Improved Collaborative Filtering

Aviv Nisgav and Boaz Patt-Shamir*

School of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel.
{avivns,boaz}@eng.tau.ac.il

Abstract. We consider the interactive model of collaborative filtering, where each member of a given set of users has a grade for each object in a given set of objects. The users do not know the grades at start, but a user can *probe* any object, thereby learning her grade for that object directly. We describe reconstruction algorithms which generate good estimates of all user grades ("preference vectors") using only few probes. To this end, the outcomes of probes are posted on some public "billboard", allowing users to adopt results of probes executed by others. We give two new algorithms for this task under very general assumptions on user preferences: both improve the best known query complexity for reconstruction, and one improving resilience in the presence of many users with esoteric taste.

1 Introduction

The *collaborative filtering* (or *interactive recommender*) problem can be described as follows. We are given n users and m objects, and each user has a *preference vector*, which consists of a grade for each object. The grades are initially unknown to the system (possibly not even to the users), but each grade can be revealed by a *probe*. For example, the objects may be books and probing is asking a user for her grade of the book (the user may need to read the book!); or the objects may be personal preference queries, and probing is presenting a query to a user by the system. The goal of collaborative filtering is to compute some function of the user grades while minimizing the *probe complexity*, defined as the maximal number of grades any user is asked to report.

The strongest possible task of collaborative filtering is reconstructing all user grades: given all grades, one can compute any function of them. Therefore one of the main questions in collaborative filtering is how expensive is it to reconstruct all grades. A basic observation is that reconstructing "esoteric" preferences (preferences that are held by only few users) may require considerably more probes than reconstructing "mainstream" preferences (preferences shared by many users). We formalize this intuitive notion using two parameters as follows. Fix a metric over preference vectors (say, Hamming distance for binary preferences). Given a *popularity parameter* $0 < \alpha \leq 1$, and a *distance parameter* $D \geq 0$, we say that a preference vector v_i is (α, D)*-prevalent* if there are at least αn users whose preference vectors are at distance at most D from v_i. A user whose preference vector is (α, D)-prevalent is called an (α, D)-prevalent user.[1]

* Supported in part by the Israel Science Foundation (grant 1372/09) and by Israel Ministry of Science and Technology.

[1] Note that α,D can be traded-off: Fix a set of preference vectors. Given any popularity parameter $\alpha \leq 1$, one can determine the smallest possible D so that a given user is (α, D)-prevalent. Similarly, given a distance parameter $D \geq 0$, the largest possible α for a user is well-defined.

Reconstruction algorithms can be distinguished according to whether their probe complexity is dependent or independent of the distance parameter D. Algorithms whose complexity depends on D may be better for small values of D: The best such algorithm known to date is Algorithm SMALL_RADIUS by Alon et al. [1], whose query complexity is $O(\frac{D}{\alpha}\log n^{2.5})$ (here and below, we omit a scaling factor of $\lceil\frac{m}{n}\rceil$, applicable only when $m > n$). The algorithm reconstructs preferences of (α, D)-prevalent users with $O(D)$ errors. The best known algorithms with probe complexity independent of D for (α, D)-prevalent users are Algorithm LARGE_RADIUS (also from [1]) that guarantees $O(D/\alpha)$ errors in probe complexity $O(\log^{3.5} n/\alpha^2)$, and Algorithm CALCULATEPREFERENCES by Gilbert et al. [8], which improves the number of errors in LARGE_RADIUS to $O(D)$ at the same asymptotic complexity. However, Algorithm CALCULATEPREFERENCES suffers from an interesting weakness: it requires users tastes to be mostly homogeneous, in the sense that almost all users must be (α, D)-prevalent (the algorithm in [8] allows for $O(\alpha n)$ users to be Byzantine). More specifically, the algorithm may produce incorrect results if many esoteric (but honest) users are present. Recalling the real world, this seems to be a significant drawback: it is an accepted truth that in many contexts, as many as 40% of the users do not have "mainstream" taste (see, e.g., [9]). We note that Algorithms SMALL_RADIUS and LARGE_RADIUS do not require homogeneity: users which are not (α, D)-prevalent get unpredictable results, but (α, D)-prevalent users still get correct output.

Our contribution. In this paper we present algorithms that improve both the distance-dependent and distance-independent cases. In Section 3 we describe Algorithm **S**, that reconstructs the preferences of (α, D)-prevalent users with at most $O(D)$ errors, using at most $O(\frac{D}{\alpha}\log^2 n)$ probes per user. Algorithm **S** improves on the best known probe complexity (of Algorithm SMALL_RADIUS [1]), and moreover, it is much simpler. In Section 4 we describe our second result: Algorithm **A**, whose probe complexity is $O(\frac{1}{\alpha}\log^3 n)$, reconstructing the preferences of (α, D)-prevalent users with $O(D)$ errors. Algorithm **A** can work with non-homogeneous population (namely not all users must be (α, D)-prevalent), while still being able to bound the effect of Byzantine users.

Related work. Much research in collaborative filtering considers the following model. There is a large dataset that contains all past choices of users (be it purchase history, or, say, movie grades), and the goal is to predict the way a user would grade an object she did not examine yet. The problem with this approach is that it ignores the existence of feedback in the model: If the system indeed affects user choices, the dataset is biased toward objects recommended by the system, and does not necessarily reflect the "true" preference of the users.

This fundamental gap is bridged by the *interactive recommender system* model [7, 4] we use. In this model the system can observe the user's reaction to recommendations and act on it. More specifically, the system proposes an object to the user, and the user, in response, informs the system of her grade for that object. (The system may deduce user feedbacks by some noisy heuristic, e.g., did the user click the proposed link?) It is usually assumed that the system starts out with no knowledge at all about user grades.

In the non-interactive model, it is common to assume a linear generative model for user's grades and apply algebraic techniques such as principal component analysis [10] or singular value decomposition [15]. Papadimitriou et al. [14] and Azar et al. [5] rigorously prove conditions under which SVD is effective. Other generative user models that were considered include simple Markov chain models [11, 12], where users randomly select their "type," and each type is a probability distribution over the objects.

Drineas et al. [7] were the first to propose the interactive model, where the algorithm tells the users which products to probe and the results of the probes are fed back to the algorithm. In [4] it was shown that in this model, a user sharing his exact preference with at least α fraction of the users ($D = 0$ in our terms), can find a product he likes in $O(\frac{\log n}{\alpha})$ probes. In [2], Awerbuch et al. show that at the same probe complexity, it is possible to reconstruct all users preference. They also prove that probe complexity $\Omega(\frac{\log n}{\alpha})$ is necessary in this case. Alon et al. [1] give the first algorithms to reconstruct preferences of (α, D)-prevalent tastes for $D > 0$, as mentioned above.

The basic interactive model was extended in a few directions. Awerbuch et al. [3] study an asynchronous model, where an adversarial (oblivious) schedule determines which user probes next. Azar et al. [6] show how to extend algorithms for binary grades to work with non-binary (discrete or continuous) grades.

Organization. In Section 2 we define the model and some notation. In Section 3 we give our algorithm with probe complexity linear in D. In Section 4 we give our second algorithm, with probe complexity independent of D, that can withstand Byzantine adversaries. Some proofs are omitted from this extended abstract.

2 Preliminaries

We first formalize the problem to be solved.

Instances. A reconstruction problem instance consists of a set P of n *users*, a set O of m *objects*, and a binary *grade* $A_{i,j}$ for each user $i \in P$ and object $j \in O$. The collection of grades of a given user i is called user i's *preference vector* or *taste*, denoted by A_i.

Given a set of objects $O' \subseteq O$, the *distance* between two users i, i' w.r.t. O', denoted $\text{dist}_{O'}(A_i, A_{i'})$ is the number of objects in O' on which i and i' disagree. We usually omit the subscript O' when distance is taken w.r.t. all objects.

Given $0 < \alpha \leq 1$ and $D \geq 0$, a preference vector v is (α, D)-*prevalent* in a given instance if there are at least αn users whose taste is at distance at most D from v. We shall abuse notation slightly and say that a user is (α, D)-prevalent if his taste is (α, D)-prevalent. For a subset $B \subseteq P$ of the users, an instance is called (α, D, B)-*homogeneous* if all users not in B are (α, D)-prevalent.

Outputs. We are given an (α, D, B)-homogeneous instance, of which we know only the number of users n, the number of objects m, and the parameters α and D. For any user i the output is a vector $\hat{A}_i$, whose intended meaning is an estimate of A_i. Our algorithms are randomized, the output accuracy statement will hold with high probability, namely with probability $1 - n^{-\Omega(1)}$, when probability is taken with respect to the coin tosses of the algorithm. Note there is no requirement regarding the output of users in B.

Algorithms. We assume the following distributed computational model. Algorithms proceed in synchronous *rounds*, where in each round, the algorithm may receive, as input, at most one grade for each user. This action is called a *probe* of the user. We assume that the results of all probes are posted on a public "billboard," i.e., they are available to all users, and the algorithm run by user i may use the grades of all previous probes made by all users to determine (typically, in a randomized way) what object user i will probe next. The maximal number of probes any user is asked to execute in a run is the *probe complexity* of the algorithm.

Simple bounds on probe complexity. Note that it is trivial to solve the recommendation problem in $O(m)$ probe complexity, by letting each user probe all objects. On the other hand $\Omega(m/\alpha n)$ probe complexity is necessary to produce estimates with $O(D)$ errors in (α, D, B)-homogeneous instances. Informally, αn users contained in a ball of diameter D need to cover between them all m objects with probes, and hence the average number of probes per user cannot be less than $\Omega(m/\alpha n)$.

3 Algorithm S: Linear dependence on D

In this section we present our first result: an algorithm for reconstructing preferences whose probe complexity is linear in D and $1/\alpha$, for (α, D)-prevalent users. We assume that α and D are given parameters. (Alon et el. [1] explain how to remove this restriction in an "anytime" algorithm, at the cost of increasing the probe complexity by a logarithmic factor and the number of errors by a constant factor.)

The algorithm presented in this section improves on Algorithm SMALL_RADIUS from [1] in terms of query complexity, and it is considerably simpler. It uses, as a building block, a known algorithm, as detailed below.

Tool: Exact reconstruction. We use an algorithm denoted **E** (mnemonic for "exact"), that solves the recommendation problem for $(\alpha, 0, B)$-prevalent instances of n users and m objects, namely instances where each user in $P \setminus B$ is a member of a set of at least αn users, all with identical preferences. Algorithm **E** produces, at the cost of $T_{\mathbf{E}}(n, m, \alpha)$ probe complexity, the preference vector of every user in $P \setminus B$. Several implementations of **E** are known [2, 1]. We use the following result, adapted from [2].

Theorem 1. *There exists an algorithm* **E** *that for any* $(\alpha, 0, B)$*-homogeneous instance with* n *users and* m *objects solves the reconstruction problem with probability at least* $1 - n^{-c}$*, using probe complexity* $O(\lceil \frac{m}{n} \rceil \cdot \frac{\log n}{\alpha})$*, for any desired constant* $c > 0$.

We note that in our algorithms, the number of invocations of **E** is polynomial in n, so by the Union Bound, we may assume that w.h.p., all invocations of **E** are successful.

Algorithm description. Algorithm **S** (see pseudo-code in Alg. 1) is very simple: The object set is broken into a few random subsets, and Algorithm **E** is applied to each of them, with popularity parameter $\alpha/4$. Repeating this procedure K times with independent random partitions of the object set yields K estimates for each object; the algorithm output at a user is, for each object, the majority of the outputs of **E** for that object. As we shall see, Algorithm **S** fails with probability $\exp(-\Omega(K))$.

Analysis. For each user $i \notin B$, define $P(i) \overset{\text{def}}{=} \{i' \in P : \text{dist}(A_i, A_{i'}) \leq D\}$, namely the set of users whose preference vectors differ from i by at most D objects We shall distinguish between objects on which i has an "unusual" opinion with respect to $P(i)$, and other objects, on which i agrees with most users in $P(i)$. The first set cannot contain too many objects, and the second set can be quite reliably reconstructed using $P(i)$.

Algorithm 1 : $\mathbf{S}(P, O, \alpha, D)$. K is a confidence parameter, $c > 8$ is a constant

```
(1) for $k \leftarrow 1$ to $K$ do
    (1a) Partition $O$ randomly into $S = cD$ disjoint subsets $O = \bigcup_{s=1}^{S} O_s$: for each object $j \in O$, independently select $s \in \{1, \dots, S\}$ uniformly at random, and let $O_s \leftarrow O_s \cup \{j\}$.
    (1b) for $s \leftarrow 1$ to $S$ do
             Invoke $\mathbf{E}(P, O_s, \frac{\alpha}{4})$.        // all players, some objects, reduced popularity
         Let $C_{i,j}^k$ denote the output for object $j$ by user $i$, for all $j \in O$ and $i \in P$.
(2) Let $C_{i,j}$ be the majority of $\{C_{i,j}^k \mid k = 1, \dots, K\}$, for all $j \in O$.
    For each user $i$ output $C_{i,1}, \dots, C_{i,m}$.
```

Formally, we define, for each user $i \notin B$, the set of objects $O(i)$ to be the objects on which user i agrees with the majority of the users in $P(i)$, i.e.,

$$O(i) \stackrel{\text{def}}{=} \left\{ j \in O \ : \sum\nolimits_{i' \in P(i)} |A_{i,j} - A_{i',j}| \ < \ \frac{|P(i)|}{2} \right\}.$$

We first state a "Markov's Inequality"-type bound on $|O(i)|$ (proof is omitted).

Lemma 1. *Any user $i \notin B$ agrees with at least $1 - \delta$ of the users in $P(i)$ on at least $m - D/\delta$ objects.*

Next, we show that for any $j \in O(i)$, in each iteration k of Algorithm **S**, Algorithm **E** computes a correct estimate of $A_{i,j}$ in Step 1b with good probability.

Lemma 2. *For all $i \in P \setminus B$, $j \in O(i)$ and $1 \le k \le K$:* $\Pr[C_{i,j}^k = A_{i,j}] \ge 1 - \frac{4}{c}$.

Proof: Consider iteration k, and let $O_{s(j)}$ be the subset j belongs to in iteration k. Let $P_s(i)$ be the set of users that agree with i on all objects in O_s, i.e., $P_s(i) = \{i' \mid \text{dist}_{O_s}(i, i') = 0\}$. It suffices to prove that $|P_{s(j)}(i)| \ge \frac{\alpha n}{4}$, because this ensures that the preconditions to the invocation of **E** are met in iteration k, and thus the lemma follows from the correctness of **E**.

First we note that for any $\beta > 0$, as distances are non-negative integers:

$$\sum_{i' \in P(i)} \text{dist}_{O_s}(i, i') \le \beta \cdot |P(i)| \implies |P_s(i)| \ge (1 - \beta) \cdot |P(i)| \qquad (1)$$

We now turn to bound the probability that $\sum_{i' \in P(i)} \text{dist}_{O_{s(j)}}(i, i') < \frac{3}{4} \cdot |P(i)|$. By the assumption that $i \notin B$ we know that $\sum_{i' \in P(i)} \text{dist}_O(i, i') \le D \cdot |P(i)|$. Recalling that $\sum_{i' \in P(i)} \text{dist}_O(i, i') = \sum_{s=1}^{S} \sum_{i' \in P(i)} \text{dist}_{O_s}(i, i')$, we may deduce that $\sum_{i' \in P(i)} \text{dist}_{O_s}(i, i') > \frac{|P(i)|}{4}$ for at most $4D$ indices s.

Consider now the random variable $\sum_{i' \in P(i)} \text{dist}_{O_{s(j)} \setminus \{j\}}(i, i')$. It is independent of the grades of j. As $j \in O(i)$, it holds $\sum_{i' \in P(i)} |A_{i,j} - A_{i'j}| \le \frac{|P(i)|}{2}$, and hence

$$\Pr\left[\sum\nolimits_{i' \in P(i)} \text{dist}_{O_{s(j)}}(i, i') \le \frac{3|P(i)|}{4}\right] \ge \\ \Pr\left[\sum\nolimits_{i' \in P(i)} \text{dist}_{O_{s(j)} \setminus \{j\}}(i, i') \le \frac{|P(i)|}{4}\right] \ \ge \ \frac{S - 4D}{S} \ = \ 1 - \frac{4}{c},$$

and we are done by Equation 1 (using $\beta = \frac{3}{4}$) and the fact $|P(i)| \geq \alpha n$. ∎

In each iteration the probability of a wrong prediction is less than $1/2$, and repeating the procedure diminishes it, as stated in the following lemma (proof omitted).

Lemma 3. *For any user $i \in P \backslash B$ and object $j \in O(i)$:* $\Pr[C_{i,j} = A_{i,j}] = 1 - e^{-\Omega(K)}$.

Before we summarize the performance of **S**, we note that using Chernoff bound it is easy to see that whenever $D = o(m/\log n)$, for m sufficiently large and for any s, it holds $|O_s| = \Omega(m/D)$ with high probability. We can now derive our first main result.

Theorem 2. *Suppose that $D = o(m/\log n)$ and $K = \Theta(\log(m+n))$. With probability $1 - mne^{-\Omega(K)}$, Algorithm **S** predicts for each user $i \in P \backslash B$ its preference vector with less than $2D$ errors. Moreover, the probe complexity is $KcD \cdot T_{\boldsymbol{E}}\left(n, \left\lceil \frac{m}{(c-1)D} \right\rceil, \alpha/4\right) = O\left(\frac{1}{\alpha}\left\lceil \frac{m}{nD} \right\rceil \cdot D \log^2(m+n)\right)$.*

4 Complexity independent of D

In this section we present our second main result: an improved algorithm estimating the preference vectors of users in an (α, D, B)-homogeneous instance, with probe complexity independent of D. An interesting problem that arises in this algorithm is the possible influence of users without (α, D)-prevalent taste. In all algorithms, the output of these "esoteric" users is unpredictable; but in the context of algorithms whose complexity is independent of D, such users may cause (α, D)-prevalent users to err too (this is the case in [8]). This problem is exacerbated in the presence of malicious users, who may fabricate their preferences on-line so as to hurt as many users as possible. Our algorithm bounds the number of errors introduced by honest but esoteric users, and the number of users affected by adaptive malicious users.

Tool: Distinguishing dissimilar users. Our algorithm uses Algorithm **Sep** essentially introduced in [13]. Algorithm **Sep** returns a users partition where each part contains users of roughly similar taste. Based on [13], the following can be proved.

Theorem 3. *Let $\mathcal{S} = \{S_1, \ldots, S_k\}$ be the result of applying **Sep**(P, O, α). Then:*
(1) For all $i_1, i_2 \in P$ with $A_{i_1} = A_{i_2}$, there exists $S \in \mathcal{S}$ s.t. $\{i_1, i_2\} \subseteq S$.
(2) Let $S \in \mathcal{S}$ be such that $|S| \geq \alpha n$. For any $j \in O$, with probability $1 - n^{-\Omega(1)}$, all users in S, except for at most $\alpha|S|$ users, have the same opinion about j.
*(3) The probe complexity of Algorithm **Sep** is $O(\lceil \frac{m}{n} \rceil \frac{\log n}{\alpha})$.*

As before, we note that in our algorithms **Sep** is invoked only poly(n) times, and hence we shall assume w.h.p. that all invocations of **Sep** are successful.

Tool: Selecting the closest vector from a set. Another procedure we use is SELECT, which receives, as input, a collection V of preference vectors, and, when run by user i, outputs the vector in V which is about the closest to A_i, the preference vector of i. More precisely, in [1] the following result is proved.

Theorem 4. *Suppose user i executes SELECT(V). Then the return value $u \in V$, with probability $1 - n^{-\Omega(1)}$, satisfies $dist(u, A_i) = O(\min\{dist(v, A_i) \mid v \in V\})$. Procedure SELECT requires $O(|V|^2 \log n)$ probes by user i.*

```
Algorithm 2 : F($P, O, \alpha$)                                   c is a constant
Partition $O$ randomly into $S = 2c^2 \log n$ disjoint subsets $O = \bigcup_{s=1}^{S} O_s$: for each object
  $j \in O$, select $s \in \{1, \ldots, S\}$ uniformly at random, and let $O_s \leftarrow O_s \cup \{j\}$.
for $s \leftarrow 1$ to $S$ do
    invoke Sep($P, O_s, \frac{\alpha}{6}$).
end for
return a $|P| \times |P|$ symmetric matrix where entry $(i, i')$ is "close" if users $i, i'$ ended in
  different subsets in less than $2c \log n$ invocations of Sep, and "far" otherwise.
```

Algorithm description. The basic idea is as follows. First we take a random subset of the objects, thus reducing the expected distance parameter from D to $O(\log n)$. To this subset we apply a distance-dependent procedure. The result is used to identify (w.h.p.) similarity of users; once this relation is established, users can adopt probe results of other users that are known to have similar taste, without risking too many errors.

This idea would work if all users were (α, D)-prevalent. But the presence of many users with non-prevalent tastes may affect the results, as their distance-dependent result may be incorrect. Such users may be incorrectly identified as "similar", and their number may overwhelm the number of (α, D)-prevalent users. To deal with this, we distinguish between two cases of non-prevalent users: honest non-prevalent users (whose taste is not determined by the execution of the algorithm) may influence only a bounded number of objects; but dishonest users, who use "bait and switch" tactics of changing their preferences on-line (as a function of the random choices taken by the algorithm), may incur much greater damage. Nevertheless, employing techniques developed for graph coalitions, we can bound the number of prevalent users influenced.

More specifically, Algorithm **F** (see Alg. 2) is used to compute a similarity relation. Note that Algorithm **F** is reminiscent of Algorithm **S**, but it uses Algorithm **Sep** instead of **E**, and its output is binary ("close" or "far") for all pairs of users. Our top-level algorithm is Algorithm **A** (see Alg. 3). It first selects a sample Ψ^k of the objects, and sends it to Algorithm **F** (Step 5). The sample size is such that the distance parameter is reduced to $O(\log n)$, which is the distance threshold Algorithm **F** is designed for. This is repeated K times. The matrix outputs of **F** are used in Step 7 to construct a "proximity graph" that indicates which users have preferences (probably) close to theirs (we note that Gilbert et al. [8] apply a similar technique). Algorithm **A** outputs, for each user and object, the majority of opinions for that object by users in its neighborhood (Step 15). To facilitate this policy, the algorithm requires (Step 1) all users to query a sufficiently large random sample of the objects, so as to ensure that each object has sufficient coverage by "close" users. The algorithm mitigates the influence of dishonest users by repeated averaging with neighborhoods of different radii in the graph (Step 8).

Below, we analyze the case where all users are honest (this is the model used by Alon et al. [1]). The analysis of the case where some users may be dishonest (as considered in [8]) is omitted from this extended abstract.

Analysis. Here we assume all users are honest, i.e., follow the protocol and their preferences are oblivious to the unfolding execution of the algorithm. We analyze Algorithm **A** with parameter $R = 1$ ($R > 1$ is useful in the case of dishonest users). As men-

Algorithm 3 : $\mathbf{A}(P, O, \alpha, D, R)$ R is a parameter, $c > 16$ is a constant

```
1:  Each user probes each object $j$ independently with probability $c \cdot \frac{\log(m+n)}{\alpha n}$.
2:  Let $S_i$ be the set of objects probed by user $i$. Let $K = 2c \log n$.
3:  for $k \leftarrow 1$ to $K$ do
4:      Select a random set $\Psi^k \subseteq O$ of $\lceil \frac{cm \log n}{D} \rceil$ objects, for some integer constant $c$.
5:      Evaluate $\mathbf{F}(P, \Psi^k, \alpha)$.                // all users, some objects
6:  end for
7:  Define a graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = P$ and $\mathcal{E} = \{(i, i') \mid i, i'$ never marked "far" by $\mathbf{F}\}$
    (i.e., nodes correspond to users and edges connect "similar" users).
8:  for all $r \in \{0, \ldots, R-1\}$ do
9:      Let $I_i(r)$ be the set of nodes at distance at most $2^r$ from $i$ in $G$.
10:     For all $j \in O$, let $I_{i,j}(r) \overset{\text{def}}{=} \{i' \in I_i(r) : j \in S_{i'}\}$, i.e. members of $I_i(r)$ that probed $j$.
11:     for all $j \in O$ do
12:         if $j \in S_i$ then
13:             set $A_{i,j}(r)$ to $A_{i,j}$ as probed in Step 1.
14:         else
15:             set $A_{i,j}(r)$ to the majority of the set $\{A_{i',j}\}_{i' \in I_{i,j}(r)}$. Break ties arbitrarily.
16:         end if
17:     end for
18: end for
19: return SELECT $(\{A_i(0), \ldots, A_i(R-1)\})$                to user $i$
```

tioned above, adopting the majority's opinion (Step 15) raises a problem with regard to non-prevalent users, who may incorrectly be identified as close. Our goal is therefore to show that on most objects, user i agrees with the majority of $I_i(0)$ (namely the users marked as "close").

Notation. We shall use the following notation. As in Section 3, we use $P(i)$ to denote the set of users whose preferences differ from those of user i on at most D objects, and $O(i)$ to denote the objects on which user i agrees with a majority of the users in $P(i)$. In addition, we denote:

- I_i^k: users not marked as "far" after k iterations of $\mathbf{F}$. We define $I_i^0 \overset{\text{def}}{=} P$.
- B_i^k: users in I_i^k who disagree with user i on more than $17D$ objects.
- Q_i^k: objects on which user i disagrees with at least $|P(i)|/3$ users in B_i^k. Formally: $Q_i^k \overset{\text{def}}{=} \left\{ j \in O : \sum_{i' \in B_i^k} |A_{i,j} - A_{i',j}| \geq \frac{|P(i)|}{3} \right\}$.

We now turn to analyze the algorithm, focusing on a generic user i, who is (α, D)-prevalent. We start by considering members of $P(i)$ (proof is omitted).

Lemma 4. *With probability $1 - n^{-\Omega(1)}$, all members of $P(i)$ are neighbors of i in the proximity graph.*

Next, we consider the influence of the non-prevalent users. While the number of objects on which user i agrees with many "similar" users can be easily bounded as in Lemma 1, bounding the number of objects on which i disagrees with too many "dissimilar" users (sufficient to influence the output of i) is more challenging. We bound the number of such objects by showing that each invocation of $\mathbf{F}$ marks many "dissimilar" users as "far." First we state a technical lemma, analogous to Lemmas 1 and 2.

Lemma 5. *Fix an invocation of Algorithm **F**. There are at most $4c\log n$ subsets O_s for which $O_s \not\subseteq O(i)$. Moreover, there are at most $4c\log n$ subsets O_s for which $O_s \subseteq O(i)$ and user i doesn't agree with at least $|P(i)|/2$ users on all object in O_s.*

We can finally bound the objects dominated by users not in $P(i)$.

Lemma 6. *With high probability, $|Q_i^K| < 17D$.*

Proof: We show that if $|Q_i^{k-1}| \geq 17D$ then $|B_i^k| \leq (1 - \frac{1}{c})|B_i^{k-1}|$. The idea is that by Theorem 3, **Sep** does not assign most users in B_i^{k-1} to the same set as i, so whenever B_i^{k-1} is big enough, many of its users are marked as "far" in iteration k. We start by identifying, for each invocation of **F**, the object subsets on which the premise of Theorem 3(2) is fulfilled, and then use a counting argument.

Consider the kth iteration of Line 5. Let S' be the set of indices such that for any $s \in S'$ it holds (1) O_s contains at least one object from Q_i^{k-1}, and (2) i agrees with at least half the users in $P(i)$ on all objects in O_s. In other words, for $P_s(i) \overset{\text{def}}{=} \{i' \mid \text{dist}_{O_s}(i,i') = 0\}$, let $S' \overset{\text{def}}{=} \left\{s : O_s \cap Q_i^{k-1} \neq \emptyset \text{ and } |P_s(i)| \geq P(i)/2\right\}$.

Assume $|Q_i^{k-1}| \geq 17D$. Then $\mathbb{E}\left[|\Psi^k \cap Q_i^{k-1}|\right] = \left|Q_i^{k-1}\right|\frac{c\log n}{D} \geq 17c\log n$, and therefore, by the Chernoff Bound, w.h.p., there are at least $16c\log n$ object-subsets in **F** containing an object from Q_i^{k-1}. By Lemma 5 there are at most $8c\log n$ subsets in which user i doesn't agree with at least $P(i)/2$ other users, so we get $|S'| \geq 8c\log n$.

For every user b in B_i^{k-1}, let ϕ_b^k be the total number of **Sep** invocations at which user b was assigned to different subset than i at the k'th iteration of **F**. Consider $s \in S'$: On one hand, as $O_s \cap Q_i^{k-1} \neq \emptyset$, then by Q_i^k definition, there is an object $j \in O_s$ on which user i disagrees with at least $P(i)/3$ of the users in B_i^{k-1}. On the other hand, as $|P_s(i)| \geq \frac{P(i)}{2}$, by Theorem 3, at most $\alpha n/6$ of those users are assigned by **Sep** to the same partition as user i. By definition of Q_i^k, $|Q_i^{k-1}| < 17D$ whenever $|B_i^{k-1}| < \frac{|P(i)|}{3}$, so we can bound the number of times "dissimilar" users are assigned to different partition as i by using the assumption $\left|B_i^{k-1}\right| \geq \frac{|P(i)|}{3} \geq \frac{\alpha n}{3}$:

$$\sum_{b\in B_i^k} \phi_b^k = \sum_{s\in S}\left|\left\{b \in B_i^k \ : \begin{array}{l} b, i \text{ are assigned to differ-} \\ \text{ent subsets by } \textbf{Sep} \text{ on } O_s \end{array}\right\}\right|$$

$$\geq \sum_{s\in S'}\left(\left|B_i^{k-1}\right| - \frac{\alpha n}{6}\right) \ \geq\ |S'|\frac{\left|B_i^{k-1}\right|}{2} \ \geq\ 4c\log n\left|B_i^{k-1}\right|.$$

As Algorithm **F** marks every user with $\phi_b^k \geq 2c\log n$ as "far" and there are $2c^2\log n$ object-subsets, then out of the users in B_i^{k-1} at least $\frac{4c\log n|B_i^{k-1}| - 2c\log n|B_i^{k-1}|}{2c^2\log n} = \frac{1}{c}|B_i^{k-1}|$ users are marked as such, and hence $|B_i^k| \leq \left(1 - \frac{1}{c}\right)|B_i^{k-1}|$. Now, since $|Q_i^K| \geq 17D$ implies $|Q_i^k| \geq 17D$ for $k = 1, \ldots, K$, we may conclude that if $|Q_i^K| \geq 17D$ then $B_i^K \leq 1$, contradiction. ▌

We can now summarize the properties of Algorithm **A** for the case of honest users.

Theorem 5. *Algorithm **A**$(P, O, \alpha, D, 1)$ predicts for each (α, D)-prevalent user its preference vector with $O(D)$ errors, with probability $1 - n^{\Omega(c)}$. Moreover, the probe complexity is $O\left(\frac{1}{\alpha}\left\lceil\frac{m}{nD}\right\rceil \cdot \log^3(m+n)\right)$.*

We note that Theorem 5 improves on Algorithm CALCULATEPREFERENCES by Gilbert et al. [8] in the case of honest users, both in terms of probe complexity, and in terms of resilience to non-prevalent users.

References

1. N. Alon, B. Awerbuch, Y. Azar, and B. Patt-Shamir. Tell me who I am: an interactive recommendation system. In *Proc. 18th Ann. ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 1–10, 2006.
2. B. Awerbuch, Y. Azar, Z. Lotker, B. Patt-Shamir, and M. Tuttle. Collaborate with strangers to find own preferences. In *Proc. 17th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 263–269, 2005.
3. B. Awerbuch, A. Nisgav, and B. Patt-Shamir. Asynchronous active recommendation systems. In *Principles of distributed systems : 11th international conference (OPODIS 2007)*, volume 4878 of *LNCS*, pages 48–61, 2007.
4. B. Awerbuch, B. Patt-Shamir, D. Peleg, and M. Tuttle. Improved recommendation systems. In *Proc. 16th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1174–1183, 2005.
5. 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.
6. Y. Azar, A. Nisgav, and B. Patt-Shamir. Recommender systems with non-binary grades. In *Proc. 23rd Ann. ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, San Jose, CA, June 2011.
7. P. Drineas, I. Kerenidis, and P. Raghavan. Competitive recommendation systems. In *Proc. 34th ACM Symp. on Theory of Computing (STOC)*, pages 82–90, 2002.
8. S. Gilbert, R. Guerraoui, F. M. Rad, and M. Zadimoghaddam. Collaborative scoring with dishonest participants. In *Proc. 22nd Ann. ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 41–49, 2010.
9. S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *Proc. 3rd ACM Int. Conf. on Web Search and Data Mining (WSDM)*, pages 201–210, New York, NY, USA, 2010. ACM.
10. K. Goldberg, T. Roeder, D. Gupta, , and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval Journal*, 4(2):133–151, July 2001.
11. J. Kleinberg and M. Sandler. Convergent algorithms for collaborative filtering. In *Proc. 4th ACM Conf. on Electronic Commerce (EC)*, pages 1–10, 2003.
12. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Recommendation systems: A probabilistic analysis. In *Proc. 39th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 664–673, 1998.
13. A. Nisgav and B. Patt-Shamir. Finding similar users in social networks: extended abstract. In *Proc. 21st Ann. ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 169–177, 2009.
14. C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proc. 17th ACM Symp. on Principles of Database Systems (PODS)*, pages 159–168. ACM Press, 1998.
15. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC)*, pages 158–167. ACM Press, 2000.