

Optimal Recommendations under Attraction, Aversion, and Social Influence

Wei Lu[†] Stratis Ioannidis[‡] Smriti Bhagat[‡] Laks V.S. Lakshmanan[†]

[†]University of British Columbia
Vancouver, B.C., Canada
{welu,laks}@cs.ubc.ca

[‡]Technicolor Los Altos Research Center
Los Altos, CA, USA
{stratis.ioannidis,smriti.bhagat}@technicolor.com

ABSTRACT

People’s interests are dynamically evolving, often affected by external factors such as trends promoted by the media or adopted by their friends. In this work, we model interest evolution through dynamic interest cascades: we consider a scenario where a user’s interests may be affected by (a) the interests of other users in her social circle, as well as (b) suggestions she receives from a recommender system. In the latter case, we model user reactions through either attraction or aversion towards past suggestions. We study this interest evolution process, and the utility accrued by recommendations, as a function of the system’s recommendation strategy. We show that, in steady state, the optimal strategy can be computed as the solution of a semi-definite program (SDP). Using datasets of user ratings, we provide evidence for the existence of aversion and attraction in real-life data, and show that our optimal strategy can lead to significantly improved recommendations over systems that ignore aversion and attraction.

Categories and Subject Descriptors

H.2.8 [Database Applications]: Data Mining

Keywords

Recommender Systems; Interest Evolution; Attraction; Aversion

1. INTRODUCTION

Users’ content consumption patterns evolve over time. For example, a user may be attracted towards content that is popular, content recommended to her by a service, or content being enjoyed by her friends. Alternatively, users may get tired of certain types of content, e.g., romantic comedy movies, and desire to consume something different and new. A key challenge for recommender systems is accurately modeling such user preferences as they evolve over time. Although traditional matrix factorization approaches can be extended to incorporate temporal dynamics of user behavior [19,20], such extensions do not identify or explicitly analyze the *factors* that influence the drift in interests.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD’14, August 24–27, 2014, New York, NY, USA.
Copyright 2014 ACM 978-1-4503-2956-9/14/08 ...\$15.00.
<http://dx.doi.org/10.1145/2623330.2623744>.

A “classic” factor influencing user interests is *attraction*: users may be attracted to content they are exposed to repeatedly and often (such as, e.g., a song played often on the radio). This phenomenon, known in psychology as the “*mere-exposure effect*” [33], is natural and intuitive, and is the main premise behind advertising [10, 15]. Nonetheless, repetition and/or overexposure can also have the opposite effect, leading to *aversion*: recent research argues that users often desire serendipitous, novel, previously unseen content [1, 3, 23, 27]. This notion is also quite natural and intuitive, but is usually not taken into account by recommender systems, yielding over-specialized, predictable recommendations [3, 23].

A third factor affecting a user’s interests is *social influence*: users may feel attracted to content consumed and liked by their friends. Trend adoption through “word-of-mouth” or “viral” marketing is also a well documented phenomenon [7, 9], and has been extensively studied since the seminal paper by Kempe et al. [17]. Nonetheless, to the best of our knowledge, the effect of social influence on interests, and its implications for recommender systems, has received attention only recently [16, 27].

Incorporating these influence factors in a recommender system raises several challenges. To begin with, under attraction and aversion, a recommender can no longer be treated as a passive entity: recommendations it makes may alter user interests, pushing them either towards or away from certain topics. Hence, traditional methods that merely profile a user and then cater to this specific profile may fall short of keeping up with these dynamics. Second, social influence implies that recommendation decisions to different users cannot be made in isolation anymore: as recommendations alter a user’s interests through attraction and aversion, social influence can spread these changes, resulting in an *interest cascade*. Therefore, optimal recommendation decisions across users need to be computed *globally*, taking into account the joint effect they have over the user’s social network.

In this work, we make the following contributions:

- We formulate a global recommendation problem in the presence of attraction, aversion, and social influence. In particular, we propose a mathematical model that incorporates these phenomena, and study the steady state behavior of user interests as a function of the recommender’s strategy in selecting which items to show to users. Under this model, we seek the optimal recommendation strategy, i.e., one that maximizes the users’ social welfare in steady state (Section 3).
- We show that, for a large recommender item catalog, obtaining the optimal recommendation strategy amounts to solving a quadratically-constrained quadratic optimization problem (QCQP). Though this problem may not be convex, we present a semi-definite program (SDP) relaxation that can be solved in polynomial time. In many cases, this solution is also

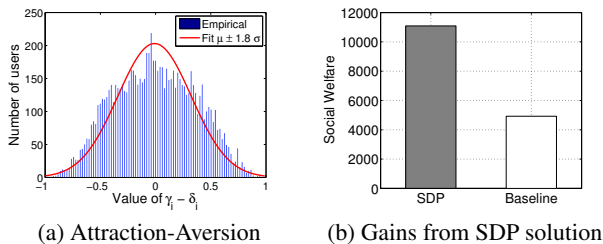


Figure 1: Illustration of aversion and attraction in MovieLens, and gains from accounting for them in optimization.

guaranteed to be an optimal solution; when the solution is not optimal, we show how a solution with a provable approximation guarantee can be constructed through randomization. We discuss how to determine whether the solution is optimal, and identify special cases for which an optimal solution is always reached, and randomization is unnecessary (Section 4).

- We provide evidence for the existence of attraction and aversion in three real-life rating datasets. We do so by developing and applying a method for learning the weight (i.e., importance) of these factors from rating data (Section 5). Applied to three real life datasets, our method indicates that between 14.0% to 19.6% of users show strong aversive or attractive behavior (Section 6).
- We conduct extensive experiments on real world datasets, and show that our recommendation algorithm is 17.5% to 125% better than a baseline algorithm in terms of social welfare achieved (Section 6).

Our analysis indicates that (a) the above phenomena are present in real-life datasets and (b) accounting for them can lead to significant gains in the improvement of recommendations. Figure 1 provides a quick illustration of these two facts (see Section 6 for a detailed account on the derivation these two figures). Figure 1(a) presents the distribution of a score measuring aversion and attraction among different users in MovieLens (with -1 indicating users with the strongest aversive behavior, and $+1$ indicating users with the strongest attractive behavior). About 7.0% of users are strongly aversive (score ≤ -0.5) while 9.0% are strongly attractive (score ≥ 0.5). Accounting for such users can lead to a significant impact on recommendations: as shown in Figure 1(b), the user social welfare more than doubles when incorporating this knowledge in recommendation decisions. Though there are clearly many factors of user behavior that are not accounted for in our analysis, we believe that these two facts, along with the SDP relaxation yielding optimal recommendations, indicate that investigating and accommodating for such phenomena is both important and tractable.

2. RELATED WORK

There has been a significant interest in modeling the temporal dynamics of user interests for various settings close to ours [19, 26, 27, 30]. Early work on matrix factorization (MF) by Koren [19] incorporates time-variant user profiles, an approach that we also adopt. We depart from this line of work by modeling, and also including in the MF process (see Section 5), factors that impact these drifts, including attraction, aversion, and social influence.

Several studies have highlighted the need for serendipity and diversity in the context of recommender systems, both of which relate to the notion of aversion we describe here. The need for serendipity was first identified by McNee et al. [23]. To address this, Yu et

al. [32] and Abbassi et al. [1] propose algorithms for recommending items that maximize a score that combines both relevance to a user as well as diversity. Ge et al. [12] focus on evaluating the lack of serendipity and diversity, and how it hurts the quality of recommendations. We depart from these works by modeling how recommendations themselves may instigate aversion or attraction among users, through a dynamic evolution of user interests.

Our approach to aversion is closer to Das Sarma et al. [27], who consider users that iteratively consume items in one out of several categories. They incorporate “boredom” and social influence in a manner similar to us: inherent item values decrease as a function of a weighted frequency of past consumption, and a user’s utility is averaged among her friend’s utilities. The authors provide bounds of the steady state performance of different consumption strategies under such dynamics. We depart by modelling user interests as multi-dimensional vectors, and using a factor-based model for user utilities, whose dynamics and steady state behavior cannot be captured by the (one-dimensional) model in [27].

The literature on social influence is vast, motivated by the viral marketing applications introduced by Domingos and Richardson [9] and further studied by Kempe et al. [17]. Our influence model is closer to *gossiping* [28], in that the interest/state of each user results from averaging the interests of her neighbors. Though we depart from classic gossiping protocols in that we incorporate additional dynamics (through attraction and aversion), similar techniques as in [28] could potentially be used to study our system in scenarios where interest evolution is asynchronous across users. In the context of matrix factorization, Jamali et al. [16] propose incorporating the distance of a user’s profile to the average profile of users in their social circle as a regularization factor in MF. This is consistent with the social influence behavior we outline in Section 3.3. We depart from this work by modeling dynamic profiles, and studying the additional effect of recommendations on user profiles through attraction and aversion.

Semi-definite programming (SDP) relaxation for quadratically-constrained quadratic programs (QCQP) lies at the core of our algorithmic contribution. Building on the seminal work by Goemans and Williamson [13], several papers have demonstrated classes of QCQPs for which an SDP relaxation gives a constant approximation guarantee [24, 25, 31]. Moreover, exact solutions of rank 1 are known to be attainable for several classes of QCQP, including when the problem has one [6] or two quadratic constraints [5]. Of special interest is the case where the quadratic objective involves non-negative off-diagonal elements, and constraints involve only quadratic terms of one variable [34], as the attraction-dominant case of our problem falls into this class (see Section 4.3). We refer the interested reader to [29] for SDP in general, and to [22, 25] for applications to quadratic programming.

3. PROBLEM FORMULATION

In what follows, we present our mathematical model of users interacting with a recommender system. We use bold script (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}$) to denote vectors, and capital script (e.g., A, B, H) to denote matrices. For either matrices or vectors, we use \geq to indicate element-wise inequalities. For symmetric matrices, we use \succeq to indicate dominance in the positive semidefinite sense; in particular, $A \succeq 0$ implies that A is positive semidefinite. For square matrices A , we denote by $\text{tr}(A)$, $\text{diag}(A)$, $\text{rank}(A)$ the trace, diagonal and rank of A , respectively. Finally, given an $n \times m$ matrix A , we denote by $\text{col} : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^m$ the column-major order representation of A : i.e., $\text{col}(A)$ maps the elements of A to a vector, by stacking the m columns of A on top of each other.

3.1 Overview

Our model assumes that user interests are dynamic: they are affected both by recommendations users receive, as well as by how other users' interests evolve. In particular, our model of user behavior takes into account the following factors:

1. *Inherent interests.* Our model accounts for an inherent predisposition users may have, e.g., towards particular topics or genres. This is static and does not change through time.
2. *Attraction.* As per the mere-exposure effect, users may exhibit attractive behavior: if a type of content is shown very often by the recommendation service, this might reinforce the desire of a user to consume it.
3. *Aversion.* Users may also exhibit aversive behavior: a user can grow tired of a topic that she sees very often, and may want to see something new or rare.
4. *Social influence.* A user's behavior can be affected by what people in her social circle (e.g., her friends or family) are interested in.

Under the joint effect of the factors above, suggestions made by the recommender instigate an *interest cascade* over the users. Suggestions alter user interests through attraction or aversion; in turn, these changes affect neighboring users as well, on account of their social behavior. These effects propagate dynamically over the users' social network. Next, we formally describe how each of these factors is incorporated in our model.

3.2 Recommender System and User Utilities

We consider n users receiving recommendations from an entity we call the recommender. We denote by $[n] \equiv \{1, 2, \dots, n\}$ the set of all users. At each time step $t \in \mathbb{N}$, the recommender suggests an item to each user in $[n]$, selected from a catalog C of available items. The user accrues a utility from the item recommended. As discussed below, the recommender's goal is to suggest items that maximize the aggregate user utility, i.e., the social welfare.

Following the standard convention in recommender systems, we assume factor-based user utilities. At each $t \in \mathbb{N}$, each user $i \in [n]$ has an *interest profile* represented by a d -dimensional vector $\mathbf{u}_i(t) \in \mathbb{R}^d$. Moreover, the item recommended to user i at time t is represented by a d -dimensional *feature profile* $\mathbf{v}_i(t) \in \mathbb{R}^d$. Then, the expected rating¹ a user i would give to the item suggested to her at time t is given by $F(\mathbf{u}_i(t), \mathbf{v}_i(t))$, where

$$F(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u}, \mathbf{v} \rangle = \sum_{k=1}^d \mathbf{u}_k \mathbf{v}_k, \quad (1)$$

i.e., the inner product between the interest and feature profiles [18, 20]. Intuitively, each coordinate of a feature profile can be perceived as an item-specific feature such as, e.g., a movie's genre or an article's topic. The corresponding coordinate in an interest profile captures the propensity of the user to react positively or negatively to this feature.

We call $F(\mathbf{u}_i(t), \mathbf{v}_i(t))$ the *utility* of user i from the suggested item at time t . Without loss of generality², we assume that the item profiles $\mathbf{v}_i \in \mathbb{R}^d$ are normalized, i.e.: $\|\mathbf{v}_i(t)\|_2 = 1$ for all $i \in [n]$, $t \in \mathbb{N}$. Under this assumption, given that a user's profile is \mathbf{u} , the

¹In practice, (1) best approximates *centered ratings*, i.e., ratings offset by a global average across users.

²Note that $F(\mathbf{u}, \mathbf{v}) = F(s\mathbf{u}, \frac{1}{s}\mathbf{v})$, for any scalar $s \in \mathbb{R}$, so we can assume that either user or feature profiles have a bounded norm.

best item to recommend to user i is the one that yields the highest expected rating; indeed, this is

$$\arg \max_{\mathbf{v} \in \mathbb{R}^d, \|\mathbf{v}\|_2=1} F(\mathbf{u}, \mathbf{v}) = \mathbf{u} / \|\mathbf{u}\|_2,$$

i.e., the item that maximizes the utility of a user i . Note that identifying items that maximize the *aggregate* utility across users (i.e., the sum of expected ratings to suggested items), is a natural goal for the recommender.

3.3 Interest Evolution

The evolution of user interests captures the four factors outlined in Section 3.1. At each time step $t \in \mathbb{N}$, the interest profile of a user $i \in [n]$ is chosen alternately between either a *personalized* or a *social* behavior. If personalized, the behavior of a user is again selected among three possible outcomes, each corresponding to inherent interests, attraction, and aversion, respectively.

The selection of which of these four behaviors takes place at a given time step is random, and occurs independently of selections at other users as well as selections at previous time slots. We denote with $\beta \in [0, 1]$ the probability that the user selects a social behavior at time slot t . The probability of selecting a personalized behavior is thus $1 - \beta$. Interests at these two distinct events are as follows:

Personalized Behavior. If a user's interest is selected through a personalized behavior, the user selects her profile through one of the three personalized factors outlined in Section 3.1. In particular, for every $i \in [n]$, there exist probabilities $\alpha_i, \gamma_i, \delta_i \in [0, 1]$ such that $\alpha_i + \gamma_i + \delta_i = 1$, and:

- **Inherent interests.** With probability α_i , user i follows her *inherent interests*. That is, $\mathbf{u}_i(t)$ is sampled from a probability distribution μ_i^0 over \mathbb{R}^d . This distribution does not vary with t and captures the user's inherent predisposition.

- **Attraction.** With probability γ_i , user i selects a profile that is *attracted* to the items suggested to her in the past. To capture this notion denote by $\mathcal{V}_i(t) = \{\mathbf{v}_i(\tau)\}_{\tau \leq t}$ the *history* of (profiles of) items suggested to a user i . Then, the interest of a user under attraction is given by $g(\mathcal{V}_i(t-1))$, a weighted average of the items suggested to it in the past. That is:

$$\mathbf{u}_i(t) = g(\mathcal{V}_i(t-1)) = \frac{\sum_{\tau=0}^{t-1} w_{t-\tau} \mathbf{v}(\tau)}{\sum_{\tau=0}^{t-1} w_{t-\tau}}. \quad (2)$$

- **Aversion.** With probability δ_i , user i selects a profile that is *repulsed* by the items suggested to her in the past; that is, her interest profile is given by

$$\mathbf{u}_i(t) = -g(\mathcal{V}_i(t-1)) = -\frac{\sum_{\tau=0}^{t-1} w_{t-\tau} \mathbf{v}(\tau)}{\sum_{\tau=0}^{t-1} w_{t-\tau}}. \quad (3)$$

To gain some intuition on (2) and (3), recall that a user's utility at time t is given by (1). Therefore, a profile generated under (2) implies that the suggestion that *maximizes* her utility at time t would be one that aligns perfectly (i.e., points in the same direction as) the weighted average g up to time $t-1$. In contrast, under the aversive behavior (3), the same suggestion *minimizes* the user's utility.

Note that the weighted average g is fully determined by the sequence weights $\{w_\tau\}_{\tau \in \mathbb{N}}$. By selecting decaying weights, a higher importance can be placed on more recent suggestions.

Social Behavior. User i 's profile is selected through social behavior with probability β . Conditioned on this event:

- **Social Influence.** A user aligns her interests with a user j selected from her social circle with probability P_{ij} . That is:

$$\mathbf{u}_i(t) = \mathbf{u}_j(t-1), \quad \text{with probability } P_{ij}, \quad (4)$$

where $\sum_j P_{ij} = 1$.

The probability $P_{ij} \in [0, 1]$ captures the influence that user j has on user i . Note that users j for which $P_{ij} = 0$ (i.e., outside i 's social circle) have no influence on i . Moreover, the set of pairs (i, j) s.t. $P_{ij} \neq 0$, defines the social network among users. We denote by $P \in [0, 1]^{n \times n}$ the stochastic matrix with elements P_{ij} , $i, j \in [n]$; we assume that P is ergodic (i.e., irreducible and aperiodic) [11].

Under these dynamics, interests evolve in the form of a dynamic cascade: suggestions made by the recommender act as a forcing function, altering interests either through attraction or aversion. Such changes propagate across users through the social network.

3.4 Recommended Item Distribution

In practice, the recommender has access to a finite ‘‘catalog’’ of items. Recalling that feature profiles have norm 1, the recommender’s catalog can be represented as a set $C \subseteq \mathbb{B}$, where $\mathbb{B} = \{\mathbf{v} \in \mathbb{R}^d : \|\mathbf{v}\|_2 = 1\}$ is the set of items of norm 1 (i.e., the d -dimensional unit ball).

We assume that the recommender selects the items $\mathbf{v}_j(t) \in \mathbb{B}$ suggested to user $i \in [n]$ by sampling them from a discrete distribution ν_i over \mathbb{B} , whose support is C . Note that the expected feature profile of a suggested item is a weighted average among the vectors in C . As such, it belongs to the *convex hull* of catalog C ; formally:

$$\bar{\mathbf{v}}_i = \int_{\mathbf{v} \in \mathbb{B}} \mathbf{v} d\nu_i \in \text{conv}(C), \quad (5)$$

Note that $\text{conv}(C)$ is a convex polytope included in \mathbb{B} .

As we will see later in our analysis (c.f. Theorem 1), the steady state user utilities depend only on the expectations $\bar{\mathbf{v}}_i$, $i \in [n]$, rather than the entire distributions ν_i . We will thus refer to $\{\bar{\mathbf{v}}_i\}_{i \in [n]}$ as the *recommender strategy*; it is worth keeping in mind that, given a $\bar{\mathbf{v}}_i \in \text{conv}(C)$, finding a ν_i such that (5) holds can be computed in polynomial time in $|C|$ (see also Section 4.4).

We further assume that the catalog C is large; in particular, for large catalog size $|C|$, we have:

$$\text{conv}(C) \simeq \mathbb{B}. \quad (6)$$

This would be true if, for example, each item in the catalog are generated in an i.i.d. fashion from a distribution that covers the entire ball \mathbb{B} ; this distribution need not be uniform³. We revisit the issue of how to pick a distribution ν_i given $\bar{\mathbf{v}}_i$, as well as how to interpret our results in the case of a finite catalog, in Section 4.4.

3.5 Recommendation Objective

Observe that, under the above dynamics, the evolution of the system is a Markov chain, whose state comprises the interest and feature profiles. We define the objective of the recommender as maximizing the social welfare, i.e., the sum of expected user utilities, in *steady state*. Formally, we wish to determine a strategy $\{\bar{\mathbf{v}}_i\}_{i \in [n]}$ (and, hence, distributions ν_i) that maximizes:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^T \sum_{i \in [n]} \langle \mathbf{u}_i(t), \mathbf{v}_i(t) \rangle = \lim_{t \rightarrow \infty} \sum_{i \in [n]} \mathbb{E}[\langle \mathbf{u}_i(t), \mathbf{v}_i(t) \rangle],$$

³Formally, $\lim_{|C| \rightarrow \infty} \text{conv}(C) = \mathbb{B}$ w.p. 1 if, e.g., items in catalog C are sampled independently from a probability distribution absolutely continuous to the uniform distribution on \mathbb{B} .

where the equality above holds w.p. 1 by the renewal theorem [11]. It is important to note that, under the interest dynamics described in 3.3, optimal recommendations to a user i cannot be obtained independently of recommendations to other users: user i 's profile depends on recommendations made not only directly to this user, but also to any user reachable through i 's social network.

4. OPTIMAL RECOMMENDATIONS

In this section, we discuss how the recommender selects which items to present to users to maximize the system’s social welfare. We begin by obtaining a closed-form formula for the social welfare in steady state, and then discuss algorithms for its optimization.

4.1 Steady State Social Welfare

Recall that μ_i^0 is the inherent profile distribution of user $i \in [n]$, and let μ_i be the steady state distribution of the profile of user i . We denote by $\bar{\mathbf{u}}_i = \int_{\mathbf{u} \in \mathbb{R}^d} \mathbf{u} d\mu_i$ and $\bar{\mathbf{u}}_i^0 = \int_{\mathbf{u} \in \mathbb{R}^d} \mathbf{u} d\mu_i^0$ the expected profile of $i \in [n]$ under the steady state and inherent profile distributions, respectively. Moreover, denote by $\bar{U}, \bar{U}^0, \bar{V} \in \mathbb{R}^{n \times d}$ the matrices of dimensions $n \times d$ whose rows comprise the expected profiles $\bar{\mathbf{u}}_i, \bar{\mathbf{u}}_i^0, \bar{\mathbf{v}}_i, i \in [n]$, respectively. Let also $A, \Gamma, \Delta \in \mathbb{R}^{n \times n}$ be the $n \times n$ diagonal matrices whose diagonal elements are the coefficients $(1 - \beta)\alpha_i, (1 - \beta)\gamma_i$, and $(1 - \beta)\delta_i$, respectively.

Then, the steady state social welfare can be expressed in closed form according to the following theorem.

THEOREM 1. *The expected social welfare in steady state is:*

$$G(\bar{V}) \equiv \text{tr} \left[(I - \beta P)^{-1} \left(A \bar{U}^0 \bar{V}^T + (\Gamma - \Delta) \bar{V} \bar{V}^T \right) \right], \quad (7)$$

where $\text{tr}(\cdot)$ denotes the matrix trace.

PROOF. Observe that at any time step $t \in [n]$ the profiles $\mathbf{u}_i(t)$ and $\mathbf{v}_i(t)$ are independent random variables. Hence,

$$\begin{aligned} \lim_{t \rightarrow \infty} \sum_{i \in [n]} \mathbb{E}[\langle \mathbf{u}_i(t), \mathbf{v}_i(t) \rangle] &= \lim_{t \rightarrow \infty} \sum_{i \in [n]} \langle \mathbb{E}[\mathbf{u}_i(t)], \mathbb{E}[\mathbf{v}_i(t)] \rangle \\ &= \sum_{i \in [n]} \langle \bar{\mathbf{u}}_i, \bar{\mathbf{v}}_i \rangle = \text{tr}(\bar{U} \bar{V}^T). \end{aligned} \quad (8)$$

Observe that by the linearity of expectation $\mathbb{E}[g(\mathcal{V}_i(t))] = \bar{\mathbf{v}}_i$ for all $t \in \mathbb{N}$ and $i \in [n]$. Thus, for $U(t) = [\mathbf{u}_i(t)]_{i \in [n]} \in \mathbb{R}^{n \times d}$ the matrix of user profiles at time t , we get that

$$\mathbb{E}[U(t)] = A \bar{U}^0 + \beta P \mathbb{E}[U(t-1)] + \Gamma \bar{V} - \Delta \bar{V}.$$

As βP is sub-stochastic and ergodic, the Perron-Frobenius theorem [11] implies that $\bar{U} = \lim_{t \rightarrow \infty} \mathbb{E}[U(t)]$ exists and

$$\bar{U} = A \bar{U}^0 + \beta P \bar{U} + (\Gamma - \Delta) \bar{V}.$$

Solving this linear system, and substituting the solution for \bar{U} in (8), yields the theorem. \square

An important consequence of Theorem 1 is that the steady state social welfare depends only on the expected profiles \bar{V} , rather than the entire distributions $\nu_i, i \in [n]$. Hence, determining the optimal recommender strategy amounts to solving the following quadratically-constrained quadratic optimization problem (QCQP):

$$\begin{aligned} &\text{GLOBAL RECOMMENDATION} \\ \text{Max.} &: \quad \text{tr} \left[(I - \beta P)^{-1} \left(A \bar{U}^0 \bar{V}^T + (\Gamma - \Delta) \bar{V} \bar{V}^T \right) \right] \\ \text{subj. to:} & \quad \|\bar{\mathbf{v}}_i\|_2^2 \leq 1, \text{ for all } i \in [n]. \end{aligned} \quad (9)$$

where the norm constraint comes from (6). Note that this is indeed a global optimization: to solve it, recommendations across different

users need to be taken into account jointly. This manifests in (9) through the quadratic term in the social welfare objective.

4.2 SDP Relaxation

The QCQP (9) is not necessarily convex. It is thus not a priori clear whether it can be solved in polynomial time. However, there is a way to reduce to a semi-definite program (SDP) relaxation, which can be solved in polynomial time. Interestingly, in many cases, the solution obtained for the SDP relaxation turns out to be an optimal solution to our original problem (9), and there is a simple and efficient test that can verify whether the obtained solution is optimal. Finally, when the solution is not optimal, it can be transformed to yield a constant-factor approximation. We are thus able to obtain a strong and elegant theoretical result for solving the GLOBALRECOMMENDATION problem.

It is important to note that the large-catalog assumption (6) is crucial to tractability: replacing the quadratic constraints with the linear constraints (5) does not lead to a problem that is amenable to an SDP relaxation. In fact, generic quadratic problems with linear constraints are known to be inapproximable, unless $P = NP$ [25].

Deriving an SDP Relaxation. We begin by describing first how to express (9) as “almost” an SDP, except for a rank constraint:

THEOREM 2. *There exists a symmetric matrix $H \in \mathbb{R}^{(nd+1) \times (nd+1)}$ and a convex polyhedral set $\mathcal{D} \in \mathbb{R}^{nd+1}$ such that GLOBAL RECOMMENDATION (9) is equivalent to:*

$$\begin{aligned} \text{Max.: } & \text{tr}(HY) \\ \text{subj. to: } & Y \succeq 0, \text{diag}(Y) \in \mathcal{D}, \text{rank}(Y) = 1, \end{aligned} \quad (10)$$

where $Y \in \mathbb{R}^{(nd+1) \times (nd+1)}$.

PROOF. Let $\mathbf{x} = \text{col}(\bar{V}) \in \mathbb{R}^{nd}$ be the column-major order vector representation of the recommender’s strategy, and $\mathbf{b} = \text{col}((I - \beta P)^{-1} A \bar{U}^0) \in \mathbb{R}^{nd}$ the vector representation of the linear term in (7). Moreover, for $Q = (I - \beta P)^{-1} (\Gamma - \Delta) \in \mathbb{R}^{n \times n}$, consider the following block-diagonal symmetric matrix, where $\frac{Q+Q^T}{2}$ is repeated d times:

$$H_0 = \begin{bmatrix} \frac{Q+Q^T}{2} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \frac{Q+Q^T}{2} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \frac{Q+Q^T}{2} \end{bmatrix} \in \mathbb{R}^{nd \times nd}.$$

Under this notation, (9) can be written as

$$\begin{aligned} \text{Max. } & \mathbf{b}^T \mathbf{x} + \mathbf{x}^T H_0 \mathbf{x} \\ \text{subj. to } & \mathbf{x}^2 \in \mathcal{D}_0, \end{aligned} \quad (11)$$

where $\mathbf{x}^2 = [\mathbf{x}_k^2]_{k \in [nd]} \in \mathbb{R}_+^{nd}$ results from squaring the elements of \mathbf{x} , and \mathcal{D}_0 the set implied by the norm constraints:

$$\mathcal{D}_0 = \{\mathbf{x}' \in \mathbb{R}^{nd} \mid \forall i \in [n], \sum_{j=1}^{nd} \mathbb{1}_{j \bmod n = i \bmod n} \mathbf{x}'_j \leq 1\}.$$

Note that \mathcal{D}_0 is a convex polyhedral set defined by linear equality constraints. Moreover, (11) can be homogenized to a quadratic program without linear terms using the following standard trick (see also [22, 25, 29]). Introducing an auxiliary variable t , the objective can be replaced by $t\mathbf{b}^T \mathbf{x} + \mathbf{x}^T H_0 \mathbf{x}$, where t satisfies the constraint $t^2 \leq 1$. Setting $\mathbf{y} = (\mathbf{x}, t) \in \mathbb{R}^{nd+1}$, this yields:

$$\begin{aligned} \text{Max.: } & \mathbf{y}^T H \mathbf{y} \\ \text{subj. to: } & \mathbf{y}^2 \in \mathcal{D}, \end{aligned} \quad (12)$$

Algorithm 1: GLOBAL RECOMMENDATION ALGORITHM

Solve SDP RELAXATION (13); let Y be its solution;
if $\text{rank}(Y) = 1$ **then**
 Let $\lambda > 0$ be the unique positive eigenvalue of Y , and \mathbf{e} the corresponding eigenvector;
 $(\mathbf{x}, t) \leftarrow \sqrt{\lambda} \cdot \mathbf{e}$;
else
 Construct a factorization $Y = Z^T Z$, $Z \in \mathbb{R}^{(nd+1) \times (nd+1)}$;
 Sample a random $u \in \mathbb{R}^{nd+1}$ from $\mathcal{N}(0, I_{nd+1})$;
 $\sigma \leftarrow \text{sgn}(Z^T u)$;
 $D \leftarrow$ a diagonal matrix containing $\sqrt{\text{diag} Y}$;
 $(\mathbf{x}, t) \leftarrow D \sigma$;
end if
Let $\bar{V} \in \mathbb{R}^{n \times d}$ be such that $\text{col}(\bar{V}) = t \cdot \mathbf{x}$;
return \bar{V} ;

where H is the following symmetric matrix:

$$H = \begin{bmatrix} H_0 & \mathbf{b}/2 \\ \mathbf{b}^T/2 & 0 \end{bmatrix} \in \mathbb{R}^{(nd+1) \times (nd+1)},$$

and

$$\mathcal{D} = \{\mathbf{y}' = (\mathbf{x}', t') \in \mathbb{R}^{nd+1} \mid \mathbf{x}' \in \mathcal{D}_0, t' \leq 1\}.$$

To see that (12) and (11) are equivalent, observe that an optimal solution (\mathbf{x}, t) to (12) must be such that $t = -1$ or $t = +1$. If $t = +1$, then \mathbf{x} is an optimal solution to (11); if $t = -1$, then $-\mathbf{x}$ is an optimal solution to (11). Finally, (12) is equivalent to (10), by setting $Y = \mathbf{y}\mathbf{y}^T$ and using the fact that $\mathbf{y}^T H \mathbf{y} = \text{tr}(H \mathbf{y}\mathbf{y}^T)$.

In particular, given an optimal solution Y to (10), an optimal solution to GLOBALRECOMMENDATION can be constructed as follows. Since $Y \succeq 0$ and $\text{rank}(Y) = 1$, there exists $\mathbf{y} \in \mathbb{R}^{nd+1}$ such that $Y = \mathbf{y}\mathbf{y}^T$. More specifically, Y has a unique positive eigenvalue λ . If \mathbf{e} is the corresponding eigenvector, $\mathbf{y} = (\mathbf{x}, t) = \sqrt{\lambda} \cdot \mathbf{e}$. An optimal solution to (9) is thus the matrix $\bar{V} \in \mathbb{R}^{n \times d}$ with column-major order representation $\text{col}(\bar{V}) = t \cdot \mathbf{x}$. \square

Problem (10) is still not convex, on account of the rank constraint on Y . However, in light of Theorem 2, a natural relaxation for GLOBAL RECOMMENDATION is the following semi-definite program, resulting from dropping this rank constraint:

SDP RELAXATION

$$\begin{aligned} \text{Max.: } & \text{tr}(HY) \\ \text{subj. to: } & Y \succeq 0, \text{diag}(Y) \in \mathcal{D}. \end{aligned} \quad (13)$$

This is a relaxation, in the sense that it increases the feasible set: any solution to (10) will also be a solution to (13). Crucially, (13) is a convex SDP problem, and can be solved in polynomial time. Moreover, if it happens that the optimal solution Y of (13) has rank 1, this solution is also guaranteed to be an optimal solution to (10), and can thus be used to construct an optimal solution to GLOBAL RECOMMENDATION, by Theorem 2. If, on the other hand, $\text{rank}(Y) > 1$, we are not guaranteed to retrieve an optimal solution to (10). However, a solution with a provable approximation guarantee can still be constructed through a randomization technique, originally proposed by Goemans and Williamson [13].

Approximation Algorithm. Algorithm 1 summarizes the steps in the approach outlined above to solving GLOBAL RECOMMENDATION. First, the algorithm obtains an optimal solution Y to the SDP (13). It then tests if $\text{rank}(Y) = 1$, i.e., if this solution happens to have rank 1. If it does, then it is also a solution to (10), and an optimal solution to (9) can be constructed as outlined in the proof of Theorem 2. In particular, Y can be written as $Y = \mathbf{y}\mathbf{y}^T$, where

$\mathbf{y} = (\mathbf{x}, t) \in \mathbb{R}^{nd+1}$ can be computed from the unique positive eigenvalue of Y and its corresponding eigenvector. The optimal solution to (9) can subsequently be obtained as the matrix $\bar{V} \in \mathbb{R}^{n \times d}$ that has a column-major order representation $\text{col}(\bar{V}) = t \cdot \mathbf{x}$.

If, on the other hand, $\text{rank}(Y) > 1$, the algorithm returns a vector (\mathbf{x}, t) constructed in a randomized fashion. In particular, the algorithm returns the vector $\sqrt{\text{diag}(Y)}$, namely the square root of Y 's diagonal elements, with each coordinate multiplied by a random sign (+1 or -1). The random sign vector $\sigma \in \{-1, +1\}^{nd+1}$ used in this multiplication is constructed as follows. Given that $Y \succeq 0$, there exists a matrix $Z \in \mathbb{R}^{n \times n}$ that factorizes Y , i.e., $Y = Z^T Z$. Such a matrix can be obtained in polynomial time from the eigendecomposition of Y . Having Z , the algorithm proceeds by sampling a random vector $u \in \mathbb{R}^{nd+1}$ from a standard Gaussian distribution. Then, σ is the binary vector computed by applying the sign operator on the coordinates of vector $Z^T u$.

The resulting random $\mathbf{y} = (\mathbf{x}, t) \in \mathbb{R}^{nd+1}$ is guaranteed to be a feasible solution to (10). Most importantly, the following approximation guarantee for the quality of the corresponding solution to GLOBAL RECOMMENDATION can be provided:

THEOREM 3 (YE [25]). *Let G^* , G_* be the maximal and minimal values of the social welfare G given by (7), evaluated over the feasible domain of (9). Let also \bar{V} be the solution generated by Algorithm 1 when $\text{rank}(Y) > 1$. Then*

$$\frac{G^* - \mathbb{E}_u[\bar{V}]}{G^* - G_*} \leq \frac{\pi}{2} - 1 < \frac{4}{7},$$

where the expectation $\mathbb{E}_u[\cdot]$ is over the Gaussian vector u .

The existence of a simple test (namely, $\text{rank}(Y) = 1$) verifying that the solution produced by Algorithm (1) is optimal is important. In fact, in Section 6, we study an extensive set of instances, involving several social network topologies and combinations of aversive and attractive behavior. *In each and every instance studied, Algorithm 1 yielded an optimal solution.* Hence, although the quadratic program (9) is not known to be within the class of problems that can be solved exactly through an SDP relaxation, the experiments in Section 6 suggest that a stronger guarantee than the one provided by Theorem 3 is attained in practice.

4.3 Special Cases

Though for generic instances of (9) we cannot obtain a better theoretical guarantee than Theorem 3, there are specific instances of (9) for which optimality is always attained, and the approximation through randomization is not necessary. As these cases are also of practical interest, we briefly review them below.

Attraction Dominance. Consider a scenario where (a) $\gamma_i > \delta_i$ for all $i \in [n]$ and (b) $\bar{U}^0 \geq \mathbf{0}$. Intuitively, (a) implies that attraction to proposed content is more dominant than aversion to content, while (b) implies that user profile features take only positive values. Hence, the matrix H in Theorem 2 has nonnegative off-diagonal elements. Although the QCQP (9) in this case is not convex, it is known that in this specific case Algorithm 1 provides an optimal, rank-1 solution [34].

Uniform Aversion Dominance. Consider a scenario where (a) all parameters are uniform across users, i.e., $\gamma_i = \gamma$ and $\delta_i = \delta$, for all $i \in [n]$, and (b) $\gamma < \delta$, i.e., aversion dominates user behavior. In this case, the QCQP (9) is convex and can thus be solved optimally by standard interior point methods in polynomial time [6].

No Personalization. Consider the scenario where the same item is recommended to all users, i.e., $\mathbf{v}_i(t) = \mathbf{v}(t)$, $\forall i \in [n]$. In this case, GLOBAL RECOMMENDATION reduces to a quadratic objective with a single quadratic constraint, in which case even though

the problem may not be convex, Algorithm 1 is guaranteed to find a rank-1, optimal solution [6].

No Social Network. In the case where $\beta = 0$, and there is no social component to the optimization, the social welfare (7) becomes separable in $\bar{\mathbf{v}}_i$, i.e., $G(\bar{V}) = \sum_{i \in [n]} G_i(\bar{\mathbf{v}}_i)$, where G_i is a quadratic function. Then, the optimization is separable, and a solution to (9) can be obtained by solving $\max_{\bar{\mathbf{v}}_i \in \mathbb{R}^d: \|\bar{\mathbf{v}}_i\| \leq 1} G(\bar{\mathbf{v}}_i)$ for each $i \in [n]$; these are again quadratic problems with a single quadratic constraint, and can be solved exactly by Algorithm 1 [6].

4.4 Finite Catalog

Recall that our analysis assumes (6), which becomes applicable for a large catalog C covering the unit ball \mathbb{B} . We describe below how a computed profile $\bar{\mathbf{v}}_i$, $i \in [n]$ can be used to construct a distribution ν_i over catalog C .

If $\bar{\mathbf{v}}_i \in \text{conv}(C)$, the recommender can select probabilities $\nu_i(\mathbf{v})$, for $\mathbf{v} \in C$, that satisfy (5); this equality, along with the positivity constraints, and the constraint $\sum_{\mathbf{v} \in C} \nu_i(\mathbf{v}) = 1$ (as ν_i is a distribution), are linear, and define a feasible set. Thus, finding a probability distribution satisfying (5) (i.e., that lies in the feasible set) is a linear program, which can be solved in polynomial time.

If, on the other hand, $\bar{\mathbf{v}}_i \notin \text{conv}(C)$, the same procedure can be applied to the projection of $\bar{\mathbf{v}}_i$ to $\text{conv}(C)$. Given that $\text{conv}(C)$ is a convex polytope, this can again be computed in polynomial time. Moreover, under (6), if the catalog C is large this projection will be close to the optimal value $\bar{\mathbf{v}}_i$.

5. PARAMETER LEARNING FROM DATA

In this section, we provide an algorithm for validating the existence of attraction and aversion phenomena in real datasets. In short, our approach involves incorporating aversion and attraction parameters into Matrix Factorization (MF) [18, 20]; we treat parameters α_i , γ_i and δ_i as regularization terms, which are learned through cross validation.

Extending MF. We focus on datasets that comprise ratings generated by users, at known times (such as the datasets used in Section 6). More specifically, we assume access to a dataset represented by tuples of the form (i, j, r_{ij}, t) where $i \in [n] \equiv \{1, \dots, n\}$ is the id of a user, $j \in [m] \equiv \{1, \dots, m\}$ the id of an item, $r \in \mathbb{R}$ the feedback (rating) provided by user i to item j and $t \in [T]$ the time at which the rating took place. Denoting by $E \subset [n] \times [m]$ the pairs appearing in tuples in this dataset, recall that matrix factorization (MF) amounts to constructing profiles $\mathbf{u}_i \in \mathbb{R}^d$, $\mathbf{v}_j \in \mathbb{R}^d$ that are solutions to:

$$\min_{\mathbf{u}_i, \mathbf{v}_j} \sum_{i \in [n], j \in [m], (i, j) \in E} (r_{ij} - \langle \mathbf{u}_i, \mathbf{v}_j \rangle)^2 + \lambda \sum_{i \in [n]} \|\mathbf{u}_i\|_2^2 + \mu \sum_{j \in [m]} \|\mathbf{v}_j\|_2^2 \quad (14)$$

where $\lambda, \mu > 0$ are regularization parameters to be learned through cross validation. Though this is not a convex problem, it is typically solved either through gradient descent or alternating least squares techniques, both of which perform well in practice [18, 20].

We incorporate attraction and aversion in this formulation as follows. First, at any time step $t \in [T]$, the profile of a user i is given by $\mathbf{u}_i(t) \in \mathbb{R}^d$. Let $E_i(t) \subseteq [m]$ be the set of items rated by user i at time t and

$$\mathcal{V}_i(t) = \{\mathbf{v}_j : j \in E_i(\tau), 1 \leq \tau \leq t\}$$

be the set of items the user has interacted with up to time t (inclusive). As in Section 3, we denote by $g(\mathcal{V}_i(t))$ the weighted average

Algorithm 2: Attraction-Aversion Learning Algorithm

Obtain $\mathbf{u}_i^0, \forall i \in [n]$, and $\mathbf{v}_j, \forall j \in [m]$ through standard MF (14);
 Compute $g(\mathcal{V}_i(t)), \forall i \in [n], \forall t \in [T]$;
 Split the dataset into k folds;
 Initialize values in α, γ, δ uniformly at random from $[0, 1]$ and project
 $(\alpha_i, \gamma_i, \delta_i)$ to the set $\{(x, y, z) \geq 0 : x + y + z = 1\}, \forall i \in [n]$;
repeat
 $(\alpha, \gamma, \delta, \kappa) \leftarrow (\alpha, \gamma, \delta, \kappa) - \rho \nabla \text{SE}_{\text{test}}(\alpha, \gamma, \delta, \kappa)$;
 Project $(\alpha_i, \gamma_i, \delta_i)$ to $\{(x, y, z) \geq 0 : x + y + z = 1\}, \forall i \in [n]$;
until Change of SE_{test} in two consecutive iterations is small enough

Case	Flixster	FilmTipSet	MovieLens
# users	4.6K	443	8.9K
# items	25K	4.3K	3.8K
# ratings	1.8M	118K	1.3M
# SN edges	44K	N/A	N/A

Table 1: Datasets statistics

of items in $\mathcal{V}_i(t)$. Then, we propose obtaining $\mathbf{u}_i(t)$ as solutions to:

$$\min_{\substack{\mathbf{u}_i(t), i \in [n], \\ t \in [T]}} \sum_{\substack{t \in [T], i \in [n], \\ (i,j) \in E_i(t)}} (r_{ij} - \langle \mathbf{u}_i(t), \mathbf{v}_j \rangle)^2 + \sum_{i \in [n], t \in [T]} (\|\mathbf{u}_i(t) - \alpha_i \mathbf{u}_i^0 - (\gamma_i - \delta_i) g(\mathcal{V}_i(t))\|_2^2 + \kappa \|\mathbf{u}_i(t)\|_2^2), \quad (15)$$

where $\mathbf{u}_i^0, \mathbf{v}_j$ are computed through standard MF (14), and $\alpha_i, \gamma_i, \delta_i, i \in [n]$ and κ are also treated as regularization parameters, to be learned through cross validation. Note that, in contrast to (14), (15) is a simple linear regression problem, and the profiles $\mathbf{u}_i(t)$, where $i \in [n], t \in [T]$, can be computed in closed form.

Learning Procedure. Based on this approach, our algorithm for learning the vectors α, γ, δ is outlined in Algorithm 2. First, we learn the inherent profiles \mathbf{u}_i^0 and the item feature profiles \mathbf{v}_j by solving (14), through stochastic gradient descent. Then, we use these profiles to learn $\alpha_i, \gamma_i, \delta_i$ through cross validation. In particular, we split the ratings dataset in k folds, and use $k - 1$ folds as a training set, and one fold as a test set. In our evaluation, we set $k = 5$. We learn $\mathbf{u}_i(t)$ by solving (15) on this restricted dataset. Using these, we compute the square error on the test set as:

$$\text{SE}_{\text{test}} = \sum_{(i,j,t) \in \text{test}} (r_{ij} - \langle \mathbf{u}_i(t), \mathbf{v}_j \rangle)^2.$$

We repeat this process across k folds and obtain an average $\overline{\text{SE}}_{\text{test}}$.

We compute vectors α, γ, δ that minimize the average $\overline{\text{SE}}_{\text{test}}$. Note that this is a function of the regularization parameters of (15), i.e., $\overline{\text{SE}}_{\text{test}} = \overline{\text{SE}}_{\text{test}}(\alpha, \gamma, \delta, \kappa)$. As (15) admits a closed form solution, so does $\overline{\text{SE}}_{\text{test}}(\alpha, \gamma, \delta, \kappa)$. Using this, we find α, γ, δ through projected gradient descent, requiring that they sum to 1.

6. EXPERIMENTS

We perform experiments to evaluate our parameter learning and social welfare-maximizing algorithms on three real-world rating datasets: Flixster, FilmTipSet, and MovieLens, as well as several synthetically generated traces. The implementations are in Matlab and we use the CVX library [14] to solve the SDP in Algorithm 1. All experiments were run on a server with AMD Opteron 6272 CPUs (eight cores at 2.1GHz) and 128GB memory.

Dataset Preparations. We first describe the three real-world rating datasets. Their statistics are summarized in Table 1.

Flixster is a social movie rating site⁴. The original dataset, collected by Jamali et al. [16], comprises 1M users, 14M undirected friendship edges, and 8.2M timestamped ratings (ranging from 0.5 to 5 stars). We use Graclus⁵ to extract a dense subgraph of the social network. Further, we filter out users and movies with less than 100 ratings so that there is enough data to learn temporal profile vectors. This gives a core of 4.6K users, 44K edges, and 25K movies.

FilmTipSet⁶ is Swedish movie fans community. The data was originally published for a research competition in the CAMRa workshop⁷. It has 16K users, 67K movies, 2.8M timestamped ratings (on the scale of 1 to 5). We select users rating no less than 100 movies in both 2004 and 2005. This gives a core of 443 users, 4.3K movies, and 118K ratings.

The third dataset is MovieLens⁸ (the 10M-user version). We focus on users that have rated at least 20 movies in the year of 2000. Note that there is no social network in MovieLens. FilmTipSet contains some social networking information, which we were unable to use in our analysis due to its extreme sparsity (85 edges for the 443 core users).

6.1 Evaluation of Parameter Learning

Learning on Synthetic Data. We first run Algorithm 2 on a synthetically generated dataset to examine its accuracy. We set $n = 100, T = 100$, and $d = 5$. Each user $i \in [n]$ consumes one random item at every time step $t \in [T]$. For all users i , we generate “ground-truth” α_i, γ_i , and δ_i uniformly at random from $[0, 1]$ and normalize them so that $\alpha_i + \gamma_i + \delta_i = 1$. The expected inherent interest profiles and item profiles are generated uniformly at random from $[0, 1]^d$. Interest profiles evolve according to the dynamics in Section 3.3, with $\beta = 0$, and weighted average g with equal weights. At each step, users generate ratings computed by taking the inner product of appropriate profile vectors.

The learning rate ρ and regularization parameter κ in Algorithm 2 are both set to be 0.001 (determined by cross validation). The convergence condition of Algorithm 2 is set to be the change in $\overline{\text{SE}}_{\text{test}}$ being smaller than 10^{-6} . We repeat the process ten times with different random starting points and report the results obtained in the repetition that gives the smallest $\overline{\text{SE}}_{\text{test}}$.

Let $\alpha_i^\ell, \gamma_i^\ell$, and δ_i^ℓ be the learned evolution probabilities. We use RMSE to define the learning error w.r.t. α_i ’s:

$$\text{RMSE}_\alpha = \sqrt{\frac{\sum_{i=1}^n |\alpha_i - \alpha_i^\ell|^2}{n}}.$$

RMSE_γ and RMSE_δ can thus be defined in the same way. Figure 2a shows the decrease of these RMSEs as the number of iterations goes up. At convergence, they are 0.08, 0.58 and 0.56 respectively. In addition, Figure 2a also shows that the test RMSE (computed as $\sqrt{\overline{\text{SE}}_{\text{test}}/|\text{test}|}$) drops steadily as the learning proceeds, which finally converges to 0.02 in a total of 5300 iterations.

In Figure 2b, we show a scatter plot of the ground-truth probabilities and the learned probabilities (at convergence). Each data point has one ground-truth probability value in x -coordinate and the corresponding learned value in y -coordinate. The $y = x$ line indicates points for which the learned and ground truth probabilities are equal. As can be seen, the algorithm recovers α_i ’s almost perfectly, while the results for γ_i ’s and δ_i ’s are also reasonably good.

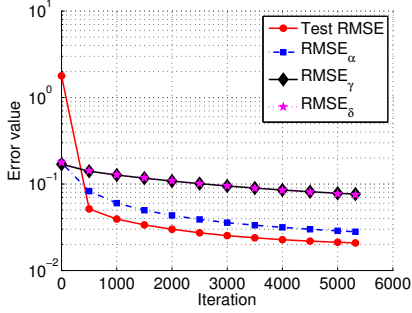
⁴<http://www.flixster.com/>

⁵<http://www.cs.utexas.edu/users/dml/Software/graclus.html>

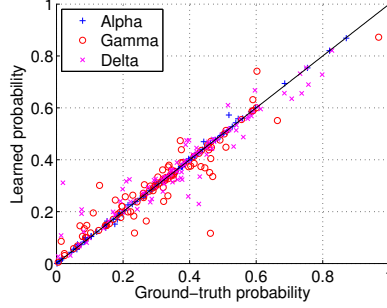
⁶<http://www.filmtipset.se/>

⁷<http://www.dai-labor.de/camra2010/>

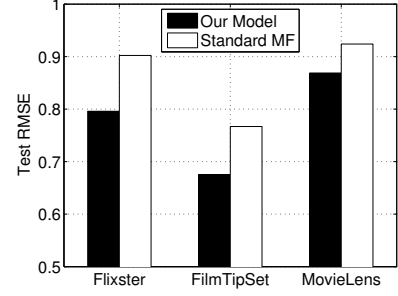
⁸<http://grouplens.org/datasets/movielens/>



(a) Test RMSE, $RMSE_{\alpha}$, $RMSE_{\gamma}$, and $RMSE_{\delta}$ over iterations

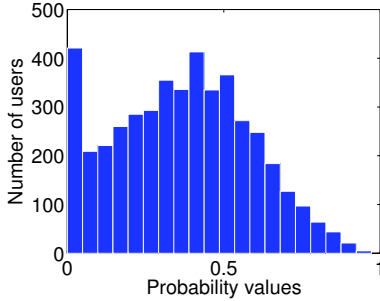


(b) Interest evolution probabilities learned on synthetic data

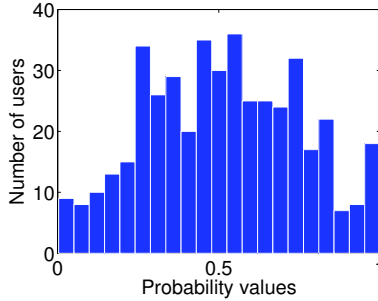


(c) Test RMSE comparisons

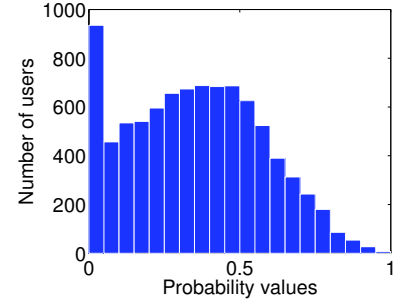
Figure 2: Evaluating parameter learning



(a) Flixster

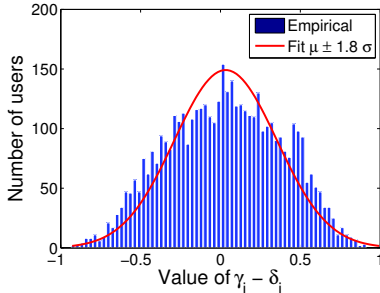


(b) FilmTipSet

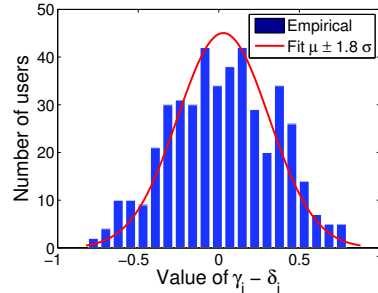


(c) MovieLens

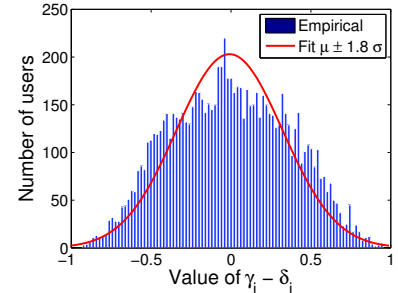
Figure 3: Learned values of α_i on three real-world datasets



(a) Flixster



(b) FilmTipSet



(c) MovieLens

Figure 4: Values of $\gamma_i - \delta_i$ on three real-world datasets

This gives us confidence in deploying the algorithm on real-world rating data, in which ground-truth parameters are not known.

Learning on Real Data. For each dataset, we sort the ratings in chronological order and split them into $T = 10$ time steps. A single time step corresponds to 3, 1.2, and 2.5 calendar months in Flixster, MovieLens, and FilmTipSet, respectively. We then run Algorithm 2 with learning rate $\rho = 0.001$, regularization parameter $\kappa = 0.001$, and number of latent features $d = 10$. Figure 3 shows the distributions of values learned for α_i 's. Furthermore, to compare the number of attraction-dominant users and the number of aversion dominant users, in Figure 4 we display the distribution of $\gamma_i - \delta_i$, along with a Gaussian distribution fitted by data within the interval $[\mu - 1.8\sigma, \mu + 1.8\sigma]$ (capturing 90% of a Gaussian distribution), where μ and σ^2 is the mean and variance of all $\gamma_i - \delta_i$'s. As can be seen, the empirical distribution has tails that are heavier than

the Gaussian (at about -0.5 and 0.5), indicating the existence of strongly aversive and strongly attracted users.

For reference, we also compare the average test RMSE on five-fold cross-validation achieved by our model and by standard MF. For standard MF, we implement the stochastic gradient descent method as in [20] with $d = 10$, learning rate 0.002, and regularization parameters determined by cross validation. As shown in Figure 2c, profiles learned by Algorithm 2 outperform standard MF in rating prediction, lowering the test RMSE by 11.8%, 11.9%, 6.18% on Flixster, FilmTipSet, and MovieLens respectively.

6.2 Social Welfare Performance

Next, we evaluate Algorithm 1, hereafter referred to as *GRA*, and compare the social welfare it yields with a baseline that ignores interest evolution. This baseline recommends to each user

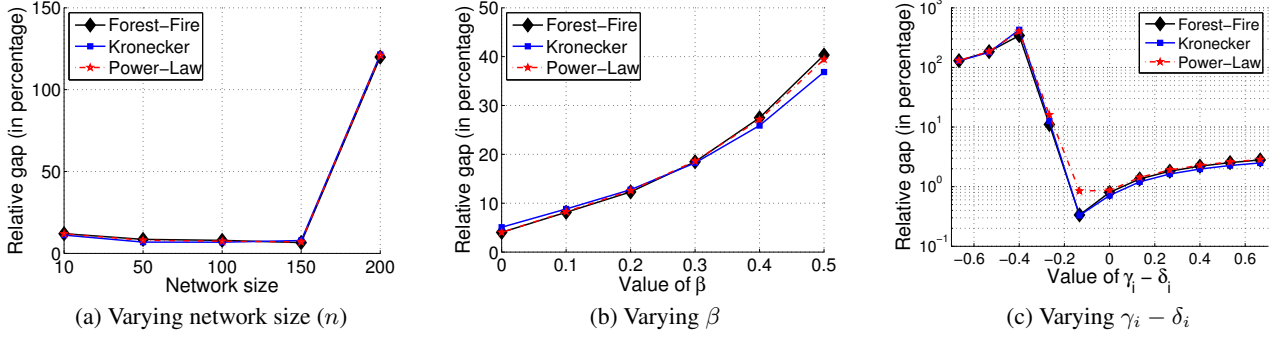


Figure 5: Relative increase in social welfare by *GRA* over *MF-Local* on synthetic datasets.

the item profile maximizing the user’s utility under the inherent profile computed by standard MF. For each user $i \in [n]$, this is $\mathbf{v}_i = \mathbf{u}_i^0 / \|\mathbf{u}_i^0\|_2$, for all $i \in [n]$. It is thus easy to see that $\|\mathbf{v}_i\|_2 = 1$ and it is in fact co-linear w.r.t. \mathbf{u}_i^0 . We hereafter refer to this baseline as *MF-Local*. In all experiments, following the literature of social influence propagation and maximization [8, 17], we set the influence probability of user j on user i to be $1/\text{deg}^{\text{in}}(i)$, where $\text{deg}^{\text{in}}(i)$ is the in-degree of node i in the network graph.

6.2.1 Experiments on Synthetic Networks

We start by evaluating the social welfare achieved by *GRA* and *MF-Local* on three different random networks that mimic the structure of a social network: Forest-Fire [4], Kronecker [21], and Power-Law [2]. For each type, we consider the following settings: Forest-Fire with forward and backward burning probability being 0.38 and 0.32 respectively, Kronecker with initiator matrix being $[0.9, 0.5; 0.5, 0.3]$, and Power-Law with exponent 2.1.

We vary the size of network graphs (i.e., number of users, n), the value of β (i.e. users’ tendency of getting influenced by friends), and the difference between γ_i and δ_i , to evaluate their effects on the performance of our *GRA* algorithm in comparison to *MF-Local*. Unless otherwise noted, α_i ’s, γ_i ’s, δ_i ’s, and inherent user profiles are sampled randomly and the process is repeated ten times, of which we take the average social welfare. Also, d is fixed to be 10. In all cases, we plot the *relative gap* in social welfare, i.e.,

$$(\text{Social Welfare}_{\text{GRA}} - \text{Social Welfare}_{\text{MF-Local}}) / |\text{Social Welfare}_{\text{MF-Local}}|.$$

Effect of Network Size. We test five different values for n : 10, 50, 100, 150, and 200 for Forest-Fire and Power-Law, and 16, 32, 64, 128, and 256 for Kronecker (by definition a random Kronecker graph has 2^w nodes where $w \in \mathbb{N}_+$ is the number of iterations of Kronecker product taken in the generation process [21]). As can be seen from Figure 5a, the gap between *GRA* and *MF-Local* is close to 10% for small graphs, but increases on all three networks for larger values of n : *GRA* achieves twice as much social welfare as *MF-Local*, for $n = 200$.

Effect of β . In this test, we vary the value of β from 0 up to 0.5. Network size is fixed at 100 for Forest-Fire and Power-Law, and 128 for Kronecker. Figure 5b shows that *GRA* significantly outperforms *MF-Local*, and more interestingly, the relative gap increases as β increases. This intuitively suggests that when the influence among users is higher, ignoring the joint effect of recommendations becomes more detrimental to maximizing the social welfare.

Effects of $\gamma_i - \delta_i$. Next, we test different values of $\gamma_i - \delta_i$, representing cases from extreme aversion dominance to attraction dominance. Network size is $n = 100$ and $\beta = 0.25$, while $\alpha_i = \alpha$,

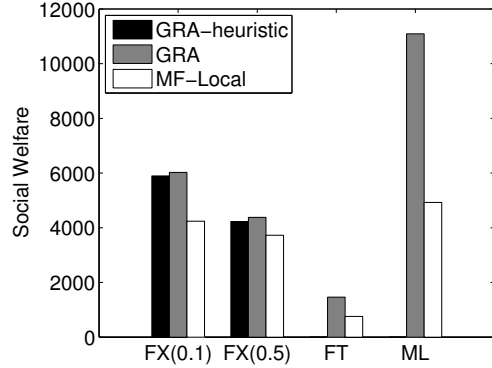


Figure 6: Social welfare achieved on: FX(0.1) and FX(0.5) – Flixster with $\beta = 0.1$ and 0.5; FT – FilmTipSet; ML – MovieLens.

$\gamma_i = \gamma$ and $\delta_i = \delta$ for all $i \in [n]$. We set α s.t. $\alpha(1 - \beta) = 0.25$, and vary $\gamma - \delta$, where $\alpha + \gamma + \delta = 1$. Relative gaps are shown in Figure 5c. All in all, we see that gaps are far more pronounced in the strongly aversive regime, as targeting to the existing profiles of users leads to suboptimal recommendations. For values less than -0.3, the social welfare under *MF-Local* is actually negative; it goes up as $\gamma - \delta$ increases, i.e., users tend towards attractive behavior. In contrast, the social welfare of *GRA* is always positive, and always greater than the one under *MF-Local*. As a result, there is a large gap for values at less than -0.3; the relative gap becomes small (but still positive) near -0.1, and then steadily increases.

It is important to note that in all evaluations of *GRA* over synthetic datasets, as well as the ones listed below on real datasets, *GRA* returned an optimal solution. That is, for all inputs tested, the matrix Y computed had rank 1. Hence, although the QCQP problem (9) is not known to be solvable in polynomial time, in practice, *GRA* outperforms the guarantee of Theorem 3.

6.2.2 Experiments on Real Data

We next compare the social welfare attained on Flixster, FilmTipSet, and MovieLens by *GRA* and *MF-Local*. For FilmTipSet and MovieLens where there is no social network considered, GLOBAL RECOMMENDATION is separable and can thus be *parallelized* (see Section 4.3): we can divide users into arbitrary subsets, run *GRA* on each of them, and then combine the total social welfare over all subsets as the final solution without any loss.

To improve the scalability of *GPA* over Flixster, and parallelize its execution, we adopt the following heuristic. First, we split the

social graph into 50 subgraphs using Graclus. Then, we solve SDP on each subgraph separately. Note that, in effect, this optimization ignores the edges between subgraphs, and thus only yields an approximation to the social welfare.

Figure 6 illustrates the performance of *GRA* and *MF-Local* on those datasets, where the values of α_i , γ_i , and δ_i are all from the learning results in Section 6.1 and dimensionality d is set to 10. We can see that *GRA* is significantly superior to *MF-Local*: on FilmTipSet (1461 vs. 757) and MovieLens (11092 vs. 4926), it achieves approximately twice the social welfare. On Flixster, we test two cases for β : 0.1 and 0.5, representing weak and strong social behavior respectively. For *GRA*, we adopt the aforementioned clustering-based heuristic to compute \bar{v}_i 's, and evaluate the welfare achieved by *GRA* in two ways: (i) simply calculating the welfare on the subgraph and taking the sum over all subgraphs (termed *GRA*-heuristic); (ii) taking the \bar{v}_i 's to calculate the social welfare on the entire graph (termed *GRA*). The values computed by method (ii) are only slightly different from (i), indicating that our clustering heuristic closely follows the true social welfare, while enabling parallelization. The relative gain of *GRA*-heuristic over *MF-Local* is 39.0% when $\beta = 0.1$ and 13.4% when $\beta = 0.5$. The running time of *GRA* is reasonably good, e.g., on a subgraph of Flixster with 94 nodes and 276 edges, *GRA* finishes in 90 seconds.

In summary, through extensive empirical evaluation on both real and synthetic data, we have demonstrated that first, the phenomenon of interest evolution, especially attraction and aversion, can indeed be observed from real-world rating data, and second, both of our learning algorithm and global recommendation algorithm are highly effective in their respective tasks.

7. CONCLUSIONS

Our study of attraction, aversion, and social influence suggests that such phenomena can be incorporated in recommendation decisions, and that the SDP relaxation approach brings relevant optimizations within the realm of tractability. The heuristic we exploited in Section 6, namely, parallelizing execution over weakly connected partitions of the social graph, highlights an approach for scalable, parallelizable solutions to the SDP relaxation. Nevertheless, further opportunities for improving efficiency exist: the sparse, block structure of the matrices in our SDP was not exploited by the generic solvers we employed. Investigating solutions that exploit this structure for higher efficiency is an interesting future direction. Moreover, although the QCQP that expresses our problem is not known to be exactly solvable through an SDP relaxation, all solutions we obtained through our experiments were actually optimal. Understanding if optimality holds for a wider class than the ones presented in Section 4.3 is also an important open problem. Finally, there are many phenomena beyond attraction, aversion and social influence that may affect a user's interests. The quadratic nature of our problem arises from the standard factor-based model for utilities: understanding if other phenomena inducing drift on profiles can also be cast in this framework is also an open question.

8. REFERENCES

- [1] Z. Abbassi et al. Getting recommender systems to think outside the box. In *Proceedings of the third ACM conference on Recommender systems*, pages 285–288. ACM, 2009.
- [2] W. Aiello, F. R. K. Chung, and L. Lu. A random graph model for massive graphs. In *STOC*, pages 171–180, 2000.
- [3] S. Amer-Yahia et al. Battling predictability and overconcentration in recommender systems. *IEEE Data Eng. Bull.*, 32(4):33–40, 2009.
- [4] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- [5] A. Beck and Y. C. Eldar. Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM Journal on Optimization*, 17(3):844–860, 2006.
- [6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [7] J. J. Brown and P. H. Reingen. Social ties and word-of-mouth referral behavior. *Journal of Consumer research*, 1987.
- [8] W. Chen, L. V. S. Lakshmanan, and C. Castillo. *Information and Influence Propagation in Social Networks*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2013.
- [9] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, pages 57–66. ACM, 2001.
- [10] X. Fang, S. Singh, and R. Ahluwalia. An examination of different explanations for the mere exposure effect. *Journal of consumer research*, 34(1):97–103, 2007.
- [11] R. G. Gallager. *Discrete stochastic processes*, volume 101. Kluwer Academic Publishers Boston, 1996.
- [12] M. Ge et al. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. *RecSys '10*.
- [13] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *JACM*, 42(6):1115–1145, 1995.
- [14] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. <http://cvxr.com/cvx>, Sept. 2013.
- [15] A. Grimes and P. J. Kitchen. Researching mere exposure effects to advertising—theoretical foundations and methodological implications. *International Journal of Market Research*, 49(2):191–219, 2007.
- [16] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, 2010.
- [17] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.
- [18] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, 2008.
- [19] Y. Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53(4), 2010.
- [20] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 2009.
- [21] J. Leskovec et al. Kronecker graphs: An approach to modeling networks. *J. of Machine Learning Research*, 11:985–1042, 2010.
- [22] Z.-q. Luo et al. Semidefinite relaxation of quadratic optimization problems. *Signal Processing Magazine, IEEE*, 27(3):20–34, 2010.
- [23] S. M. McNee et al. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI EA*, pages 1097–1101, 2006.
- [24] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization methods and software*, 9(1-3):141–160, 1998.
- [25] Y. Nesterov, et al. Semidefinite programming relaxations of nonconvex quadratic optimization. In *Handbook of semidefinite programming*, pages 361–419. Springer, 2000.
- [26] K. Radinsky, et al. Modeling and predicting behavioral dynamics on the web. In *WWW*, 2012.
- [27] A. D. Sarma et al. Understanding cyclic trends in social choices. In *WSDM*, pages 593–602, 2012.
- [28] D. Shah. *Gossip algorithms*. Now Publishers Inc, 2009.
- [29] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.
- [30] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, 2011.
- [31] Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Math. Prog.*, 84(2):219–226, 1999.
- [32] C. Yu, L. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: Diversification in recommender systems. In *EDBT*, 2009.
- [33] R. B. Zajonc. Attitudinal effects of mere exposure. *Journal of personality and social psychology*, 9:1–27, 1968.
- [34] S. Zhang. Quadratic maximization and semidefinite relaxation. *Mathematical Programming*, 87(3):453–465, 2000.