Adaptive Collaboration in Peer-to-Peer Systems (Extended Abstract)

Baruch Awerbuch
Johns Hopkins University
baruch@acm.org

Boaz Patt-Shamir
Tel Aviv University
boaz@eng.tau.ac.il

David Peleg
Weizmann Institute
david.peleg@weizmann.ac.il

Mark Tuttle
HP Labs
mark.tuttle@hp.com

Abstract

We consider a simple model for reputation systems such as the one used by eBay. In our model there are n players, some of which may exhibit arbitrarily malicious (Byzantine) behavior, and there are m objects, some of which are bad. The goal of the honest players is to find a good object. To facilitate collaboration, the system maintains a shared billboard. A basic step of a player consists of consulting the billboard, probing an object to learn its true value, and posting the result on the billboard for the benefit of others. Probing an object incurs a unit cost to the player, and consulting the billboard is free. The dilemma of an honest player is how to balance between the desire to reduce its cost by taking advantage of the reports posted by honest peers, and the fear of being exploited by adopting reports posted by malicious players.

In prior work, we presented an algorithm solving this problem in an asynchronous model, and we analyzed the total cost *of the probes made by honest players during the algorithm. In this paper, we focus on the* individual cost, *and we consider a synchronous model in which each player takes a step in each round. Our prior algorithm has individual cost $O\left(\frac{1}{\alpha}\log n\right)$ in this model, assuming that an α fraction of players are honest. In this paper, we prove that no algorithm can guarantee individual cost of less than $\Omega\left(\frac{1}{\alpha}\right)$, which is essentially constant if there are enough honest players. Our main result is a new algorithm that achieves $O(1)$ individual cost when there are many honest players, and achieves individual cost $O\left(\frac{1}{\alpha}\frac{\log n}{\log\log n}\right)$ even when there are not. We also show that this algorithm generalizes to other interesting scenarios.*

1 Introduction

The commerce giant eBay depends heavily on its reputation system for its success [15]. After each transaction, the system invites each party to rate the other party on a public billboard maintained by the system. Looking up the record of the other party is a key step before making any transaction. Unfortunately, malicious users can collude and post false information on this billboard, inducing other users into fraudulent transactions with catastrophic results [6, 9]. In this paper, we study algorithmic solutions to using such billboards effectively, even in the face of malicious users.

1.1 Our model

In prior work, we proposed [1] a simple model for studying collaboration on eBay. In this model, there are n players and m objects. Each object has an unknown *value* and known *cost*, both specified as real nonnegative numbers. We divide the objects into *good* (high value) and *bad* (low value) objects. A player can *probe* each object, discovering its value and incurring its cost. The goal of each player is to find a good object at minimal cost.

To facilitate collaboration among the players, the system supports a public billboard where players report their experience with objects. Consulting the billboard is free, and is intended to help reduce the probing costs. Unfortunately, while some players are *honest* and follow the protocol, others are *dishonest* and can behave in an arbitrary fashion, including colluding and posting bogus reports on the billboard so as to maximize the cost to the honest players.

In this model, we studied [1] the total cost to the honest players of finding good objects. We considered an asynchronous model, where a basic step is a single player reading the billboard, probing an object, and updating the billboard; the player schedule is assumed to be under the control of the adversary. We gave a simple algorithm where the *total cost* to the honest players of finding good objects is $O(\frac{1}{\beta} + n\log n)$, regardless of the number of dishonest players, where $0 < \beta \leq 1$ is the fraction of good objects.

1.2 Our Results

In this paper, we focus on the *individual* cost to the honest players. The asynchronous model is obviously not a good model for studying bounds on individual cost. A schedule that runs a single player by itself forces that player to find the good object on its own without any assistance from any other player. We must make some restriction on the allowable schedules to get any meaningful results. Synchronous models are a convenient abstraction of asynchronous models where players are running at more or less the same speed. Furthermore, we can often simulate synchronous behavior in asynchronous environments with the use of timestamps (an integral part of any posting on any real billboard). In this paper, we consider a synchronous model where computation proceeds in a sequence of rounds, and each player probes one object in each round until it probes a good object. In this model, we prove nearly tight bounds on the expected number of rounds (and hence the expected cost) until an honest player finds a good object.

We start in Section 3 with two simple lower bounds. The first is due to the collective amount of work required until *any* player finds a good object, and the second holds even in the case where all other honest players have already found a good object. Together, these results imply that no algorithm can halt in less than expected $\Omega\left(\frac{1}{\alpha\beta n}+\frac{1}{\alpha}\right)$ rounds, where $0<\alpha\leq 1$ and $0<\beta\leq 1$ denote the fraction of honest players and good objects.

Next, in Section 4, we present our main result: a randomized algorithm DISTILL that is always better than the asynchronous algorithm of [1]. Specifically, the asynchronous algorithm, when considered under a synchronous schedule (say, round robin), halts in expected time $O\left(\frac{\log n}{\alpha\beta n}+\frac{\log n}{\alpha}\right)$. Most players are honest in the real world (meaning α is close to 1), but in this case the expected cost to an honest player is still $\Omega(\log n)$. In contrast, the DISTILL algorithm has $O(1)$ individual cost when most players are honest, and improves the $O(\log n)$ individual cost by a $O(\log\log n)$ factor even when many players are dishonest. The individual cost is stated precisely in Theorem 4, but if there are just n^{ϵ} dishonest players for some constant $\epsilon>0$, then the expected cost per player is just the constant $O\left(\frac{1}{1-\epsilon}\right)$. The idea of the algorithm is quite intuitive, as the following example demonstrates.

Informally, the idea is that the algorithm works by refining a set of candidate objects. To illustrate the idea, suppose for the moment that there are $m=n$ objects and only $\sqrt{n}$ dishonest players. Consider the following three-phase algorithm which is a simplification of DISTILL. Each phase i consists of two rounds in which each player probes a random object from a candidate set C_i and posts the result on the billboard. Each candidate set C_i is the set of objects recommended by at least θ_i players on the billboard at the start of phase i, where θ_i is a threshold for phase i. We use $\theta_1=0$, $\theta_2=1$, and $\theta_3=\sqrt{n}/2$.

We claim that each candidate set contains the good object i_0 with constant probability. The set C_1 contains i_0 since it contains all objects. In phase 1, each probe by an honest player probes i_0 with probability $1/n$; there are at least $2(n-\sqrt{n})\geq n$ probes by honest players in the two rounds of phase 1; so at least one honest player recommends i_0 by the end of phase 1 with probability at least $1-(1-1/n)^n>1-1/e$. The set C_2, therefore, contains i_0 with constant probability, and contains at most $\sqrt{n}+1\approx\sqrt{n}$ objects since the $\sqrt{n}$ dishonest players can recommend at most $\sqrt{n}$ bad objects. In phase 2, each probe by an honest player probes i_0 with probability at least $1/\sqrt{n}$, and there are at least $2(n-\sqrt{n})\geq n$ probes by honest players in the two rounds of phase 2, so we expect $\sqrt{n}$ players to recommend i_0 by the end of phase 2. By Markov's inequality, at least $\sqrt{n}/2$ players recommend i_0 with constant probability. The set C_3, therefore, contains i_0 with constant probability, and contains at most 3 objects since the $\sqrt{n}$ dishonest players can recommend at most 2 bad objects $\sqrt{n}/2$ times. In phase 3, each player can probe these 3 objects and halt within 3 rounds.

Obviously, the simplistic analysis above breaks down when the number of dishonest players is large. The hard part of the analysis in Section 4 is to show that regardless of the number of dishonest players, and regardless of the adversarial strategy they employ, the algorithm does quite well in terms of the expected individual cost.

1.3 Related work

Our algorithms depend on collaboration and recommendations, and are closely related to the literature on collaborative filtering or recommendations systems.

Most prior research on recommendation systems focused on a centralized, off-line version of the problem, where the algorithm is presented with a lot of historical preference data, and the task is to generate a single recommendation that maximizes the utility to the user. This is usually done by heuristically identifying clusters of users [14] (or products [16]) in the data set, and using past grades by users in a cluster to predict future grades by other users in the same cluster. SVD was shown also to be effective for the off-line problem [10]. Some of these systems enjoy industrial success, but they are known to perform poorly when prior data is less than plentiful [17], and they are extremely vulnerable even to mild attacks [12, 13]. Canny [5] gives a distributed secure and private SVD computation for the off-line version of the problem.

Theoretical studies of recommendation systems usually take the latent variable model approach: a stochastic process is assumed to generate noisy observations, and the goal of an algorithm is to approximate some unknown parameters of the model. Kumar *et al.* [11] study the off-line problem for a model where preferences are identified with past choices (purchases). In this model there are clusters of products. Each user has a probability distribution over clusters; a user first chooses a cluster by his distribution, and then chooses a product uniformly at random from that cluster. The goal is to recommended a product from the user's most preferred cluster. Kleinberg and Sandler [8] generalized this model to the case where the choice within a cluster is governed by an arbitrary probability distribution, and also consider the mixture model, in which each cluster is a probability distribution over all products. Azar *et al.* [3] consider a model where there exists an unknown user-product preference matrix which can be approximated by a low-rank matrix. The system observes this matrix only after its entries were subjected to random additive noise and then to random omissions. They use SVD to reconstruct the original preferences.

Recommendation systems are often vulnerable to malicious users spamming the system. For example, web search algorithms [4, 7] are a form of recommendation systems, but these algorithms essentially compute the popularity of a page, and are known to be vulnerable malicious users who generate lots of links to a page to boost the perceived popularity of the page. Such popularity-style algorithms actually enhance the power of malicious users. Empirical evidence to that effect is provided in the work by Kamvar *et al.* [6], which studies trust in the context of authenticity of files downloaded in peer-to-peer systems. Using a variant of Kleinberg's algorithm [7], they assign a trust value to each peer. They comment that this approach is useful only if there are nodes that are known *a priori* to be trustworthy: Otherwise, "forming a malicious collective in fact heavily boosts the trust values of malicious nodes."

In previous work, we demonstrated [2] that our asynchronous algorithm [1] for reputation systems can also be used as an on-line solution to the recommendation problem, even in the presence of malicious users. The DISTILL algorithm in this paper has the limitation of being a synchronous algorithm and requiring knowledge of α, but it has the advantage of also being a particularly efficient solution to the recommendation problem when the number of honest players is large. The difficult task in this paper is to make the DISTILL technique work even when the number of honest players is small, as we do in Section 4. In addition to using the DISTILL algorithm in recommendation systems, we can use the algorithm when different objects have different costs, and when the goal is to find the best object, whose value is not known in advance, as we show in Section 5.

2 Model

The basic entities in our model are n *players* and m *objects*. Each object has intrinsic unknown *value* and known *cost*, both real non-negative numbers. The collection of objects is divided into two sets: good (high value) objects and bad (low value) ones. The basic operation of a player consists of *probing* an object. In probing an object i, the player pays the (known) cost of i and learns the (hitherto unknown) value of that object. Intuitively, the goal of players is to find a good object while incurring minimal cost. The algorithm presented in Section 4 assumes unit costs; in Section 5 it is extended to the general cost model.

2.1 System environment

By convention, players post the value of objects they have probed after each step they take. It is assumed that

- each message on the billboard is reliably tagged by the identity of the posting player and a timestamp, and that
- the billboard is "append only," in that no message is ever erased from the billboard.

An execution of the system consists of rounds of player steps. In a round, each active player reads the billboard and optionally probes an object and writes to the billboard. A player is called active so long as it hasn't probed a good object.

2.2 Object models

Goodness may or may not be testable by a single player. In the *local testing* model, a player can always determine whether an object is good after probing it. This is the case, for example, when an object is good if its value exceeds a known threshold. In a model without local testing, goodness is defined only by the parameter β, where an object is deemed good if it is one of the βm top valued objects. In particular, a maximum value object can be searched with local testing only if the maximum is known; otherwise, a search algorithm without local testing must be applied, using $\beta = 1/m$. In Sections 4 and 5 we present algorithms that work with and without local testing.

2.3 Player models

We define an *honest* player to be one who always follows the protocol. We denote the fraction of honest players by α (there are αn honest players). Dishonest players follow the *Byzantine* model, that allows them to behave in arbitrary ways. When the algorithm is randomized, it is important to spell out the interplay between the coin flips and the player actions: an *oblivious* adversary must set the actions of the dishonest players independent of the outcome of coin flips; an *adaptive* adversary may determine the actions of the dishonest players based on the results of past coin flips. Our algorithms work against a Byzantine adaptive adversary, which make them very robust. Our lower bounds use a much more benign model, and hence hold in this Byzantine model.

3 Lower Bounds

This section presents two simple lower bounds on the expected number of probes by an honest player in a search with or without local testing. These, in turn, imply lower bounds on the worst-case search time. Each lower bound corresponds to a term in the upper bound of Theorem 4. One lower bound says that it takes some work to discover a good object, even if identities of all honest players are known. The other lower bound formalizes the intuition that distinguishing between friends and foes necessitates probing if appearances are identical. Both proofs use Yao's Minimax Lemma [18].

Our first lower bound is based on the observation that the combined number of probes taken by the honest players should be sufficient to ensure that at least one of them hits a good object.

Theorem 1 *Any randomized algorithm for search (with or without local testing), has an instance where the expected number of probes executed by an individual player is at least* $\Omega\left(\frac{1}{\alpha\beta n}\right)$ *for all* $0 < \alpha, \beta \le 1$.

Proof: We prove the lower bound for the search problem with local testing, hence it also holds for the harder version without local testing. We use Yao's Lemma [18]. Fix m objects and fix n players, of which the first αn are honest. Let $\mathcal{I}$ be the set of all $\binom{m}{\beta m}$ possible labeling of βm objects as good and the others as bad. Consider any deterministic algorithm A that works on $\mathcal{I}$, let us compute the average running time of A on $\mathcal{I}$. Since A is deterministic and since the identity of honest players is fixed across all instances in $\mathcal{I}$, each player probes objects in a fixed order so long as no other player has reported finding a good object. Since the instance is random, we can think of each player as drawing balls from an urn without replacement; furthermore, without loss of generality we might as well assume that no two honest players ever try the same bad object (i.e., the algorithm ensures full cooperation, since the honest players know what reports are trustworthy). Since the "urn" contains m balls, of which a fraction β are good, the expected number of probes until a good object is found is $\frac{m+1}{\beta m+1} = \Omega\left(\frac{1}{\beta}\right)$. Since in each round there are at most αn probes of honest players in each round, the result follows. □

Next, we formalize the intuition that when the adversary maintains complete symmetry between players, no algorithm can easily discern honest players from dishonest ones. The dishonest players in the proof follow the protocol, except that the object values they report are the values dictated by the adversarial strategy. Thus, an instance of the problem consists of defining which players are honest, and what is the value of each object to each player.

Theorem 2 *For any randomized algorithm for search (with or without local testing), there exists an instance and an oblivious adversarial strategy followed by the dishonest players where the expected number of probes executed by an individual player is* $\Omega(\frac{1}{\alpha+\beta}) = \Omega(\min\{\frac{1}{\alpha}, \frac{1}{\beta}\})$.

Proof: For notational simplicity, assume w.l.o.g. that the system contains $n+1$ (rather than n) players of which $\alpha n+1$ (rather than αn) are honest, and that αn, βm, $1/\alpha$ and $1/\beta$ are all integers. Also assume that player 0 is honest. Our goal is to bound from below the number of probes done by player 0. By Yao's Lemma, it is sufficient to prove that there exists a probability distribution over the instances such that for any deterministic algorithm, the average number of probes done by player 0 is $\Omega(\min\{\frac{1}{\alpha}, \frac{1}{\beta}\})$.

Let $B = \min\{\frac{1}{\alpha}, \frac{1}{\beta}\}$. Players $1, \ldots, n$ are partitioned into $\frac{1}{\alpha}$ disjoint subsets $P_1, \ldots, P_{1/\alpha}$, where $|P_k| = n\alpha$ for $k = 1, \ldots, 1/\alpha$. Also, the objects are partitioned into $\frac{1}{\beta}$ disjoint subsets $O_1, \ldots, O_{1/\beta}$, where $|B_k| = m\beta$ for $k = 1, \ldots, 1/\beta$. Our distribution consists of B equiprobable input instances $\mathcal{I}_k$, for $k = 1, 2, \ldots, B$, defined as follows. Player 0 is honest in all instances. The probe values of each player $j \neq 0$ are defined as follows. Let $S^j(i)$ be the value player j reports for object i. If $j \in P_k$, then

$$S^j(i) = \begin{cases} 1, & i \in O_k, \\ 0, & \text{otherwise.} \end{cases}$$

These probe values are thus independent of the particular input instance. For each $k = 1, 2, \ldots, B$, the "true value" function of input instance $\mathcal{I}_k$ is defined as

$$S(i) = \begin{cases} 1, & i \in O_k, \\ 0, & \text{otherwise.} \end{cases}$$

so in instance $\mathcal{I}_k$, $S = S^j$ for the players $j \in P_k$. For player 0, $S^0 = S$ in every instance. Thus for $k = 1, 2, \ldots, B$, in $\mathcal{I}_k$ the set of honest players is $P_k \cup \{0\}$ and the set of good objects is exactly O_k. In every instance, the players in P_k view the world *as if the input instance is* $\mathcal{I}_k$. If $B < 1/\alpha$, then the players in $P_{B+1}, \ldots, P_{1/\alpha}$ simply don't ever report any result in any instance.

It is convenient to define a "null" input instance, denoted $\hat{\mathcal{I}}$, in which the players have the same probe value functions as in the instances $\mathcal{I}_k$, but the "true value" function is $S(i) = 0$ for every i. This instance has no good objects, so it violates the assumption that there are at least βm good objects, but we use it solely for the purpose of analyzing the other instances.

Let a deterministic algorithm A be given. Denote the execution of algorithm A over an instance $\mathcal{I}$ by $\mathcal{E}(\mathcal{I})$. Let the sequence of objects probed by player 0 in execution $\mathcal{E}(\hat{\mathcal{I}})$ be $o_1, o_2, \ldots, o_m$ (note that as instance $\hat{\mathcal{I}}$ contains no good objects, the probing will stop only after player 0 exhausts all objects). For $k = 1, \ldots, B$, let r_k be the smallest integer such that $o_{r_k} \in O_k$ (namely, $o_i \notin O_k$ for all $i < r_k$). By the construction, it can be verified (say, by induction on the round number) that for every $k = 1, 2, \ldots, B$, each player behaves identically in the two executions $\mathcal{E}(\hat{\mathcal{I}})$ and $\mathcal{E}(\mathcal{I}_k)$ for the first $r_k - 1$ rounds. Also note that in execution $\mathcal{E}(\mathcal{I}_k)$ (for $k = 1, 2, \ldots, B$), player 0 probes an object with score 1 for the first time on round r_k, and hence it incurs exactly r_k probes. Hence the expected number of probes incurred by player 0 in an instance taken from our distribution is $\frac{1}{B}\sum_{k=1}^{B} r_k$.

Finally, note that since $r_k \neq r_{k'}$ for $k \neq k'$, we have $\sum_{k=1}^{B} r_k > B^2/2$, and therefore the expected number of probes incurred by player 0 on this distribution is at least $B/2 = \Omega(\min\{\frac{1}{\alpha}, \frac{1}{\beta}\})$. □

Note that the trivial algorithm where each player probes a random object in each step (disregarding the billboard completely) will terminate in $O(1/\beta)$ expected time. When $1/\alpha$ is much smaller than $1/\beta$, as is the case when a large fraction of the players are honest, we might hope that these honest players could collaborate and do even better. As we have already mentioned, our prior algorithms [1, 2] halt in $O\left(\frac{\log n}{\alpha\beta n} + \frac{\log n}{\alpha}\right)$ expected rounds, which is $O\left(\frac{\log n}{\alpha}\right)$ when $m = n$. In the next section, we show that we can do better, and sometimes much better.

4 A Sublogarithmic Search Algorithm With Local Testing

This section presents Algorithm DISTILL for search with local testing that halts in a constant number of rounds when the number of honest players is large, always halts within $O\left(\frac{1}{\alpha} \cdot \frac{\log n}{\log\log n}\right)$ rounds when $m = n$. This result is stated precisely in Theorem 4.

Algorithm DISTILL is formally presented in Figure 1. The fundamental idea of the algorithm is to use only positive reports, and allow each player to make only one such report, called the player's *vote*. The algorithm maintains an explicit set of *candidate* (or potentially good) objects which is being reduced as the execution progresses. The algorithm consists of a series of calls to subroutine ATTEMPT. ATTEMPT first reduces the candidate set to a manageable size (Steps 1.1–1.2 take care of the case where $m \gg n$). Thereafter, in Step 2 ATTEMPT performs a succession of stages that is aimed to reduce the candidate set: In order for a current candidate object to be a member in the next set, it must receive sufficiently many votes *in this stage* (specifically, we use half the expected value as the threshold); candidates that do not receive sufficiently many votes in a stage drop from contention. We show that the number of stages in the reduction process of a single invocation of ATTEMPT can be bounded by a sublogarithmic function of n (times $1/\alpha$), and the probability of success in an invocation (defined to occur when most honest players have found a good object) is a positive constant. The latter holds provided that initially, the candidate set contains at least one good candidate and not too many bad candidates, which is guaranteed by Step 1 with positive constant probability. The algorithm has an additional wrinkle to ensure termination of all honest players. Call a player *satisfied* in a given state if at that state it has already found a good object and stopped probing (i.e., it has a vote). Consider the state when most honest players are already satisfied, and therefore don't vote any more. To guarantee quick termination of the remaining players, the algorithm stipulates that at every second step, each player makes a probe that follows a recommendation of a randomly chosen player. This is done by always probing using Subroutine PROBE&SEEKADVICE.

To facilitate the analysis of DISTILL, we introduce the following notation.

Notation 3 *Given $0 < \alpha < 1$ and n, define*

$$\Delta \stackrel{\text{def}}{=} \log\left(\frac{1}{1-\alpha} + \log n\right).$$

Theorem 4 *For some constants k_1 and k_2, the expected termination time of each player under Algorithm* DISTILL *is $O\left(\frac{1}{\alpha\beta n} + \frac{1}{\alpha} \cdot \frac{\log n}{\Delta}\right)$ for any adaptive Byzantine adversary.*

Before proving Theorem 4, let us interpret the expression for the expected time. For $m = n$ and $\beta > 0$, Theorem 4 says that the expected time complexity is never more than $O\left(\frac{1}{\alpha} \cdot \frac{\log n}{\log\log n}\right)$. If the fraction of dishonest players is small, the complexity is significantly better: it may even be independent of n, as the following corollary states.

Corollary 5 *If $m = n$ and $\alpha \geq 1 - \frac{1}{n^\epsilon}$ for some $\epsilon > \frac{1}{\log n}$, then the expected termination time is $O(1/\epsilon)$.*

To prove Theorem 4, we first state the high level argument, and then delve into the details in the ensuing lemmas.

Proof: Consider termination first. In Lemma 6 below we show that once there are more than $\alpha n/2$ satisfied honest players, each remaining unsatisfied player finds a good object after $O\left(\frac{1}{\alpha}\right)$ additional expected rounds. So let us assume from now on that the number of unsatisfied honest players is at least $\alpha n/2$, and analyze the total number of rounds. We claim that each invocation of ATTEMPT takes $O\left(\frac{k_1}{\alpha\beta n} + \frac{k_2}{\alpha} + \frac{\log n}{\alpha\Delta}\right)$ rounds, and that each such invocation succeeds with probability at least $1 - e^{-k_1/2} - e^{-k_2/16} - 9e^{-k_2/64}$. By the code, Step 1

```
Shared Variables (computable from the shared billboard data):
• $\ell_t(i)$: The number of votes object $i$ receives *in iteration* $t$ of Step 2.

Subroutine ATTEMPT:
        Prepare initial candidate set
1.1     for $k_1/\alpha\beta n$ times do invoke subroutine PROBE&SEEKADVICE($\{1,\ldots,m\}$).
1.2     Let $S$ be the set of objects with at least one vote.
1.3     for $k_2/\alpha$ times do invoke subroutine PROBE&SEEKADVICE($S$).
1.4     Let $C_0$ be all objects that got at least $k_2/4$ votes at Step 1.3;
1.5     Let $t \leftarrow 0$ ($t$ is a running iteration index). Denote $c_t = |C_t|$.
        Distill candidate set
2       while $c_t > 0$ do
2.1         for $1/\alpha$ times do invoke subroutine PROBE&SEEKADVICE($C_t$).
2.2         $C_{t+1} \leftarrow \left\{ i \in C_t \mid \ell_t(i) > \frac{n}{4c_t} \right\}$; $t \leftarrow t+1$.

Subroutine PROBE&SEEKADVICE($S$):
        Pick a random object from the set $S$ and probe it.
        Pick a random player $j$, and probe the object $j$ votes for, if exists.

Main algorithm:
        Invoke subroutine ATTEMPT repeatedly until done.

Termination:
        Whenever a good object is probed, post result on billboard and halt; that
            object is the probing player's vote.
```

Figure 1. *Algorithm* DISTILL. *The parameters k_1 and k_2 are determined later.*

takes $O\left(\frac{k_1}{\alpha\beta n} + \frac{k_2}{\alpha}\right)$ rounds; Lemma 7 shows that the number of iterations of the **while** loop is $O\left(\frac{\log n}{\Delta}\right)$, and by the code for Step 2.1, each such iteration takes $O\left(\frac{1}{\alpha}\right)$ rounds. Lemma 10 below shows that Step 2 succeeds with probability at least $1 - 9e^{-k_2/64}$ if C_0 contains a good object, and Lemma 8 shows that this condition is satisfied by Step 1 with probability at least $1 - e^{-k_1/2} - e^{-k_2/16}$. Hence, for any $k_1 \geq 1$ and $k_2 \geq 192$, say, the expected number of invocations of ATTEMPT is at most 5, and the theorem follows. □

The following lemma establishes the termination condition.

Lemma 6 *If at some state there are at least $\alpha n/2$ satisfied players, then any unsatisfied player will find a good object in $4/\alpha$ additional expected rounds.*

Proof: Consider a state in which at least $\alpha n/2$ honest players are satisfied. Then there are at least $\alpha n/2$ votes for good objects, and hence the expected number of times a player probes an object following the vote of a randomly chosen player until it probes a good object is at most $2/\alpha$. The lemma follows from the fact that since all probes are done via PROBE&SEEKADVICE, every second probe follows a vote of a randomly chosen player. □

Since the algorithm terminates quickly once half the honest players are satisfied, we need only analyze how long it takes for this many honest players to become satisfied. The key to the analysis is that each player has only one vote. This property allows us to bound the total number of probes done in Step 2. We remark that it is relatively simple to show a weaker $O(\log n)$ bound on the number of iterations of the **while** loop in the special case of large α values; the following key lemma proves a tighter and more general bound.

Lemma 7 *If there are at least $\alpha n/2$ unsatisfied players, then each invocation of* ATTEMPT *contains $O\left(\frac{\log n}{\Delta}\right)$ expected iterations of the* **while** *loop.*

Proof: Consider all iterations of the **while** loop within a single invocation of ATTEMPT. The number of votes cast in iteration t that are required for an object in C_{t-1} to be a member in C_t is, by Step 2.2, at least $\frac{n}{4c_{t-1}}$. Let b_t denote the number of bad objects which are still candidates in C_t.

Using this notation, the number of votes cast by dishonest players in iteration t is at least $b_t \cdot \frac{n}{4c_{t-1}}$. Since the total number of votes by dishonest players throughout the execution of the algorithm is at most $(1-\alpha)n$, summing over all rounds we have that

$$\sum_t b_t \frac{n}{4c_{t-1}} \leq (1-\alpha)n \,. \tag{1}$$

Let $\mathcal{T}^* = \{t \mid c_t > 0\}$, i.e., $\mathcal{T}^*$ is the set of iterations in which the **while** loop takes place. Let $T^* = |\mathcal{T}^*|$. To prove the lemma we bound T^* as follows. We divide $\mathcal{T}^*$ into two sets:

$$\mathcal{T} = \{t \in \mathcal{T}^* \mid b_t > c_t/2\}\,, \quad \mathcal{T}' = \{t \in \mathcal{T}^* \mid b_t \leq c_t/2\}\,.$$

First, note that after two expected iterations of $\mathcal{T}'$, the number of unsatisfied players drops below $\alpha n/2$; this is true because if $b_t \leq c_t/2$ in any round t, then the probability that at least half of the unsatisfied players probe a good object at that round is at least $1/2$. Thus, it is sufficient to bound $T \stackrel{\text{def}}{=} |\mathcal{T}|$. Simplifying Equation 1 and using the definition of $\mathcal{T}$, we get

$$\sum_{t\in\mathcal{T}} \frac{c_t}{c_{t-1}} \leq \sum_{t\in\mathcal{T}} \frac{2b_t}{c_{t-1}} \leq 8(1-\alpha)\,. \tag{2}$$

We use Equation 2 to obtain an upper bound on T as follows. By Step 2.2 of the code, $c_t/c_{t-1} \leq 1$ for all t; By Step 1.4 of the code, $c_0 \leq n$; also by definition, $c_{T^*} \geq 1$. Hence we have that

$$\prod_{t\in\mathcal{T}} \frac{c_t}{c_{t-1}} \geq \prod_{t\in\mathcal{T}^*} \frac{c_t}{c_{t-1}} = \prod_{t=1}^{T^*} \frac{c_t}{c_{t-1}} = \frac{c_{T^*}}{c_0} \geq \frac{1}{n}\,. \tag{3}$$

We now distinguish between two cases. The first case is that $\alpha < 1 - \frac{1}{\log n}$, in which $\Delta = \Theta(\log\log n)$. In this case we apply the Means Inequality to Equation 2, and by plugging in Equation 3 we get that

$$\sum_{t\in\mathcal{T}} \frac{c_t}{c_{t-1}} \geq T\left(\prod_{t\in\mathcal{T}} \frac{c_t}{c_{t-1}}\right)^{1/T} \geq T\left(\frac{1}{n}\right)^{1/T}.$$

Combined with Equation 2, we have $T\left(\frac{1}{n}\right)^{1/T} \leq 8(1-\alpha)$, or $\left(\frac{T}{8(1-\alpha)}\right)^T \leq n$. For $\alpha < 1 - \frac{1}{\log n}$, this solves to $T \leq O\left(\frac{\log n}{\log\frac{\log n}{1-\alpha}}\right) = O\left(\frac{\log n}{\log\log n}\right) = O\left(\frac{\log n}{\Delta}\right)$.

It remains to consider the case $\alpha \geq 1 - \frac{1}{\log n}$, in which $\Delta = \Theta(\log\frac{1}{1-\alpha})$. In this case, $8(1-\alpha) < 1$ for large n values. Since Equation 2 trivially implies that $c_t/c_{t-1} \leq 8(1-\alpha)$ for all $0 \leq t \leq T$, it follows from Equation 3 that $(8(1-\alpha))^T \geq 1/n$, and therefore $T = O\left(\frac{\log n}{\log\frac{1}{1-\alpha}}\right) = O\left(\frac{\log n}{\Delta}\right)$. □

We now turn to analyze the success probability in Step 1 of ATTEMPT.

Lemma 8 *If there are at least $\alpha n/2$ unsatisfied honest players, then the probability that C_0 contains a good object is at least $1-(e^{-k_1/2}+e^{-k_2/16})$.*

Proof: The total number of probes made by honest players at Step 1.1 is at least $\frac{\alpha n}{2}\cdot\frac{k_1}{\alpha\beta n} = \frac{k_1}{2\beta}$, and each of these probes hits a good object with probability β. Thus, the probability that no good object is probed by an honest player in Step 1.1 is at most $(1-\beta)^{k_1/2\beta} < e^{-k_1/2}$ for $\beta > 0$. Next, let us compute the probability that a good object i_0 is in C_0, given that some player voted for i_0 in Step 1.1. In each round of Step 1.3, the expected number of votes i_0 receives is at least $\alpha/2$, since there are at least $\alpha n/2$ honest players probing, and at most n objects were voted for in Step 1.1. Hence the expected total number of votes i_0 gets in Step 1.3 is at least $\frac{\alpha}{2}\cdot\frac{k_2}{\alpha} = \frac{k_2}{2}$. Hence, by Step 1.4 of the code, i_0 will be in C_0 if it gets at least half of the number of votes it expects to get in Step 1.3. The result follows, since by the Chernoff bound we have

$$\begin{aligned}
&\mathbf{P}\left[i_0 \notin C_0 \mid i_0 \text{ marked in Step 1.1}\right] \\
&\quad< \mathbf{P}\left[\ell_0(i_0) < \frac{1}{2}\mathbf{E}\left[\ell_0(i_0) \mid i_0 \text{ marked in Step 1.1}\right]\right] \\
&\quad< e^{\frac{1}{8}\mathbf{E}\left[\ell_0(i_0)\mid i_0 \text{ marked in Step 1.1}\right]} \leq e^{-k_2/16}\,.
\end{aligned}$$

□

Using the following technical lemma, we bound the success probability in Step 2 of ATTEMPT.

Lemma 9 *Given a sequence $\sigma = \{c_0, c_1, c_2, \ldots c_T\}$ of positive integers, and a constant $0 < a < 1$, let us denote*

$$f(\sigma) = \sum_{t=1}^{T} \frac{c_t}{c_{t-1}} \text{ and } g_a(\sigma) = \sum_{t=0}^{T} a^{1/c_t}.$$

Then for all sequences σ of non-increasing positive integers, $g_a(\sigma) \leq (\lceil f(\sigma)\rceil + 1)a^{1/c_0}$.

Proof: Fix a first element $c_0 > 0$. Let $\sigma^* = \{c_0, c_1, c_2, \ldots c_T\}$ be the sequence maximizing $g_a(\sigma)$ among all non-increasing sequences that start with c_0 and satisfy $f(\sigma) \leq B$ for some given constant B. Define $r_t = \frac{c_t}{c_{t-1}}$ for any $0 < t \leq T$, and consider the sequence of ratios $\rho = \{r_t\}_{t=1}^{T}$. We first claim that ρ is non-increasing: For suppose not. Then there exists a t_0 such that

$r_{t_0} < r_{t_0+1}$. Consider the sequence $\sigma' = \{c'_0, c'_1, \ldots, c'_T\}$ obtained from σ^* be setting

$$c'_t = \begin{cases} c_t & \text{if } t \neq t_0 \ , \\ c_{t_0-1} \cdot r_{t_0} & \text{if } t = t_0 \ . \end{cases}$$

It is easy to see that $f(\sigma') = f(\sigma)$ (we just transposed r_{t_0} with r_{t_0+1}). However, since $a^{1/x}$ is a monotonically increasing function of x for $a < 1$, and since $c'_{t_0} > c_{t_0}$ by construction, we have that $g_a(\sigma') > g_a(\sigma)$, contradicting the maximality of σ^*. After establishing the monotonicity of the ratio sequence ρ, we next claim that the following stronger property holds:

Claim A. *In σ^*, we have $c_0 = c_1 = \ldots = c_{\lfloor B \rfloor}$. If B is integer, then there are no more elements. Otherwise, the last element is $c_{\lfloor B \rfloor+1} = c_0/(B - \lfloor B \rfloor) < c_0$.*

Note that Claim A implies the lemma: if B is not integer, then $T = \lfloor B \rfloor + 2 = \lceil B \rceil + 1$; and if B is integer, then $T = \lfloor B \rfloor + 1 = \lceil B \rceil + 1$. In both cases, $a^{1/c_t} \leq a^{1/c_0}$ for all $t \leq T$.

We prove Claim A by contradiction: Assume the claim is false. Then it must be the case that

$$c_{T-2} > c_{T-1} > c_T > 0 \ . \tag{4}$$

Consider the sequence $\sigma'' = \{c''_0, c''_1, c''_2, \ldots c''_T\}$ defined by

$$c''_t = \begin{cases} c_t & \text{if } t \leq T-2 \ , \\ c_{T-1}+1 & \text{if } t = T-1 \ , \\ c_T - 1 & \text{if } t \leq T \ . \end{cases}$$

Note that possibly, $c''_T = 0$: in this case we have that $\frac{c''_T}{c''_{T-1}} = 0$, and we abuse notation slightly and use the convention that $a^{1/c''_T} = 0$ (which is consistent with the definition of g_a). We complete the proof by showing that σ'' has all the required properties and that $g_a(\sigma'') > g_a(\sigma^*)$, contradicting the maximality of σ^*. First, note that σ'' is non-increasing by Equation 4. Also we have by Equation 4 that

$$\begin{aligned} f(\sigma'') &= \sum_{t=1}^{T} \frac{c''_t}{c''_{t-1}} \\ &= f(\sigma^*) - \left(\frac{c_{T-1}}{c_{T-2}} + \frac{c_T}{c_{T-1}} \right) \\ &\qquad + \left(\frac{c_{T-1}+1}{c_{T-2}} + \frac{c_T - 1}{c_{T-1}+1} \right) \\ &= f(\sigma^*) + \left(\frac{c_{T-1}+1}{c_{T-2}} - \frac{c_{T-1}}{c_{T-2}} \right) \\ &\qquad - \left(\frac{c_T}{c_{T-1}} - \frac{c_T - 1}{c_{T-1}+1} \right) \\ &< f(\sigma^*) + \frac{1}{c_{T-2}} - \frac{1}{c_{T-1}} \\ &< f(\sigma^*) \leq B \ . \end{aligned}$$

Finally, we prove that $g_a(\sigma'') > g_a(\sigma^*)$, by noting that

$$\begin{aligned} & g_a(\sigma'') - g_a(\sigma^*) \\ &= \left(a^{1/(c_{T-1}+1)} + a^{1/(c_T-1)} \right) - \left(a^{1/c_{T-1}} + a^{1/c_T} \right) \\ &= \left(a^{1/(c_{T-1}+1)} - a^{1/c_{T-1}} \right) - \left(a^{1/c_T} - a^{1/(c_T-1)} \right) \\ &> 0 \ , \end{aligned}$$

where the final inequality follows because the second derivative of $a^{1/x}$ (as a function of x) is negative for $0 < a < 1$ and $x > 0$. $\square$

Lemma 10 *If there are at least $\alpha n/2$ honest unsatisfied players throughout the execution of the* **while** *loop, then the probability that there is a good object in the final candidate set C_T is at least $1 - 9e^{-k_2/64}$.*

Proof: Let i_0 be a good object. We first estimate $\mathbf{P}\left[i_0 \notin C_{t+1} \mid i_0 \in C_t\right]$. Summing these probabilities over all iterations will give us an upper bound on the probability that i_0 survives all iterations, given that $i_0 \in C_0$. So consider an iteration t, and assume $i_0 \in C_t$. By Step 2.2, and since there are at least $\alpha n/2$ unsatisfied players, we have that $\mathbf{E}\left[\ell_t(i_0)\right] \geq \frac{\alpha n}{2} \cdot \frac{1}{\alpha} \cdot \frac{1}{c_t} = \frac{n}{2c_t}$. By Chernoff bound, we have

$$\begin{aligned} \mathbf{P}\left[i_0 \notin C_{t+1} \mid i_0 \in C_t\right] &= \mathbf{P}\left[\ell_t(i_0) < \frac{1}{2}\mathbf{E}\left[\ell_t(i_0)\right]\right] \\ &< e^{-\mathbf{E}[\ell_t(i_0)]/8} \leq e^{-n/16c_t} \ . \end{aligned}$$

Thus, $\mathbf{P}\left[i_0 \notin C_T \mid i_0 \in C_0\right] \leq \sum_{t=0}^{T-1} e^{-n/16c_t}$ for any $T > 0$. Now apply Lemma 9. By Step 2.3, $C_t \subseteq C_{t-1}$ always and hence the sequence $\sigma = \{c_t\}$ is non-increasing; by Equation 2, $f(\sigma) \leq 8(1-\alpha)$ and hence $f(\sigma) < 8$ for $\alpha > 0$; and by Step 1.4, $c_0 \leq \frac{4n}{k_2}$. Therefore, taking $a = e^{-n/16}$ in Lemma 9,

$$\begin{aligned} \mathbf{P}\left[i_0 \notin C_T \mid i_0 \in C_0\right] &\leq \sum_{t=0}^{T-1} e^{\frac{-n}{16c_t}} \\ &\leq (\lceil 8(1-\alpha) \rceil + 1) \cdot e^{\frac{-n}{16c_0}} \\ &\leq 9 \cdot e^{-k_2/64} \ . \end{aligned}$$

$\square$

4.1 Multiple votes and erroneous votes

Our analysis makes heavy use of the fact that each player is allowed to submit a positive vote for only one object. This bounds the amount of damage a dishonest player can do by voting for bad objects. There is nothing special about the number 1, however, and we can allow each player to submit positive votes for up to f objects. Our analysis also makes heavy use of the fact that votes by honest players are correct, but it is reasonable to expect that even honest players will submit erroneous votes by mistake from time to time. We can tolerate incorrect votes by an honest player as long as one of its positive votes is correct. With both of these extensions, it is not difficult to see that the asymptotic result of Theorem 4 remains unchanged so long as $f = o(\frac{1}{1-\alpha})$.

5 High Probability Algorithm And Its Applications

Algorithm DISTILL, with $k_1 = O(1)$ and $k_2 = O(1)$, ensures that the expected termination time for each player is $O(\log n)$ (assuming, say, that $\alpha = \Omega(1/\log\log n)$ and that $\beta = \Omega(1/\alpha n \log n)$). If we are interested in a bound on the time in which the *last* player terminates, naïve application of the analysis of Section 4 yields a bound of $O(\log^2 n)$ under the same setting for α and β. In fact, we can do much better. Let DISTILL$^{\mathsf{HP}}$ (where HP stands for high-probability) denote a variant of Algorithm DISTILL with $k_1 = \Theta(\log n)$ and $k_2 = \Theta(\log n)$. We now have the following simple observation.

Theorem 11 *Algorithm DISTILL$^{\mathsf{HP}}$ terminates in $O\left(\frac{\log n}{\alpha\beta n} + \frac{\log n}{\alpha}\right)$ rounds with probability $1 - n^{-\Omega(1)}$ for any adaptive Byzantine adversary.*

Proof: (Sketch) The arguments used to prove Theorem 4 still apply, showing that the probability of failure in a single iteration is $n^{-\Omega(1)}$ if k_1 and k_2 are $\Omega(\log n)$. Lemma 6 is replaced by a lemma that shows that once the majority of honest players are satisfied, all other players terminate in $O\left(\frac{\log n}{\alpha}\right)$ additional rounds with probability $1 - n^{-\Omega(1)}$. For Lemma 7, we note that the argument works as is, augmented by the observation that throughout the execution of the algorithm, the total number of rounds in which there are less than $\alpha n/2$ satisfied honest players and the good objects constitute a majority of the current candidate set, is $O\left(\frac{\log n}{\alpha}\right)$ with high probability. Lemma 8 and Lemma 10 hold verbatim (Lemma 9 is independent of k_1, k_2). □

5.1 Guessing α

One disadvantage of the algorithm is that α is hardwired in the code. This can be overcome by the standard technique of doubling (in our case, halving), applied in conjunction with the high-probability algorithm described above. Specifically, set k_1, k_2 so as to ensure that the algorithm terminates in $k_3 \frac{\log n}{\alpha}(\frac{1}{\beta n} + 1)$ rounds with probability at least $1 - n^{-2}$. Note that k_1 and k_2 are independent of α. The existence of such a constant is guaranteed by Theorem 11.

Now, for $i = 0, 1, 2, \ldots \log n$, we run this algorithm exactly $2^i k_3 \log n(\frac{1}{\beta n} + 1)$ rounds, where in the ith run we set $\alpha \leftarrow 2^{-i}$ in the code. Let α_0 be such that there are $\alpha_0 n$ honest players. Clearly, once $i = \log(1/\alpha_0)$, the algorithm will succeed: the only "after effects" from previous executions will be that some honest players may be satisfied, and some dishonest votes were possibly cast. As for the time complexity, it is obvious that the overall time complexity is at most twice the time complexity of the last iteration, i.e., all honest players will terminate with high probability after $O\left(\frac{\log n}{\alpha_0 \beta n} + \frac{\log n}{\alpha_0}\right)$ rounds.

5.2 Multiple Costs

Algorithm DISTILL minimizes the time complexity, which implicitly means that it works under the unit cost model. If we apply algorithm DISTILL in the general cost model as is, the expected cost per player may be very high with respect to the best solution, because the algorithm may probe very expensive objects even if there is a cheap good object. However, letting q_0 denote the cost of the cheapest good object, where w.l.o.g. the minimal object cost is 1, we have the following using standard techniques.

Theorem 12 *There is an algorithm such that each honest player finds a good object with probability $1 - n^{-\Omega(1)}$ while paying only $O\left(q_0 \frac{m \log n}{\alpha n}\right)$ for any Byzantine adaptive adversary.*

Proof: Aggregate objects with similar cost in *cost classes*, where class i contains all objects whose cost is in the range $[2^i, 2^{i+1})$ (assuming, without loss of generality, that all costs are at least 1). We execute a series of instances of algorithm DISTILL$^{\mathsf{HP}}$: First, we run the algorithm only on objects of class 0; then we run the algorithm on objects of class 1 and so on. Each instance is run under the minimal assumption that there is only one good object in class i, i.e., in version i, we have $\beta = 1/m_i$, where m_i denotes the number of objects in class i. We claim that the cost incurred to an honest player by this iterative algorithm is only a factor of $O(\log n/\alpha)$ times the cost of the cheapest good object (assuming $m = \Theta(n)$). To see that, let $i_0 = \log q_0$. Then the total cost to an honest player under this algorithm is at most

$$\begin{aligned}
&\sum_{i=0}^{i_0} 2^{i+1}\left(\frac{m_i \log n}{\alpha n} + \frac{\log n}{\alpha}\right) \\
&\quad = \frac{\log n}{\alpha} \sum_{i=0}^{i_0} 2^{i+1}\left(\frac{m_i}{n} + 1\right) \\
&\quad \le O\left(2^{i_0} \frac{m \log n}{\alpha n}\right) = O\left(q_0 \frac{m \log n}{\alpha n}\right) .
\end{aligned}$$

□

5.3 Search without local testing

As stated, Algorithm DISTILL relies on local testing. Nevertheless, it turns out that it straightforward to tweak Algorithm $\text{DISTILL}^{\mathsf{HP}}$ to solve the search problem without local testing. The idea is to still rely on the restriction that each player is allowed to have only one vote; in the current context, however, the vote of player j is the highest value object j has personally probed so far. Note that in contrast to the local testing case, the vote of a player can therefore change as the execution progresses. With this interpretation, we run the algorithm for a prescribed number of steps (which depends on β, assumed to be part of the input in this case). Applying Algorithm $\text{DISTILL}^{\mathsf{HP}}$, all players stop at the prescribed time; with extremely high probability, all honest players have found a good object. Thus we obtain the following.

Theorem 13 *There is an algorithm without local testing such that each honest player finds a good object with probability* $1-n^{-\Omega(1)}$ *in* $O\left(\frac{\log n}{\alpha\beta n}+\frac{\log n}{\alpha}\right)$ *rounds for any Byzantine adaptive adversary.*

6 Conclusion and open problems

In this paper we have studied the parallel time complexity of eBay-like problems. Our main result is that even in the presence of dishonest players, the honest players can halt in a constant number of rounds when there are many honest players, and can halt in $O(\log n/\log\log n)$ rounds even when there are not. Besides the obvious open problem of narrowing the gap between the upper and lower bounds, we believe that several that issues came up in the analysis deserve further study. To begin with, our algorithm uses only positive recommendations (“this object is good”), and flatly ignores bad recommendations (“that object is bad”). Can bad recommendations be used to close the gap between the upper and lower bounds? (Alternatively stated: “Is slander useless?”) Second, we have decoupled the objects from the players. What is the effect of associating each object with a player? Third, in market systems like eBay, the reputation of an object influences its cost: A seller with little positive reputation will make up for it by setting a low price to his object. What is the effect of incorporating feedback via pricing into the model? Finally, note that our model does not make use of any non-trivial notion of “trust.” It seems interesting to understand whether such a notion can be useful in our model.

References

[1] B. Awerbuch, B. Patt-Shamir, D. Peleg, and M. Tuttle. Collaboration of untrusting peers with changing interests. In *Proc. 5th ACM Conf. on Electronic Commerce (EC)*, pages 112–119, May 2004.

[2] B. Awerbuch, B. Patt-Shamir, D. Peleg, and M. Tuttle. Improved recommendation systems. In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, Jan. 2005. To appear.

[3] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proc. 33rd ACM Symp. on Theory of Computing (STOC)*, pages 619–626, 2001.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[5] J. F. Canny. Collaborative filtering with privacy. In *Proc. IEEE Symp. on Security and Privacy*, pages 45–57, 2002.

[6] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. 12th World Wide Web Conf. (WWW)*, 2003.

[7] J. Kleinberg. Finding authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[8] J. Kleinberg and M. Sandler. Convergent algorithms for collaborative filtering. In *Proc. 4th ACM Conf. on Electronic Commerce (EC)*, pages 1–10, 2003.

[9] P. Kollock. The production of trust in online markets. In S. R. Thye, M. W. Macy, H. Walker, and E. J. Lawler, editors, *Advances in Group Processes*, volume 16, pages 99–124. Elsevier Psychology, 1999.

[10] B. S. G. K. J. Konstan and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC)*, pages 158–167, 2000.

[11] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Recommendation systems: A probabilistic analysis. In *Proc. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 664–673, 1998.

[12] S. K. Lam and J. Ried. Shilling recommender systems for fun and profit. In *Proc. 13th Conf. on World Wide Web (WWW)*, pages 393–402, 2004.

[13] M. P. O’Mahony, N. J. Hurley, and G. C. M. Silvestre. Utility-based neighbourhood formation for efficient and robust collaborative filtering. In *Proc. 5th ACM Conf. on Electronic Commerce (EC)*, pages 260–261, 2004.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proc. 1994 ACM conf. on Computer Supported Cooperative Work*, pages 175–186, 1994.

[15] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation sytems. *Comm. of the ACM*, 43(12):45–48, Dec. 2000.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. 10th Int. Conf. on World Wide Web (WWW)*, pages 285–295, 2001.

[17] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proc. 25th Ann. Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’02)*, pages 253–260, 2002.

[18] A. C. Yao. Probabilistic computations: toward a unified measure of complexity. In *Proc. 17th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 222–227, 1977.