_k} \pi(S'),\end{aligned}$$

where $\bar{p} = \sum_{S' \in S_s} p_{S'} \pi(S')$. The last inequality is implied by

$$\sum_{S' \in S_s} p_{S'} \frac{r_i 1_{i \in S'}}{\sum_{k \in S'} r_k} \pi(S') \geq \sum_{S' \in S_s} p_S \frac{r_j 1_{j \in S'}}{\sum_{k \in S'} r_k} \pi(S'),$$

which is further implied by

$$\begin{aligned}&\sum_{S' \in S_s} p_{S'} \frac{r_i}{\sum_{k \in S'} r_k} \pi(S') 1_{i \in S', j \notin S'} \\ &\quad \geq \sum_{S' \in S_s} p_{S'} \frac{r_j}{\sum_{k \in S'} r_k} \pi(S') 1_{i \notin S', j \in S'}.\end{aligned}$$

The last inequality can be rewritten as

$$\begin{aligned}&\sum_{A \in S_{ij}} p_{A\cup\{i\}} \frac{r_i}{r_i + \sum_{k \in A} r_k} \pi(A \cup \{i\}) \\ &\quad \geq \sum_{A \in S_{ij}} p_{A\cup\{j\}} \frac{r_j}{r_j + \sum_{k \in A} r_k} \pi(A \cup \{j\}).\end{aligned} \tag{8}$$

Now, $r_i/(r_i + \sum_{k \in A} r_k) \geq r_j/(r_j + \sum_{k \in A} r_k)$ indeed holds; thus, inequality (8) is true provided that

$$p_{A\cup\{i\}} \pi(A \cup \{i\}) \geq p_{A\cup\{j\}} \pi(A \cup \{j\}), \text{ all } A \in S_{ij},$$

which holds if both A and B hold. □

6.4 Frequency Move to Set

In this section, we examine the suggestion rule `FM2S`. We show that this algorithm tends to suggest only a subset of sufficiently true popular items ("competing set") and precisely characterize the competing set of items.

Theorem 5. *The equilibrium points of `FM2S` are characterized as follows:*

1. ***Competing set of items.*** *Assume that the state of the algorithm `FM2S` has a stationary regime. In the stationary regime, only a subset of items $\{1, 2, \ldots, c'\}$ are suggested with strictly positive probability, where c' is the largest integer i such that $s \leq i \leq c$ and*

$$r_i > \left(1 - \frac{s}{i}\right) h_i(r), \tag{9}$$

with $h_i(r)$ the harmonic mean of $r_1, \ldots, r_i$, i.e.,

$$h_i(r) := \frac{1}{\frac{1}{i}\left(\frac{1}{r_1} + \cdots + \frac{1}{r_i}\right)}.$$

2. ***Frequency at which an item i is suggested.***

$$s_i = \begin{cases} 1 - \left(1 - \frac{s}{c'}\right) \frac{h_{c'}(r)}{r_i}, & i = 1, 2, \ldots, c', \\ 0, & i = c'+1, \ldots, c. \end{cases} \tag{10}$$

3. ***Average precision of the suggestion set.***

$$\mathbb{E}(\mathrm{prec}(S)) = \frac{v}{s}\left(1 - \left(1 - \frac{s}{c'}\right)\frac{h_{c'}(r)}{h_v(r)}\right),$$

where c' is the size of the competing set defined in item 1 and v is from the definition of the precision (1).

Proof. Proof is available in the Appendix. □

Item 1 shows that an item i can appear in the suggested set with a strictly positive probability only if its true popularity r_i is larger than the asserted threshold. This threshold is defined per item and is given by the right-hand side of (9), which depends on the true popularity r and the suggestion set size s. This is an interesting result showing that the algorithm eventually locks down to suggesting items only from a subset of items. We have already encountered that the same property holds under TOP. Note that in FM2S, the competing set of items is such that each item in this set is more popular than any other item that is not in this set, where the popularity is with respect to the true popularity. This property does not hold for TOP as we showed that it can lockdown to a set of items that contains a less preferred item than an item that is not in the set. From item 1 of Theorem 5, it follows that the size of the competing set of items depends on the suggestion size in the following way:

Corollary 2. *For any given distribution of the true popularity scores r, the size of the competing set $c'(s)$ is nondecreasing with the suggestion set size s.*

The resolution of avoiding the lockdown to an inappropriate set using FM2S comes at the price of allowing some imprecision of the suggested set. Items 2 and 3 characterize the popularity of the suggested items in terms of the suggestion set size and the true popularity rank scores. The result under item 2 tells us that for an item i, the mean number of suggestion sets between successive suggestion sets *not* containing the item i is equal to $1/(1-s_i) = r_i/[(1-s/c')h_{c'}(r)]$, thus, proportional to the true popularity score of the item i. Combining with (9), we have the following.

Corollary 3. *For each item i that is in the competing set, $\frac{1}{1-s_i} \geq \frac{r_i}{r_{c'}}$.*

Hence, the mean number of suggestion set presentations between successive presentations of suggestion sets not containing a competing item i is at least the ratio of the item i rank score and the rank score of a least popular competing item. Lastly, we note the following worst-case bound on the average precision of the suggestion set size under FM2S.

Corollary 4. *We have $\mathbb{E}(\mathrm{prec}(S)) \geq s/c$, for all r.*

This gives a lower bound that holds uniformly over r and this bound is tight, i.e., for any c and $s \leq c$, there exists r that achieves the equality. Indeed, if $s = c$, then $\mathbb{E}(\mathrm{prec}(S)) = 1$ holds. For the case $s < c$, take $r_i = 1/c + \epsilon$ for $i = 1, 2, \ldots, s$ and $r_i = 1/c - s/(c-s)\epsilon$ for $i = s+1, \ldots, c$ and let $\epsilon \downarrow 0$.

Note that we have not obtained analogous result to item 3 of Theorem 3 when rank rule 1 combined with FM2S converges to the true popularity ranking, but note that the popularity ranking of the competing items can be inferred from the frequencies of item suggestions (item 2 of Theorem 5) that conforms to the true popularity ranking.

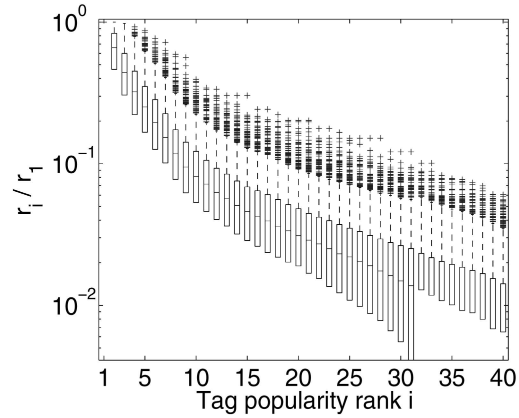

Fig. 4. The tag rank scores inferred from our del.icio.us sample that we use as true popularity rank scores. The graph shows the median ratio of the rank i score and rank 1 score of samples obtained for each bookmark in our data set.

7 Numerical Results

In this section, our goal is to evaluate performance of the algorithms that we analyzed in earlier sections using samples of real-world distributions for item popularity scores. The item popularity scores are inferred from the tagging histories of a set of bookmarks that we sampled from del.icio.us.

Data set. Our data set contains entire tagging histories for about 1,200 distinct bookmarks from the popular social bookmarking Web service del.icio.us. We sampled this set of bookmarks by collecting them from the *popular* del.icio.us Web page, sampled approximately each 15 minutes over a month period (27 October to 6 December 2006). For each page in our data set, we estimate the true popularity distribution of tags r and use these distributions to evaluate the expressions derived in our analysis. Fig. 4 summarizes the distributions r per bookmark by showing the boxplots for the ratio r_i/r_1 for tag ranks 1-40. We observe that the median of relative popularity rank score, r_i/r_1, versus rank i closely follows exponential decay with exponent about $5/4$, for ranks $i \leq 9$, and otherwise, another exponential decay but with a smaller exponent of about $1/3$.[4] In particular, the results suggest that a suggestion set to the top seven tags on average covers tags with true popularity at least 15 percent of the rank 1 tag popularity. This provides a justification for limiting the suggestion set to seven tags.

Learning the true popularity ranking. We first evaluate the threshold imitation probability for the algorithm TOP, as established in Theorem 1. Recall that if the imitation probability is larger than this threshold, TOP can lockdown to a set of items that are not top true popular. We have computed the threshold imitation probability for each bookmark in our data set. In Fig. 5, we show the empirical cumulative distribution function of the computed threshold imitation probabilities for a range of suggestion set sizes

4. This dichotomy may be a result of the ranking/suggesting rule in del.icio.us and the suggestion set size limited to seven tags therein.

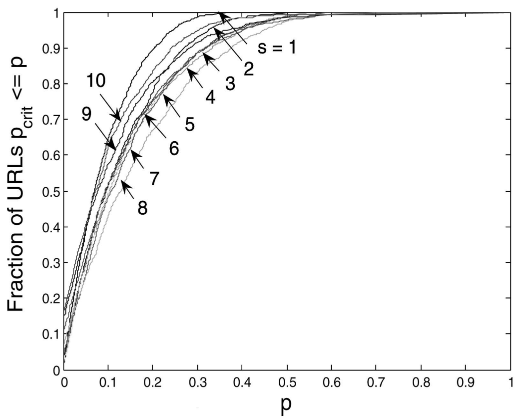

Fig. 5. TOP: Critical imitation for the suggestion set size s ranging from 1 to 10 beyond which the algorithm can fail to learn the top k tags with $k = s$. The median values of the critical imitation are around 0.1.

from 1 to 10 items. The median threshold imitation probability is around 0.1 across all the suggestion set sizes, suggesting that already at small level of imitation, the undesired lockdown may happen. We also examined the threshold imitation beyond which TOP cannot guarantee learning the true popularity ranking of top k true popular tags. We refer the interested reader for these results to Appendix [22] (Fig. 4). In summary, results suggest that TOP may result in failing to learn the true popularity ranking, for small values of the imitation probability.

Precision of suggestions. We next evaluate the mean precision of the suggestions made by algorithms PROP, M2S, and FM2S. In Fig. 9, we show the respective mean precisions of the suggested items for PROP, M2S, and FM2S for a range of suggestion set size from 1 to 40 tags. We observe that the mean precision under FM2S is better than that under either PROP or M2S. The mean precision under M2S is better than under PROP, and this is particularly emphasized for small suggestion set sizes (the range of practical relevance). The mean precision under FM2S is at least 80 p for suggestion set sizes greater or equal to 7, which may be regarded a good performance.

We next examine the suggestion rule FM2S in more detail. We first examine the frequency at which a tag is

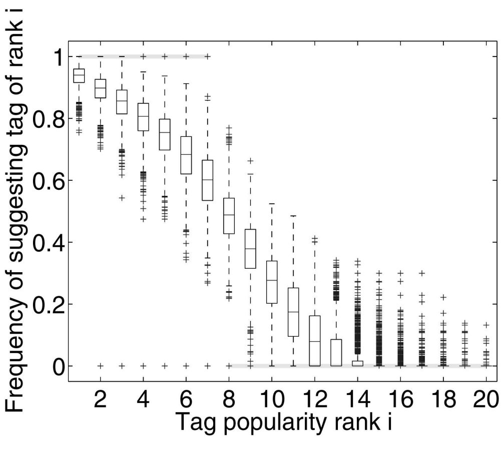

Fig. 6. The frequencies of tag suggestions under FM2S, with the suggestion set size $s = 7$, follow a "smoothed version" of the "step function."

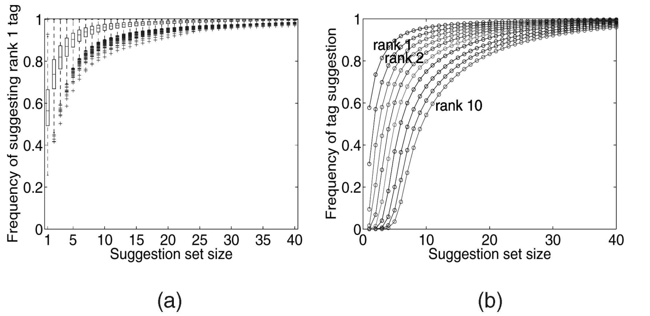

Fig. 7. FM2S. (a) The frequency at which rank 1 tag is suggested versus suggestion set size. (b) Same as in left but showing mean frequencies at which tags rank 1-10 are suggested.

suggested versus the true popularity rank of this tag; see Fig. 6 for results obtained with the suggestion set size set to seven tags. This illustrates the main properties of the suggestion rule FM2S: locking down to displaying a set of items with high true popularity, the gradual decrease of the frequency with which an item is suggested with its true popularity rank. In Fig. 9a, we have examined the average precision of the suggestion set under FM2S. We now examine the frequencies with which individual tags appear in the suggestion set versus their true popularity ranks and across a range of suggestion set sizes. The results are shown in Fig. 7. In particular, we note that on average, the rank 1 tag of a bookmark appears in more than 90 percent of suggestion sets, indicating good performance. In Theorem 5, we have determined how many items eventually get into the suggestion set ("competing set") with strictly positive probability as a function of the suggestion set size and the true popularity rank scores. How does the size of the competing set for the true popularity ranks in our data set across differ with suggestion set sizes? Fig. 8 shows the ratio of the competing set size and the suggestion set size for suggestion set sizes ranging from 1 to 40 tags. We observe that the competing set size is about twice or less than the suggestion set size, for suggestion set sizes ≥ 5 tags. This suggests that the competing set tends to increase proportionally with the suggestion set size.

Convergence speed. We have evaluated speed of convergence of the algorithms considered in this paper by simulations. For space reasons, we omit to present these

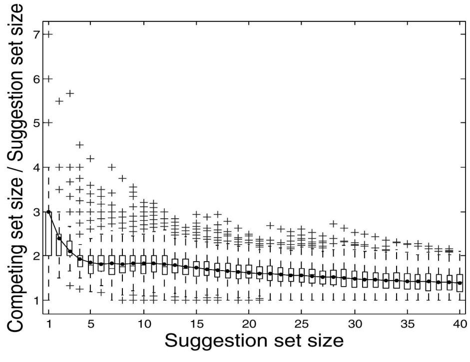

Fig. 8. The number of competing tags versus the suggestion set size for the algorithm FM2S.

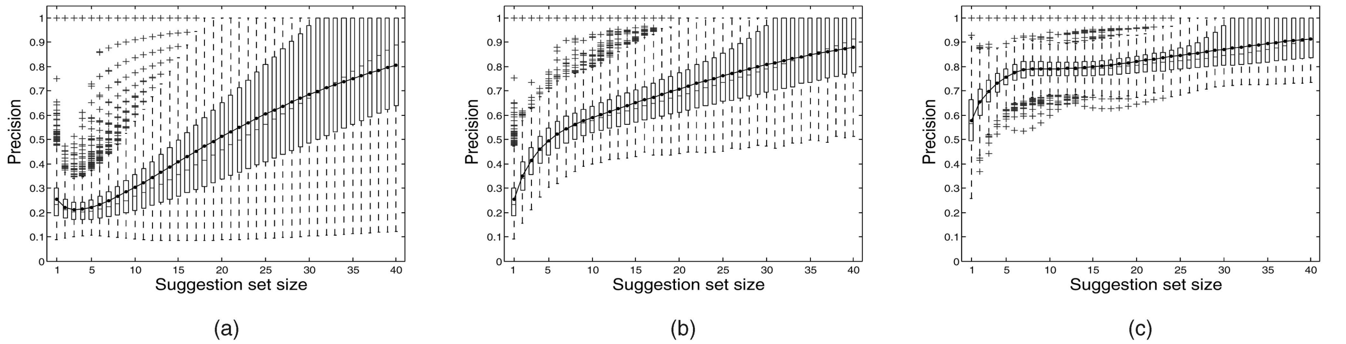

Fig. 9. Average precision of the suggestion set: (a) PROP, (b) M2S, and (c) FM2S versus the suggestion set size. The results are obtained using Theorems 2, 3, 4, and 5 with the inferred true tag popularities.

results here and refer the interested reader to [22, Fig. 6]. In summary, the results provided no evidence that either M2S or FM2S are slower than TOP.

8 Concluding Remarks

We proposed simple randomized algorithms for ranking and suggesting popular items designed to account for popularity bias. We focused on understanding the limit ranking of the items provided by the algorithms, and how it relates to that of the true popularity ranking and assessed the quality of suggestions as measured by the true popularity of suggested items. We believe that the problem posed in this paper opens interesting directions for future research including analysis of convergence rates of the ranking algorithms, consideration of alternative ranking and suggesting rules, and alternative user choice models.

Appendix

A.1 Proof of Theorem 1

Item 1. Let $V_i(t)$ be the cumulative number of item i selections over t item selections. Under the user's choice model and the TOP POPULAR suggestion rule, V is a Markov chain specified by the transition probabilities

$$P(V'|V) = \alpha_i(V),$$

where $V' = V + e_i$ with e_i a vector of dimension c with all the coordinates equal to 0, but the ith equal to 1, and

$$\alpha_i(V) = (1-p)r_i + p\frac{r_i 1_{i\in S_V}}{\sum_{j=1}^{c} r_j 1_{j\in S_V}},$$

where S_V is a set of s most popular items with respect to the rank scores V.

By the law of large numbers, $V(t)/t$ converges to $\rho(S)$ as t goes to infinity, where $\rho(S)$ is given by (3) and S is a suggestion set that satisfies $\min_{i\in S}\rho_i(S) \geq \max_{i\in C\setminus S}\rho_i(S)$. Combining with (3), the latter condition can be rewritten as (4).

Item 2. *Direct*: Suppose that $\rho(S)$ with $S = \{1,\dots,s\}$ is a unique stationary ranking, for given imitation rate p. From (4), we have

$$\min_{S'\in S_s\setminus\{1,\dots,s\}} f(S') > \frac{p}{1-p}, \tag{11}$$

where

$$f(S) = \left(\sum_{j\in S} r_j\right)\left(\frac{\max_{j\in S_s\setminus S} r_j}{\min_{j\in S} r_j} - 1\right). \tag{12}$$

Note that condition (11) is equivalent to $p < f^*/(1+f^*)$ where $f^* := \min_{S'\in S_s\setminus\{1,\dots,s\}} f(S')$ and, recall that S_s contains subsets of C of size s. It suffices to show that $f^* = \min_{0\leq i<s<j\leq c} a(i,j)$, where a is defined in the theorem. To that end, partition the set S_s in the following way. For a set $S' \in S_s$, let i and j be such that the following hold: $r_{i+1} = \max_{k\in C\setminus S'} r_k$ and $r_j = \min_{k\in S'} r_k$. From (12), we have

$$\min_{S'\in S_s\setminus\{1,\dots,s\}} f(S') = \min_{0\leq i<s<j\leq c}\ \min_{S':\{i,j\}\subseteq S'} g(i,j,S'),$$

where $g(i,j,S') = (r_1+\cdots+r_i+\sum_{k\in S'\setminus\{i,j\}} r_k)(\frac{r_{i+1}}{r_j}-1)$. The result follows by noting that

$$\begin{aligned}
&\min_{S':\{i,j\}\subseteq S'} g(i,j,S') \\
&= (r_1+\cdots+r_i+r_{j-s+i+1}+\cdots+r_j)\left(\frac{r_{i+1}}{r_j}-1\right) \\
&= a(i,j).
\end{aligned}$$

Converse: Suppose $p \geq p_{\text{crit}}(r,s)$. By the above identities, this means that there exists a set $S' \in S_s \setminus \{1,\dots,s\}$ such that (4) holds, which completes the proof. □

A.2 Proof of Theorem 2

Item 1. V is a Markov chain with the transition probabilities specified by

$$P(V'|V) = \alpha_i(V), \tag{13}$$

with $V' = V + e_i$, where

$$\alpha_i(V) = (1-p)r_i + p\sum_{S'\in S_s}\frac{r_i 1_{i\in S'}}{\sum_{j\in S'} r_j} f_{S'}(V),$$

$$f_{S'}(V) := \frac{\sum_{j\in S'} V_j}{\sum_{S'\in S_s}\sum_{j\in S'} V_j}.$$

Note that $\sum_{S'\in S_s}\sum_{j\in S'} V_j(t) = m\cdot t$, where $m := \binom{c-1}{s-1}$. This follows from

$$\sum_{S' \in S_s} \sum_{j \in S} V_j(t) = \sum_{j=1}^{c} \left(\sum_{S' \in S_s} 1_{j \in S'} \right) V_j(t)$$
$$= \binom{c-1}{s-1} \sum_{j=1}^{c} V_j(t) = \binom{c-1}{s-1} \cdot t = m \cdot t.$$

Let $v(t) = \mathbb{E}(V(t))$. It follows that

$$v_i(t+1) = v_i(t) + (1-p)r_i + p \frac{1}{m \cdot t} \sum_{S' \in S_s} \frac{r_i 1_{i \in S'}}{\sum_{j \in S'} r_j} \sum_{j \in S'} v_j(t),$$

where $m := \binom{c-1}{s-1}$. For $\bar{\rho}(t)$ defined as $\bar{\rho}(t) = v(t)/t$, we have

$$\bar{\rho}_i(t+1) = \bar{\rho}_i(t) + \frac{1}{t} \Bigg((1-p)r_i + p \frac{1}{m} \sum_{S' \in S_s} \frac{r_i 1_{i \in S'}}{\sum_{j \in S'} r_j} \sum_{j \in S'} \bar{\rho}_j(t) - \bar{\rho}_i(t) \Bigg).$$

The limit points of the last system can be studied by considering the following ordinary differential system:

$$\frac{d}{du} \bar{\rho} = (1-p)r + (pA - I)\bar{\rho}, \tag{14}$$

where $u(t) = \log(t)$, I is a $c \times c$ identity matrix, and $A = (a_{ij})$ is given by (6).

The proof follows by noting that: 1) the condition (5) is the stability condition for the system of the linear differential equations (14), 2) r is a stationary point of (14), and finally 3) $\rho(+\infty) = \lim_{t \to +\infty} \bar{\rho}(t)$.

Saying that r is a stationary point means that for $\bar{\rho}_i = r$, it holds $(d/du)\bar{\rho}_i = 0$ which for the system (14) is the same as

$$\bar{\rho}_i = (1-p)r_i + p \frac{1}{m} \sum_{S' \in S_s} \frac{r_i 1_{i \in S'}}{\sum_{j \in S'} r_j} \sum_{j \in S'} \bar{\rho}_j.$$

The proof follows by plugging $\bar{\rho} = r$ in the last identity:

$$r_i = (1-p)r_i + p \frac{1}{m} \sum_{S' \in S_s} \frac{r_i 1_{i \in S'}}{\sum_{j \in S'} r_j} \sum_{j \in S'} r_j$$
$$= (1-p)r_i + p r_i \frac{1}{m} \left(\sum_{S' \in S_s} 1_{i \in S'} \right)$$
$$= (1-p)r_i + p r_i = r_i.$$

Item 2. Under condition (5), we have that the limit distribution of the suggestion set is

$$\pi(S) = \binom{c-1}{s-1}^{-1} \sum_{j \in S} r_j, \quad S \in S_s,$$

where S_s contains all subsets of C of size s. We have

$$s_i = \Pr(i \in S) = \sum_{S' \in S_s} \pi(S') 1_{i \in S'}$$
$$= \binom{c-1}{s-1}^{-1} \sum_{j=1}^{c} \left(\sum_{S' \in S_s} 1_{i \in S', j \in S'} \right) r_j$$
$$= r_i + \frac{s-1}{c-1}(1 - r_i).$$

The expected precision follows by computing the expected value of the precision defined in (1) using the latter probability distribution. □

A.3 Proof of Corollary 1

We use the well known row-sum bound for the spectral radius of a non-negative matrix A:

$$\sigma(A) \leq \max_i r(i),$$

where $r(i)$ is row-sum of the ith row of the matrix A. We have

$$r(i) = s \frac{1}{\binom{c-1}{s-1}} \sum_{S' \in S_{s-1}(C \setminus \{i\})} \frac{1}{1 + \frac{\sum_{k \in S'} r_k}{r_i}}, \ i = 1, \ldots, c,$$

where $S_{s-1}(C \setminus \{i\})$ contains all subsets of set $C \setminus \{i\}$ of size $s-1$. The asserted sufficient condition follows from $p < 1/\max_i r(i) \Rightarrow p < 1/\sigma(A)$ and $1/\max_i r(i) \geq (1 + (\sum_{k=c-s+1}^{c} r_k)/r_1)/s$. □

A.4 Proof of Theorem 3

Item 1. Let $X_i(t) = 1$ if just before the tth selection of an item, item i is in the suggestion set and $X_i(t) = 0$ otherwise. It suffices to consider the dynamics of the suggested set for the imitation parameter $p = 0$ as for any other value $0 < p < 1$; we have only rescaling of the time with the factor $(1-p)$ and thus the limit distribution of X remains the same. X is a Markov chain with the state space $E_s = \{x \in \{0,1\}^c : \sum_k x_k = s\}$ and the transition probabilities defined by

$$p(y|x) = \begin{cases} \frac{1}{s} r_i (1 - x_i) x_j, & \text{for } y = x + e_i - e_j, \\ \frac{1}{s} \sum_k r_k x_k, & \text{for } y = x. \end{cases} \tag{15}$$

From the facts that 1) X is a Markov chain on a finite state space and 2) the transition matrix is irreducible (i.e., the graph defined by the transition matrix is connected), we have that X has a unique stationary distribution π. In fact, X is a reversible Markov chain, i.e., for $x, y \in E_s$, it satisfies

$$p(y|x)\pi(x) = p(x|y)\pi(y). \tag{16}$$

We now show that (16) holds for

$$\pi(x) = \lambda \prod_{k:\, x_k = 1} r_k, \tag{17}$$

where λ is the normalization constant. It suffices to show that (16) holds for $y \neq x$. Using (15), we rewrite (16) for x with $x_i = 0$ and $x_j = 1$

$$\pi(x) r_i = \pi(x + e_i - e_j) r_j.$$

Now, note that for π defined by (17), we have

$$\pi(x) = \lambda \prod_{k:\, x_k = 1, k \notin \{i,j\}} r_k,$$
$$\pi(x + e_i - e_j) = \lambda \prod_{k:\, x_k = 1, k \notin \{i,j\}} r_k.$$

The asserted result follows.

Remark 1. It turns out that the same dynamics, but in continuous time, is equivalent to that of exclusion process (see Liggett [12, Chapter VIII]). With an exclusion process, we have a set of sites. Each site can be occupied by at most one particle. Each particle attempts to move to a site at instances of a Poisson process with positive rate. A particle at site u attempts to move to site v with given probability $p(u,v)$, and such an attempt is successful only if site v is not already occupied by a particle. We note that our process X is an exclusion process where sites are items, particles are items in the suggestion set, each particle attempts to move at instances of a Poisson process with rate $1/s$, and $p(u,v) = r_v$. The transition rates of X are specified by

$$X \to X + e_i - e_j \text{ with rate } \frac{1}{s} r_i (1 - X_i) X_j. \tag{18}$$

Items 2 and 2a. Let $\phi_r(A) := \prod_{k \in A} r_k$, $A \subseteq C$. The long-run frequency s_i at which item i is suggested is given by

$$s_i = \frac{\sum_{S' \in S_s : i \in S'} \phi_r(S')}{\sum_{S' \in S_s} \phi_r(S')}. \tag{19}$$

We need to show that $r_i \geq r_j$ implies $s_i \geq s_j$. Note that $s_i \geq s_j$ is equivalent to

$$\sum_{S' \in S_s : i \in S'} \phi_r(S') \geq \sum_{S' \in S_s : j \in S'} \phi_r(S').$$

Now, the last inequality is equivalent to

$$\sum_{S' \in S_s : i \in S', j \notin S'} \phi_r(S') \geq \sum_{S' \in S_s : i \notin S', j \in S'} \phi_r(S'),$$

which is further equivalent to

$$r_i \sum_{S' \in S_s : i \in S', j \notin S'} \phi_r(S' \setminus \{i\}) \geq r_j \sum_{S' \in S_s : i \notin S', j \in S'} \phi_r(S' \setminus \{j\}).$$

The result follows from the last inequality as it is the same as saying that $r_i \geq r_j$ because

$$\begin{aligned} \sum_{S' \in S_s : i \in S', j \notin S'} \phi_r(S' \setminus \{i\}) &= \sum_{S' \in S_s : i \notin S', j \in S'} \phi_r(S' \setminus \{j\}) \\ &= \sum_{S' \in S_{s-1}(C \setminus \{i,j\})} \phi_r(S'), \end{aligned} \tag{20}$$

where $S_s(A)$ contains all subsets of set A of size s.

Item 2b. We want to show that $r_i \geq r_j$ implies $s_i/r_i \leq s_j/r_j$. Combining with (19), the latter is the same as

$$\sum_{S' \in S_s : i \in S'} \phi_r(S' \setminus \{i\}) \leq \sum_{S' \in S_s : j \in S'} \phi_r(S' \setminus \{j\}).$$

The last inequality can be rewritten as

$$\begin{aligned} &\sum_{S' \in S_s : \{i,j\} \subseteq S'} \frac{1}{r_i} \phi_r(S') + \sum_{S' \in S_s : i \in S', j \notin S'} \phi_r(S' \setminus \{i\}) \\ &\leq \sum_{S' \in S_s : \{i,j\} \subseteq S'} \frac{1}{r_j} \phi_r(S') + \sum_{S' \in S_s : j \in S', i \notin S'} \phi_r(S' \setminus \{j\}). \end{aligned}$$

Now, note that by $r_i \geq r_j$, indeed

$$\sum_{S' \in S_s : \{i,j\} \subseteq S'} \frac{1}{r_i} \phi_r(S') \leq \sum_{S' \in S_s : \{i,j\} \subseteq S} \frac{1}{r_j} \phi_r(S').$$

It remains only to show that

$$\sum_{S' \in S_s : i \in S', j \notin S'} \phi_r(S' \setminus \{i\}) \leq \sum_{S' \in S_s : j \in S', i \notin S'} \phi_r(S' \setminus \{j\}).$$

But this clearly holds in view of (20).

Item 3. The result follows from Theorem 4. Indeed, consider any suggestion set S such that $i \notin S$ and $j \in S$ for some i and j for which it holds $r_i \geq r_j$. Let $S' = S \setminus \{j\} \cup \{i\}$. We then have

$$\pi(S) = \lambda \prod_{k \in S \setminus \{j\}} r_k \cdot r_j \leq \lambda \prod_{k \in S \setminus \{j\}} r_k \cdot r_i = \pi(S'),$$

which shows that condition A holds. We assumed that $p_S \equiv p$ for $0 \leq p < 1$; thus, condition B is true.

A.5 Proof of Theorem 5

A.5.1 System State and Dynamics

Let $W_i(t)$ be the cumulative number of item i selections when this item was not suggested. Let $X_i(t) = 1$ if candidate i is in the suggestion set and $X_i(t) = 0$ otherwise. Let $Z(t)$ be the minimum $W_i(t)$ over the items i that are in the suggestion set, i.e.,

$$Z(t) = \min\{W_i(t),\ i \in C : X_i(t) = 1\}.$$

Further, let $M(t)$ be the number of items that are in the suggested set with $W_i(t)$ of such an item equal to $Z(t)$, i.e.,

$$M(t) = \{i \in C : X_i(t) = 1, W_i(t) = Z(t)\}.$$

It is readily checked that $\Phi(t) = (W(t), X(t), M(t), Z(t))$ fully describes the system dynamics and is a Markov process specified by the transition probabilities:

$$\begin{aligned} p(\Phi'|\Phi) &= r_i(1 - X_i) 1_{W_i < Z}, \\ p(\Phi''|\Phi) &= r_i(1 - X_i) 1_{W_i = Z} \frac{X_j 1_{W_j = Z}}{M} 1_{M > 1}, \\ p(\Phi'''|\Phi) &= r_i(1 - X_i) 1_{W_i = Z} X_j 1_{W_j = Z} 1_{M = 1}, \\ p(\Phi|\Phi) &= 1 - p(\Phi'|\Phi) - p(\Phi''|\Phi) - p(\Phi'''|\Phi), \end{aligned}$$

where

$$\begin{aligned} \Phi' &= \Phi + (e_i, 0, 0, 0), \\ \Phi'' &= \Phi + (e_i, e_i - e_j, -1, 0), \\ \Phi''' &= \Phi + (e_i, e_i - e_j, s - 1, 1). \end{aligned}$$

In the sequel, we consider Φ redefined as follows. Let $D(t) = Z(t) - W(t)$ and consider the Markov process

$$\Phi(t) = (D(t), X(t), M(t)) \tag{21}$$

specified by the transition probabilities:

$$\begin{aligned} p(\Phi'|\Phi) &= r_i(1 - X_i) 1_{D_i > 0}, \\ p(\Phi''|\Phi) &= r_i(1 - X_i) 1_{D_i = 0} \frac{X_j 1_{D_j = 0}}{M} 1_{M > 1}, \\ p(\Phi'''|\Phi) &= r_i(1 - X_i) 1_{D_i = 0} X_j 1_{D_j = 0} 1_{M = 1}, \\ p(\Phi|\Phi) &= 1 - p(\Phi'|\Phi) - p(\Phi''|\Phi) - p(\Phi'''|\Phi), \end{aligned}$$

where

$$\Phi' = \Phi + (-e_i, 0, 0),$$
$$\Phi'' = \Phi + (-e_i, e_i - e_j, -1),$$
$$\Phi''' = \Phi + (1 - e_i, e_i - e_j, s - 1).$$

A.5.2 Necessary Condition for Positive Recurrence of Φ

The following lemma provides a necessary condition for Φ to be positive recurrent.

Lemma 2. *If for a given r and s, Φ is positive recurrent, then r and s must satisfy*

$$r_c > \frac{c-s}{\frac{1}{r_1} + \cdots \frac{1}{r_c}}. \quad (22)$$

Proof. By definition of Φ,

$$D_i(t+1) = \begin{cases} D_i(t) + 1, & \text{w. p.} \quad \alpha_i(\Phi(t)), \\ D_i(t) - 1, & \text{w. p.} \quad \beta_i(\Phi(t)), \\ D_i(t), & \text{w. p.} \quad \gamma_i(\Phi(t)), \end{cases} \quad (23)$$

where

$$\alpha_i(\Phi) = \sum_{j=1}^{c} r_j(1 - X_j)1_{D_j=0,M=1} - r_i(1 - X_i)1_{D_i=0,M=1},$$
$$\beta_i(\Phi) = r_i(1 - X_i)(1 - 1_{D_i=0,M=1}),$$
$$\gamma_i(\Phi) = 1 - \alpha_i(\Phi) - \beta_i(\Phi).$$

We have that Φ is irreducible and positive recurrent, hence, it is ergodic. We thus have $\lim_{t\to+\infty}(\mathbb{E}(D_i(t+1)) - \mathbb{E}(D_i(t))) = 0$ for all $i \in C$. From (23) and the last identity, it follows that for arbitrarily fixed $m \in C$, we have for all $i \in C$

$$\lim_{t\to+\infty} \{r_i[1 - \mathbb{E}(X_i(t))] - r_m[1 - \mathbb{E}(X_m(t))]\} = 0. \quad (24)$$

It follows that for all $i \in C$

$$\mathbb{E}(X_i(+\infty)) = 1 - r_m[1 - \mathbb{E}(X_m(+\infty))]\frac{1}{r_i}. \quad (25)$$

Note that

$$\sum_{i=1}^{c} X_i(t) = s, \text{ for all } t \geq 0. \quad (26)$$

Combining the last two identities, we have $r_m(1 - \mathbb{E}(X_m(+\infty))) = \left(1 - \frac{s}{c}\right)h_c(r)$. Inserting the last identity into (25), we obtain

$$\lim_{t\to+\infty} \mathbb{E}(X_i(t)) = 1 - \left(1 - \frac{s}{c}\right)\frac{h_c(r)}{r_i}, \; i \in C. \quad (27)$$

As Φ is positive recurrent, it must be $\mathbb{E}(X_i(+\infty)) > 0$, for all $i \in C$. By the above identity, this is equivalent to $1 - \left(1 - \frac{s}{c}\right)\frac{h_c(r)}{r_i} > 0$, for $i \in C$, which after some elementary calculus can be rewritten as (22). □

Remark 2. We showed that condition (22) is necessary for Φ to be positive recurrent. While we believe that the condition (22) would also be sufficient, it remains open to prove this. To this end, one may try the Lyapunov stability methods.

A.5.3 The Case when Φ Is Not Positive Recurrent

We next consider the more general case where r and s can be such that Φ is not positive recurrent. In this case, we have that some items do not recurrently enter the suggestion set. Assume that C' is a subset of C such that

1. $((D_i, X_i), i \in C')$ is positive recurrent and
2. $\mathbb{E}(X_i(+\infty)) = 0$, for $i \in C \setminus C'$.

Lemma 3. *Conditions 1 and 2 imply that all the following conditions hold:*

$$\lim_{t\to+\infty} \mathbb{E}(X_i(t)); > 0, \quad \text{for } i \in C', \quad (28)$$

$$r_j(1 - \mathbb{E}(X_j(+\infty)) = r_i(1 - \mathbb{E}(X_i(+\infty))), \; i, j \in C', \quad (29)$$

$$r_j(1 - \mathbb{E}(X_j(+\infty))) \geq r_i, \; i \in C \setminus C', j \in C'. \quad (30)$$

Proof. Condition (28) follows directly by the positive recurrence in condition 1. By the same arguments as in the preceding section, it follows that condition (29) must hold. We next show that (30) must hold. From (23), it follows that for $k \in C'$ and $i \in C \setminus C'$,

$$\begin{aligned} &\lim_{t\to+\infty} [\mathbb{E}(D_i(t+1)) - \mathbb{E}(D_i(t))] \\ &\quad = r_k(1 - \mathbb{E}(X_k(+\infty)) - r_i. \end{aligned} \quad (31)$$

To contradict, let us assume that (30) does not hold, i.e., $r_i < r_k(1 - \mathbb{E}(X_k(+\infty))$, for $k \in C'$ and $i \in C \setminus C'$. From (31), it follows that

$$\lim_{t\to+\infty} [\mathbb{E}(D_i(t+1)) - \mathbb{E}(D_i(t))] < 0, \quad \text{for } i \in C \setminus C',$$

which implies that $\mathbb{E}(D_i(t))$ tends to $-\infty$ at t goes to infinity, which cannot hold as by definition, $\Pr(D_i(t) \geq -1) = 1$, for any $t \geq 0$ and $i \in C$. □

We next define the set C' in terms of the parameters r and s. We have the following.

Lemma 4. *Any set C' that satisfies conditions (28), (29), and (30) must be such that $C' = \{1, \ldots, a\}$ for some $a \in \{\min(s, c), \ldots, c\}$.*

Proof. Indeed, the assertion is true as for $i \in C'$ and $r_j \geq r_i$, conditions (28), (29), and (30) imply $j \in C'$. □

It remains only to show that a in the lemma is equal to the expression asserted in the theorem. By condition 2, (29), and (26), we have

$$\mathbb{E}(X_i(+\infty)) = 1 - \left(1 - \frac{s}{a}\right)\frac{h_a(r)}{r_i}, \; i = 1, \ldots, a. \quad (32)$$

Now, note that (28) is equivalent to $r_a > (a - s)/(\frac{1}{r_1} + \cdots + \frac{1}{r_a})$. By simple rearrangements, we have that the last condition is equivalent to: $a - r_a(\frac{1}{r_1} + \cdots + \frac{1}{r_a}) < s$. Let $f(a)$ be defined by the left-hand side of the last inequality. It can be easily checked that $f(a)$ is nondecreasing with a. Hence, it follows that a is such that $\min(s, c) \leq a \leq c'$, where

$$c' = \max\left\{i \in \{b, \ldots, c\} : i - r_i\left(\frac{1}{r_1} + \cdots + \frac{1}{r_i}\right) < s\right\}$$

with $b := \min(s, c)$. We next show that, in fact, it must be $a = c'$ for the conditions (28), (29), and (30) to hold. To contradict, suppose that $a < c'$. By the definition of c', we have $r_{c'+1} > (c' + 1 - s)/(\frac{1}{r_1} + \cdots + \frac{1}{r_{c'+1}})$. By a simple algebra, we note that the last condition is equivalent to: $r_{c'+1} > (c' - s)/ (\frac{1}{r_1} + \cdots + \frac{1}{r_{c'}})$. But this violates condition (30); hence, a contradiction.

Item 2. The asserted frequencies of the appearance of items in the suggestion set follow from (32).

Item 3. Take the expectation in (1) and use $\mathbb{E}(1_{i \in S}) = s_i$, where s_i's are given by (10). □

Acknowledgments

The work by J. Cruise and P. Marbach was done in part while an intern and a visiting researcher with Microsoft Research Cambridge.

References

[1] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically Efficient Allocation Rules for the Multiarmed Bandit Problem with Multiple Plays—Part i: i.i.d. Rewards," *IEEE Trans. Automatic Control,* vol. 32, no. 11, pp. 968-976, Nov. 1987.

[2] J.R. Anderson, "The Adaptive Nature of Human Categorization," *Psychological Rev.,* vol. 98, no. 3, pp. 409-429, 1991.

[3] A.L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science,* vol. 286, pp. 509-512, 1999.

[4] S. Brams and P. Fishburn, *Approval Voting.* Birkhauser, 1983.

[5] S. Chakrabarti, A. Frieze, and J. Vera, "The Influence of Search Engines on Preferential Attachment," *Proc. Symp. Discrete Algorithms (SODA),* 2005.

[6] J. Cho, S. Roy, and R.E. Adams, "Page Quality: In Search of an Unbiased Web Ranking," *Proc. ACM SIGMOD '05,* 2005.

[7] S. Golder and B.A. Huberman, "The Structure of Collaborative Tagging Systems," *J. Information Science,* vol. 32, no. 2, pp. 198-208, 2006.

[8] http://en.wikipedia.org/wiki/Approval_voting, 2009.

[9] J. Kleinberg and M. Sandler, "Using Mixture Models for Collaborative Filtering," *Proc. 36th Ann. ACM Symp. Theory of Computing (STOC),* 2004.

[10] R. Kumar, P. Rajagopalan, and A. Tomkins, "Recommendation Systems: A Probabilistic Analysis," *Proc. 39th Ann. Symp. Foundations of Computer Science (FOCS),* 1998.

[11] T.L. Lai and H. Robbins, "Asymptotically Efficient Adaptive Allocation Rules," *Advances in Applied Math.,* vol. 6, pp. 4-25, 1985.

[12] T.M. Liggett, *Interacting Particle Systems,* second ed. Springer, 2006.

[13] R.D. Luce, *Individual Choice Behavior: A Theoretical Analysis.* Dover, 1959.

[14] R.M. Nosofsky, "Choice, Similarity, and the Context Theory of Classification," *J. Experimental Psychology,* vol. 10, no. 1, pp. 104-114, 1984.

[15] S. Pandey, S. Roy, C. Olston, J. Cho, and S. Chakrabarti, "Shuffling Stacked Deck: The Case for Partially Randomized Ranking of Search Engine Results," *Proc. 31st Int'l Conf. Very Large Data Bases (VLDB),* 2005.

[16] R.M. Phatarfod, "On the Matrix Occurring in a Linear Search Problem," *J. Applied Probability,* vol. 18, pp. 336-346, 1991.

[17] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill Education, 1983.

[18] S. Sen, S.K. Lam, A.-M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F.M. Harper, and J. Riedl, "Tagging, Communities, Vocabulary, Evolution," *Proc. 2006 20th Anniversary Conf. Computer Supported CooperativeWork (CSCW),* 2006.

[19] H.A. Simon, "Bandwagon and Underdog Effects and the Possibility of Election Predictions," *Public Opinion Quarterly,* vol. 18, pp. 245-253, 1954.

[20] F. Suchanek, M. Vojnović, and D. Gunawardena, "Social Tagging: Meaning and Suggestions," *Proc. 17th ACM Conf. Information and Knowledge Management (CIKM),* Oct. 2008.

[21] C.J. van Rijsbergen, *Information Retrieval,* second ed. Butterworths, http://www.dcs.gla.ac.uk/Keith/Preface.html, 1979.

[22] M. Vojnović, J. Cruise, D. Gunawardena, and P. Marbach, "Ranking and Suggesting Tags in Collaborative Tagging Applications," Technical Report MSR-TR-2007-06, Microsoft Research, Feb. 2007.

[23] Z. Xu, Y. Fu, J. Mao, and D. Su, "Towards the Semantic Web: Collaborative Tag Suggestions," *Proc. Workshop Collaborative Web Tagging Workshop at the WWW 2006,* May 2006.

Milan Vojnović received the BSc and MSc degrees in electrical engineering from the University of Split, Croatia, in 1998 and 1995, respectively, and the PhD degree in communication systems from EPFL, Switzerland, in 2003. He is a researcher in the Systems and Networking Group at Microsoft Research Cambridge, United Kingdom. His research interests are in design and analysis of computer systems and networks. He received IWQoS 2007 Best Student Paper Award with Shao Liu and Dinan Gunawardena for a work on congestion control protocols, ACM SIGMETRICS 2005 Best Paper Award with Laurent Massoulié for a work on the performance of peer-to-peer file dissemination, IEEE INFOCOM 2005 Best Paper Award with Jean-Yves Le Boudec for a work on the random mobility models, and ITC-17 2001 Best Student Paper Award with Jean-Yves Le Boudec for a work on analysis of equation-based congestion control. In 2005, he has been awarded ERCIM Cor Baayen Award.

James Cruise received a degree and is currently working toward the PhD degree in mathematics at Cambridge University. He spent three months completing an internship at Microsoft Research, Cambridge, United Kingdom. Recently, he has moved to a post at Bristol University. His work has focused on asympotics for queuing theory.

Dinan Gunawardena received the BA and MA degrees in computer science from Cambridge University, United Kingdom, in 1999 and 1995, respectively. He is a research software development engineer with the Systems and Networking Group, Microsoft Research Cambridge, United Kingdom. His research interests are in the design and development of multimedia computer systems, and embedded systems and networks. He received the IWQoS 2007 Best Student Paper Award with Shao Liu and Milan Vojnovic for a work on congestion control protocols. He has filed numerous patents in the area of computer systems and networks, and has contributed to networking research that has shipped in the Microsoft Vista Operating System. He also contributed to key enabling technology that is implemented in the Asymmetric Digital Subscriber Line (ADSL) standards.

Peter Marbach received the Eidg Dipl El-Ing from the ETH Zurich, Switzerland, in 1993, the MS degree in electrical engineering from the Columbia University, New York, in 1994, and the PhD degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, in 1998. He was appointed as an assistant professor in 2000, and as an associate professor in 2005, at the Department of Computer Science, University of Toronto. He has also been a visiting professor at Microsoft Research, Cambridge, United Kingdom, at the Ecole Polytechnique Federal at Lausanne (EPFL), Switzerland, and at the Ecole Normale Superieure, Paris, France, and a postdoctoral fellow at Cambridge University, United Kingdom. His research interests are in the fields of communication networks, in particular in the area of wireless networks, peer-to-peer networks, and online social networks. He has received the IEEE INFOCOM 2002 Best Paper Award for his paper "Priority Service and Max-Min Fairness." He is on the Editorial Board of the *ACM/IEEE Transactions of Networking*.