Theoretical Computer Science 628 (2016) 40–49

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Comparison-based interactive collaborative filtering

Yuval Carmel, Boaz Patt-Shamir [*,1]

School of Electrical Engineering, Tel Aviv University, Tel Aviv 6997801, Israel

ARTICLE INFO

Article history:
Received 7 September 2015
Received in revised form 20 January 2016
Accepted 7 March 2016
Available online 10 March 2016
Communicated by M. Latapy

Keywords:
Parallel algorithms
Recommender systems
Social networks
Collaborative filtering

ABSTRACT

In this work we study the interactive model of comparison-based collaborative filtering. Each *player* prefers one *object* from each pair of objects. However, revealing what is a player's preference between two objects can be done only by asking the player specifically about that pair, an action called *probing*. The goal is to (approximately) reconstruct the players' preferences with the smallest possible number of probes per player. The per-player number of probes can be reduced if there are many players who share a similar taste, but a priori, players do not know who to collaborate with. In this work, we present the model of comparison-based interactive collaborative filtering, analyze a few possible taste models and present distributed algorithms whose output is close to the best possible approximation to the players' taste.

1. Introduction

Recommendation systems have become a significant part of our lives in the past few years. Most people encounter recommendation systems on a daily basis, while buying a book, choosing which movie to watch, buying groceries in the supermarket, or even finding a life mate. Collaborative Filtering is one of the prevalent approaches to recommendation systems, especially large scale systems (such as Netflix [7]). The idea in collaborative filtering is to take advantage of the existence of many players with similar preferences which can collaborate by sharing the load of trying the objects and identifying objects they perceive as good.

Following [4,5,10,17,20], we distinguish between *interactive* and *non-interactive* recommendation systems, which differ in assumption and usage. In non-interactive recommendation systems, the algorithm is fed all known preferences as collected from the users in the past, and the goal is to output (possibly few) unknown preferences. This model is very popular, and conceptually easy to implement, but it does not take into account the dynamics of the system after the output is made.

In interactive recommendation systems (first introduced as "competitive" in [10]), on the other hand, while the goal remains to predict preferences, it is assumed that no preference is known a priori to the system or the users, but information about preferences can be found by asking the user to try some objects, in an action called *probe*. For example, a probe may ask the user to read a book and ask him[2] for his opinion on the book. The results of probes not only determine the predictions the algorithm makes, but also determine the identity of future objects to be probed. The goal of interactive algorithms is to predict players' preferences while minimizing the number of probes, since probing is assumed to be costly. In the interactive model, the system doesn't suffer from the cold start problem [14,21] since items are being recommended

* Corresponding author.
E-mail addresses: yuval9859@gmail.com (Y. Carmel), boaz@tau.ac.il (B. Patt-Shamir).

1 Supported in part by a grant from the Israel Ministry of Science, Technology and Space and the French Ministry of Higher Education and Research.

2 Or her. For uniformity, we have arbitrarily chosen to refer to players as males.

http://dx.doi.org/10.1016/j.tcs.2016.03.010
0304-3975/

to the users continuously throughout the running of the algorithm (we treat querying objects in the interactive model as recommending new items to the user while optimizing the next recommendations by the user's probe result).

The interactive model is as follows (cf. Section 2). There are n players and m objects; for each player, there are preferences over the objects, represented by his *preference vector* or *taste*. An entry in the vector of a player can be revealed only by asking that player to preform the appropriate probe. (The player is not assumed to know his preferences in advance, but to discover the result of each probe once it is made.) Probe results are assumed to be posted on a shared "billboard" (modeling eBay feedback records, IMDb reviews, etc.), so that each player can run his algorithm to find which probe to do next, as well as compute preference predictions.

The existence of a billboard does not solve the problem, since players still need to decide whose results to adopt. We assume that some tastes are popular, namely many players have them. Concretely, given a *popularity factor* $0 < \alpha \le 1$, and a *distance parameter* $D \ge 0$, we say that a preference vector v_j is (α, D)-*popular* if there are at least αn players whose preference vectors differ from v_j by at most D entries. Note that in order to reconstruct any taste (preference vector), $\Omega(m)$ probes are required just to cover all objects, and hence, to reconstruct his preferences to within $O(D)$ errors, the average number of probes per player with an (α, D)-popular taste cannot be less than $\Omega(\frac{m-D}{\alpha n})$.

In most previous work, preferences are simply an absolute grade for each object. The grades are usually binary or decimal with the interpretation of "the user like/dislikes the object a" or "the object a belongs to the user's top x objects" respectively. In this work we introduce a *comparison-based* interactive collaborative filtering model (talking about comparison-based non-interactive CF is out of context, since the comparison-based queries affect the users' ranking), where preferences are expressed only over pairs of objects, with the interpretation of "the user prefers object a over object b". In the interactive model, a comparison-based probe means that the user is presented with two objects, and responds with his preference between them (which may also include "equal" or "incomparable"). We note that it is well known that comparison-based preferences are more intuitive, consistent and accurate than absolute grading (we elaborate below). However, it is not quite clear what can be assumed about the structure of comparison-based preferences. In this work we study a few simple user models. The simplest model is that the user preference between a pair of objects is independent of his preferences over other pairs, and possibly the most structured model is when the pairwise preferences are induced by an underlying total order over all objects. In between, one can consider pairwise preference induced by partial order.

Our contribution The main technical contribution of this work is a comparison-based algorithm for reconstructing preferences induced by an underlying total order. First we present Algorithm $\mathcal{DP}$ (Section 3) for instances with distance parameter $D = 0$. With high probability, Algorithm $\mathcal{DP}$ reconstructs $(\alpha, 0)$-popular preference vectors exactly, incurring at most $O(\frac{1}{\alpha}\log n(\log\log n + \log\frac{1}{\alpha}))$ probes per player, assuming $m = n$.[3]

Our main result is Algorithm $\mathcal{DPD}$ (Section 4), that uses Algorithm $\mathcal{DP}$ as a subroutine, and solves problem instances with distance parameter $D > 0$ w.h.p. Algorithm $\mathcal{DPD}$ reconstructs (α, D)-popular preference vectors with at most $O(D)$ errors and using at most $O(\frac{D^2}{\alpha}\log^2 n \log\frac{\log n}{\alpha}))$ probes per player. We also consider the case where each user perceives the object set as a disjoint union of a few *object categories*, such that objects within a category are totally ordered, but objects in different classes are incomparable. This model is appropriate in the case that the object set is eclectic, e.g., cars and restaurants. We show how to reconstruct the taste in this case without prior knowledge on the categorizations, and even when different users have different categorizations.

Related work Collaborative filtering is studied quite intensively, but mostly from the non-interactive perspective (see, e.g., [7,11,18], which usually use different algorithms to predict users preference on objects based on previous data). The interactive model was introduced by Drineas et al. [10] (referred to there as "competitive recommendation systems") and developed by many over the years [4,5,17,20]. In the absolute grade model, where a preference vector specifies a grade for each object, it was shown by Awerbuch et al. [5], that all $(\alpha, 0)$-popular preferences can be reconstructed exactly, using $O(\frac{1}{\alpha}\log n)$ user probes (assuming for simplicity that the number of objects roughly equals the number of users, i.e., $m = \Theta(n)$). Alon et al. [4] extend this result with instances in which the distance parameter is $D > 0$ and provide a reconstruction for all (α, D)-popular users preferences using $O(\frac{1}{\alpha}D^{3/2}\log^2 n)$ user probes with at most $O(D)$ errors. In addition, [4] provides an algorithm which reconstructs all (α, D)-popular tastes with $O(D/\alpha)$ errors using $O(\log^{3.5} n/\alpha^2))$ user probes. Nisgav and Patt-Shamir [17] improve these results, presenting algorithms for reconstructing (α, D)-popular users preferences with $O(D)$ errors and probe complexity either $O(\frac{D}{\alpha}\log^2 n)$ or $O(\frac{1}{\alpha}\log^3 n)$.

Comparison-based recommendation systems are considered superior to absolute grading systems in stability and more natural for human interaction based ratings. For example, in an experiment held by Jones et al. [12], user preferences expressed by comparisons were measured as 20% more stable over time than preferences expressed by 5-star grading scale. Since a 10% increase in accuracy is considered significant (that was the goal of the million-dollar Netflix challenge [7]), a 20% reduction in inconsistency is clearly meaningful in this context. Moreover, users tend to prefer comparison-based grading, finding it more intuitive than 5-star rating-based grades [8,12].

It is therefore not surprising that there are some proposals for comparison-based model recommendation systems. Loepp et al. [15,16] provide an interactive collaborative filtering algorithm which uses a priori knowledge about the objects to op-

[3] In general, the probe complexity results should be multiplied by $\lceil m/n \rceil$. For simplicity, we usually omit this factor and assume $m = n$.