Closed-Loop Opinion Formation

Larissa Spinelli
Boston University
lspinell@bu.edu

Mark Crovella
Boston University
crovella@bu.edu

ABSTRACT

When information sources are moderated by recommender systems, so-called “filter bubbles” may restrict the diversity of content made available to users, potentially affecting their opinions. User opinions may in turn affect the output of recommender systems. In this work we ask how the dynamical system defined by user and recommender systems behaves, as each element evolves in time. In particular, we look at whether the use of recommender system can affect user experience and user opinions in a systematic way. We define and analyze three metrics to understand those effects – intensity, simplification, and divergence – and we explore both link-based and ratings-based recommender systems. Our results suggest that previous studies of this problem have been too simplistic, and that user opinions can evolve in complex ways under the influence of personalized information sources.

CCS CONCEPTS

• **Information systems** → *Collaborative filtering*;

KEYWORDS

Filter Bubble, User Experience, Metrics

ACM Reference format:
Larissa Spinelli and Mark Crovella. 2017. Closed-Loop Opinion Formation. In *Proceedings of WebSci '17, Troy, NY, USA, June 25-28, 2017,* 10 pages.
DOI: http://dx.doi.org/10.1145/3091478.3091483

1 INTRODUCTION

Recommender systems are an increasingly prevalent part of online services, and increasingly mediate access to online resources. Recommender systems are found in systems for online-shopping, video streaming, news feeds, search queries and social media. Recommender systems are employed not just to explicitly give recommendations, but to implicitly guide users, as in the selection and ordering of items in a Facebook news feed.

The term *filter bubble* refers to a narrowed access of information caused by personalization, often in combination with search engines [22]. The term and associated literature raises the concern that recommender systems may have an effect on society, for example by influencing user opinions. Given the prevalence of recommender systems, it is natural to ask whether they can have an effect on user opinions, and what the nature of that effect is.

In this paper we take this question a step further, and ask how user opinions *and* recommender systems together change as they interact over time. This question arises because many recommender systems are adaptive, making and incorporating observations of users preferences and choices, even as the users are themselves reacting to the recommended items (again, take as an example a Facebook news feed). This constitutes a kind of closed loop, or dynamical system. The central question we ask is: how do user opinions evolve when users and recommender systems each take input from the other, over time?

To answer this question requires significantly abstracting both recommender systems and users in a manner that captures essential properties. Taking recommender systems first, these are usually classified according to how the recommendations are made [2]: *content-based filtering* systems recommend items based on features of the previous items evaluated by the user, while *collaborative filtering* systems recommend items that people with similar tastes and preferences had evaluated before. Since the latter are based on the opinions of users, our focus is on collaborative filtering systems. We use two different models of collaborative filtering to compare results and explore differences: link based systems and ratings based systems. Link based systems are exemplified by online retailers, which rely on the past records of purchases by the set of all users. Ratings based systems are exemplified by movie, music, and hotel sites sites that recommend new (or previously seen) experiences for a user based on their explicit feedback. These two types of systems are often addressed using the *neighbor* approach and *latent factor* models, respectively [24].

Turning to user modeling, the central effect to be captured is how user opinion about an item is affected by the fact the item has been presented to the user by the recommender system. Here, we study a range of options, from the case where the user's reaction is random, to cases in which users have maximally negative or maximally positive reactions to the item presented. These reactions are captured by the recommender system and used to update its knowledge base.

While there is little work to date that has addressed this question in the form that we pose it, parts of our study have connection to the work in [8]. That paper constructed abstractions of three link based recommender systems, and analyzed the effects on user opinion under certain assumptions. Part of our work looks more deeply at the same systems, examining the effect of key assumptions, and shows effects that differ considerably from what is predicted in [8]. However our work also goes beyond link based systems to ratings based systems, where we find that the set of phenomena (system metrics) to be studied are more diverse.

Beyond abstracting the key system elements, we also must identify the properties of user opinions that are of interest. We study three key properties of the evolving set of user opinions: the *intensity* of individual user's opinions, the *simplicity* of individual

WebSci '17, Troy, NY, USA
© 2017 ACM. 978-1-4503-4896-6/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3091478.3091483

user's opinions, and the *divergence* of the opinions of the entire set of users. We make these concrete in the form of specific metrics.

Our results show a surprisingly subtle interaction between properties of recommender system algorithms, actions of users, and initial system state. We show that small differences in algorithms (e.g., whether an algorithm returns the "best" item or merely a "good" item) have a strong effect on whether user opinions undergo simplification as they evolve. We show that the initial distribution of opinions can determine whether users become simplified over time. And we show that recommender systems can cause user opinions to diverge and simplify, but generally only if the recommender system accurately predicts user preferences. On the other hand, we show that a recommender system does not necessarily increase the intensity of user opinions, as long as the system makes good recommendations to the user. We conclude that there is striking richness of interactions that are observed when user opinions and recommender systems form a dynamical system, despite the high level of abstraction necessary for a study like ours. Furthermore, the nature of the effects observed suggests that the increasing prevalence of recommender systems deserves attention and care.

2 FRAMEWORK

We formalize the problem as follows. We consider a system of n *users* and m *items*. Items are objects over which the user has an opinion or has a connection with, e.g., products, movies, books, or news articles.

The knowledge used by the recommender system is held in an $n \times m$ matrix $\mathbf{M}$. In link-based systems, entries in $\mathbf{M}$ are in $\{0, 1\}$, while in ratings based systems entries in $\mathbf{M}$ are in $\mathbb{R}$. In ratings systems, $\mathbf{M}$ is only partially known. In that case, let Ω denote the index set $\{(i_1, j_1), (i_2, j_2), ...\}$ of known (observed) entries of $\mathbf{M}$ and $\bar{\Omega}$ the index set of unknown entries of $\mathbf{M}$. We use $\mathbf{M}^{\Omega}$ to denote the known entries of $\mathbf{M}$.

In a ratings-based system, a matrix completion algorithm decomposes $\mathbf{M}$ into matrices $\mathbf{X}$ and $\mathbf{Y}$ (k by n and k by m respectively) for a given latent space dimension k such that $\mathbf{C} = \mathbf{X}^T\mathbf{Y}$ and $||\mathbf{M}^{\Omega} - \mathbf{C}^{\Omega}||_2$ is minimized.

We model a user's set of preferences or opinions as an s dimensional vector $\mathbf{u_i} \in \mathbb{R}^s$. We think of this vector as the location of the user in a "preference space." This abstract representation can capture a variety of user characterizations. For instance, in a link based system, (such as specified in [8]) the user vector can be interpreted as a distribution over categories of items to which the user is linked. On the other hand, for a ratings based system using matrix completion $\mathbf{u_i}$ can represent the completed vector of items opinions, i.e., $\mathbf{u_i} = \mathbf{c_i}$ (i-th row of $\mathbf{C}$) or it can represent the projection of the user's ratings in the latent space, $\mathbf{u_i} = \mathbf{x_i}$ (i-th row of $\mathbf{X}$).

Dynamical System. The process we study in this paper is a dynamical system. Hence we introduce time indexing, where $\mathbf{M_t}$ and $\mathbf{u_{it}}$ denote respectively the state of $\mathbf{M}$ and $\mathbf{u_i}$ at time t. The conceptual approach we take is described abstractly as follows.

Let $f(i, \mathbf{M})$ represent a recommender system algorithm; f returns the next item r suggested for user i. Let $g(i, r)$ represent the response of the user to the recommendation; this can be either a rating for the item r or the decision to link to item r (e.g., to purchase item r). This action then provides additional knowledge for the recommender system. Hence after a user has responded to an item, $\mathbf{M}_{t+1}(i, r)$ is set to $g(i, r)$, and Ω is updated to include (i, r).

In this setting we are interested in the dynamical system $\{\mathbf{M}_t\}, t = 0, 1, \ldots$ whose dynamics are governed by

$$\mathbf{M}_{t+1}(i, r) = g(i, f(i, \mathbf{M}_t)).$$

Denote by *Pref*$(i, \mathbf{M})$ the function that computes the user preference vector $\mathbf{u_i}$. The principal characterization of the system we use is via $\mathbf{u}_{it} = \mathit{Pref}(i, \mathbf{M}_t)$.

This general framework allows for a wide variety of investigations. To assess the impact of recommender system algorithms, we consider two different f functions. The two f functions return either the top rated, or a randomly chosen, item for that user. We also consider a range of g functions, which reflect the degree to which whether users tend to be favorably, or even unfavorably, influenced by the recommended items.

Metrics. Previous opinion formation studies have modeled user opinions as scalar values, leading to a one-dimensional representation of user preferences [8]. Nonetheless, empirical studies show that user opinions are better characterized as occupying a higher-dimensional space, e.g., 20 to 40 dimensions [4]. Hence the set of user preferences can be viewed as cloud of points, which we consider to be centered at the origin.

While one-dimensional views of user preferences lead to the single metric of *polarization* to describe opinion dynamics, the more realistic multidimensional view of user preferences provides the basis for a more diverse set of relevant metrics. We introduce three definitions that generalize polarization in different ways to describe how user preferences may evolve over time.

First, we define **intensity** as a per-user metric that captures the strength of a user's preferences. In our point-cloud view, intensity could be conceived of as the distance of the user point from the origin. Hence, given a user i at point $\mathbf{u_{it}}$ at time t, if at time $t' > t$, we have $||\mathbf{u_{it'}}|| > ||\mathbf{u_{it}}||$, we say the user's intensity has increased.

Second, we define **simplification** as a per-user metric that captures the diversity of items that the user prefers, i.e., the spread of user preferences as a distribution. Hence simplification consists of a reduction in the entropy of the user's vector as a distribution, $H(\mathbf{u_i})$. For a given user i

$$H(\mathbf{u_i}) = -\sum_{j=1}^{s}\left(\frac{\mathbf{u_i}[j]}{\sum_{l=1}^{s}\mathbf{u_i}[l]}\right)\log_s\left(\frac{\mathbf{u_i}[j]}{\sum_{l}^{s}\mathbf{u_i}[l]}\right)$$

Then, for a user i at point $\mathbf{u_{it}}$ at time t, if at time $t' > t$, we have $H(\mathbf{u_{it'}}) < H(\mathbf{u_{it}})$, we say the user's opinions have undergone simplification. We also use the term *diversification* as the opposite of simplification, i.e., an increase the entropy of the user's preference vector.

Finally, we define **divergence** as a property of a set of users that captures the similarities among the users' preference vectors, i.e., the degree two which any two user's preferences are alike. To measure this we use the average correlation coefficient over all pairs of user preference vectors, $\bar{\rho} = \frac{2}{n^2-n}\sum_{i>j}\rho(\mathbf{u_i}, \mathbf{u_j})$ where ρ is the standard correlation coefficient. Then, if we have for a set of users at time $t' > t$ that $\bar{\rho}_{t'} < \bar{\rho}_t$ we say that the user set has increased divergence.

3 LINK BASED SYSTEMS

A large class of recommender systems can be abstracted using a graph; we call these *link based* systems. A link based system is modeled as a bipartite graph $G = (V_1, V_2, E)$, where nodes in V_1 represent users, nodes in V_2 represent items, and E is the set of edges, i.e., connection among those nodes. In such a system the recommender algorithm is a function f that takes as input G and a node $i \in V_1$ and outputs a node $j \in V_2$. The representation of those connections varies with the specificity of each system. Edges can be unweighted to simply represent viewing of a video or purchase of an product such as at [8], or weighted edges representing scores or ratings such as at [7].

In this section we study three abstractions of link based systems introduced in [8]. Our goal is twofold: first, we use these as a comparison case for our study of ratings based systems in the next section; and second we seek to extend and probe the limits of the analysis performed in [8].

3.1 Polarization: theoretical analysis

In [8], the authors investigate whether certain link based recommender systems have a polarization effect, i.e., whether the recommender system dynamics result in an increased divergence of opinions.

The authors analyzed three random-walk based recommender algorithms inspired by well-known algorithms from literature: SALSA [16] (*SimpleSALSA*), Item-based Collaborative Filtering [17] (*SimpleICF*), and Personalized PageRank [21] (*SimplePPR*) , described in more detail below. We follow their framework, in which items have labels $l \in$ {"RED", "BLUE"}, and there are an equal number of items of each label. That is, $|V_1| = n$ and $|V_2| = m = 2w$ with w items of each label.

These analyses considered cases in which users respond either with or without what was termed *biased assimilation.* In that context biased assimilation specifically means that the probability that a user i accepts an item recommendation is proportional to the quantity of items that i has of that label. On the other hand, without biased assimilation the probability that a user accepts a given recommendation is label-independent. Specifically, let x_i be the fraction of "RED" items owned by i. A recommender algorithm is polarizing with respect to i if: (1) when $x_i > \frac{1}{2}$ the probability that than a recommended item accepted by user i is "RED" is greater than x_i, and (2) when $x_i < \frac{1}{2}$, the probability that the recommended item accepted by user i is "RED" is less than x_i.

The authors in [8] conclude through analysis that *SimpleSALSA* and *SimpleICF* are polarizing only if users respond with biased assimilation; in contrast, *SimplePPR* is always polarizing. That analysis includes three assumptions: (1) the number of "RED" and "BLUE" items is equal; (2) the set of items is arbitrarily large (i.e., they study the system properties as $m \rightarrow \infty$); and (3) the recommender system may recommend the same item multiple times, even if the user has already linked to it. The third assumption is not explicitly stated but is implicit. One of the goals of this section is to show that analysis of these algorithms are quite sensitive to these assumptions and that as a result, conclusions about polarization are necessarily more nuanced.

3.2 Model

We encode the link based model in a binary matrix $\mathbf{M}$ such that $\mathbf{M}(i, j) = 1$ iff there exists an edge $e_{ij} \in E$ that connects $i \in V_1$ to $j \in V_2$. Once that there are equal quantities of items of each label we fixed the item label with respect of its position in $\mathbf{M}$, for instance, the first $\frac{m}{2}$ items have "RED" label and consequently last $\frac{m}{2}$ items have label "BLUE".

The recommender algorithms studied in [8] were defined in terms of a random walk on G. However, analysis becomes clearer if we express the algorithms in terms of a Markov chain. Let $\mathbf{P}_{\mathbf{V_i},\mathbf{V_j}}$ be the transition matrix between elements $i' \in V_i$ to $j' \in V_j$. So, $\mathbf{P}_{\mathbf{V_1},\mathbf{V_2}}$ is n by m matrix can be calculated from $\mathbf{M}$ such that each element $p_{ij} = \frac{m_{ij}}{\sum_j m_{ij}}$ and $\mathbf{P}_{\mathbf{V_2},\mathbf{V_1}}$ is m by n matrix such that $p_{ij} = \frac{m_{ji}}{\sum_i m_{ji}}$. Since G is bipartite $\mathbf{P}_{\mathbf{V_1},\mathbf{V_1}} = 0$ and $\mathbf{P}_{\mathbf{V_2},\mathbf{V_2}} = 0$. Thus:

$$\mathbf{P} = \begin{bmatrix} \mathbf{0} & \mathbf{P}_{\mathbf{V_1},\mathbf{V_2}} \\ \mathbf{P}_{\mathbf{V_2},\mathbf{V_1}} & \mathbf{0} \end{bmatrix}$$

We can now rewrite the random-walk based recommender algorithms as Markov chain based algorithms as follows.

Algorithm 1 *SimpleSALSA* using transition matrix $\mathbf{P}$

```
Require: M and a user $i$ .
Compute $\mathbf{P}^3$, a three-step transition on the transition matrix $\mathbf{P}$
Choose an item $j$ according to the distribution $\mathbf{P}^3(i, \cdot)$.
Return $j$
```

Algorithm 2 *SimpleICF* using transition matrix $\mathbf{P}$

```
Require: M and a user $i$.
Compute $\mathbf{P}^2$, a two-step transition on the transition matrix $\mathbf{P}$
Choose $k$ according to the distribution $\mathbf{P}(i, \cdot)$
Compute $j = \arg\max_s \mathbf{P}^2(k, s)$
Return $j$
```

Algorithm 3 *SimplePPR* using transition matrix $\mathbf{P}$

```
Require: M and a user $i$.
Compute $\mathbf{P}^3$, a three-step transition on the transition matrix $\mathbf{P}$
Compute $j = \arg\max_k \mathbf{P}^3(i, k)$
Return $j$
```

Note that none of these algorithms are prevented from recommending an item to which the user has already linked. While some systems such as music recommenders may suggest items multiple times, for many other systems it is not desirable to recommend items that have already been purchased or viewed (e.g., books, movies, news articles).

To study the case in which repeated recommendation of an already-linked item is not allowed, we adjusted the above algorithms during simulation. In each case we simply ensured that the algorithm did not return an item that had been already linked by the user. Denoting Ω as the set of items already linked, we ensure that each item k such that $(i, k) \in \Omega$ is removed from the distribution used in *SimpleSALSA* and eliminated from consideration in the arg max computations in *SimpleICF* or *SimplePPR*. In such cases, to ensure that a recommendation is always possible, we replace the zeros in $\mathbf{P}$ with a small value $0 < \epsilon \ll 1$.

3.3 Simulation and Analysis

We would like to understand the role of each factor – recommender algorithm, user behavior, and initial system settings – in the formation of user opinion in the linked model G.

For this, we use simulation consisting of the following steps: (i) First, initialize $\mathbf{M}$ according to a probability distribution $q(\cdot)$; (ii) Provide recommendations for all users according to Algorithm 1, Algorithm 2 or Algorithm 3; (iii) For each user, accept the suggested recommendation with some probability p, where p can be a fixed probability (when users respond without biased assimilation), or can vary with the fraction of items with same label that the user has (for biased assimilation). (iv) The matrix $\mathbf{M}$ is updated synchronously with the accepted items . The system evolves (repeating steps (ii) to (iv)) for $T = 1000$ timesteps. We repeat each simulation 30 times and report confidence intervals.

For simulations in this section we define user preference $\mathbf{u_i}$ as a vector consisting of the distribution c_{li} of items the user i is linked to over all labels l. We use two metrics: (i) average user entropy, i.e. $E[H(\mathbf{u_i})] = \frac{1}{n}\sum_i H(\mathbf{u_i})$ where $H(\mathbf{u_i}) = -\sum_{l=1}^{s} c_{li}\log_s(c_{li})$ (ii) the system entropy $H(\mathbf{U})$, where

$$H(\mathbf{U}) = -\sum_{l=1}^{s}\left(\frac{\sum_i c_{li}}{n}\right)\log_s\left(\frac{\sum_i c_{li}}{n}\right).$$

Hence the property we study is simplification (as defined above) which in this case measures the tendency for users to link primarily to one label type versus the other. We use simplification rather than polarization because it measures the degree to which user opinions become more extreme in cases where there may be more than two labels, and hence is more general than polarization.

3.3.1 Case Study 1: including already-linked items. Our first set of results studies the case in which the system is allowed to recommend an already-linked item. The settings simulated were 4700 users and 3700 items. Each user was initialized with an average of 40 known items uniformly distributed between labels. Those settings are inspired in by **MovieLens** [23] dataset. The label-independent acceptance probability for non-biased user responses was $p = 1.0$, i.e., the user always accepts the recommended item.

Figure 1a and Figure 2a show the simplification effects for Case Study 1 over 1000 steps. The figure shows that none of the 3 algorithms show significant variation over time, with or without biased assimilation. This is surprising because the theoretical analysis of [8] suggests that *SimpleSALSA* and *SimpleICF* should have distinct behavior (not polarizing and polarizing) when comparing label-independent and biased assimilation.

However, a closer look at those simulation outcomes revealed that they were mostly outputting the same recommendation over time.

A simple inspection of the *SimpleICF* and *SimplePPR* algorithms and their respective random process reveals that without constraints preventing the repetition of the same item, and given a finite number of users and items, the selection step (through the arg max computation at line 2) of *SimplePPR* and *SimpleICF* is likely to always return the same item; consequently few or no updates are made on $\mathbf{M}$, explaining the constant entropy measured. In contrast, when an infinite number of items and users is considered a higher randomness in the selection step is expected. Therefore,the chances of return the same item is reduced.

This leads to two conclusions: (1) the simplification (or polarization) effects of a real (finite) recommender system are different than those of a idealized infinite system; and (2) the previous conclusions that *SimpleSALSA* and *SimpleICF* can be non-polarizing in some cases need to be re-examined for realistic systems.

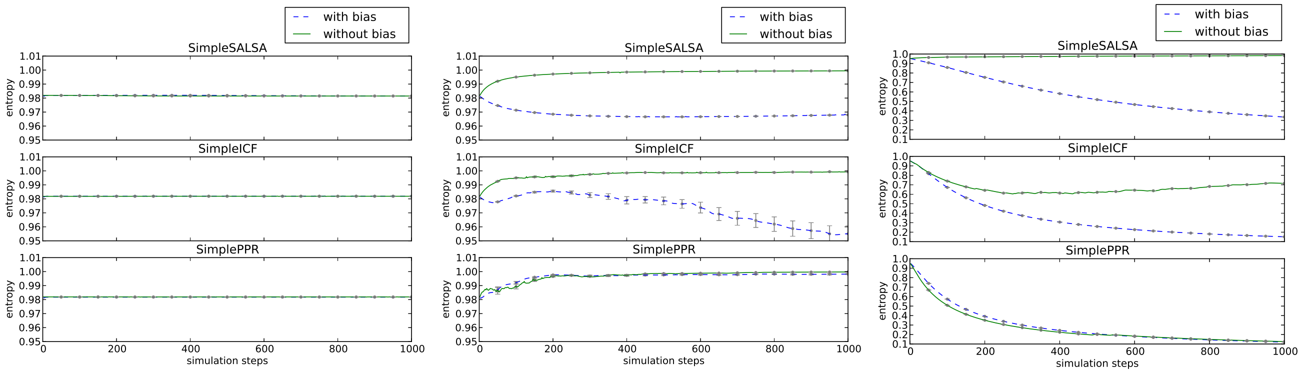
3.3.2 Case Study 2: no already-linked items. Next, in Case Study 2 we look at how results change when the system is not allowed to repeat already-linked items. In all other respects, simulation settings are the same as Case 1.

The simplification of user preference in Case Study 2 is shown in Figure 1b. The figure shows that (i) without biased assimilation all algorithms have a diversification effect; (ii) with biased assimilation *SimpleSALSA* lead to simplification, *SimpleICF* has an initial diversification followed by a simplification effect and, *SimplePPR* leads to a diversification effect. This shows that when linked items are not repeated, results are very different from those in Case Study 1, and from the systems analyzed in [8].

We also analyze for Case Study 2 the simplification of the system, i.e. the diversity of the combined preferences of all users as it evolves over time. Figure 2b shows that for*SimpleSALSA* and *SimplePPR* the system has a similar and almost constant effect regardless of whether users respond in a label independent way or with biased assimilation. However, *SimpleICF* suffers a simplification effect when there is biased assimilation.

The diversification effects for biased assimilation in Figure 1b for the initial steps of *SimpleICF* and *SimplePPR* may seem counterintuitive, but they can be understood as resulting from the less-diverse set of recommendations that those algorithms provide to the user. These algorithm provide less diverse recommendations as a result of the arg max step that each employs. We note that the most well connected items are those most likely to be recommended because of their higher values in $\mathbf{P}^2$ or $\mathbf{P}^3$. Although initially there is on average the same quantity of items of each label there are also well connected items of both labels due to randomness in the connection pattern. As a result, the lower randomness of recommendations in *SimpleICF* and *SimplePPR* often results in the system recommending the same item repeatedly until the user accepts it. This leads initially to each user linking to items with both labels approximately equally, regardless of whether the user's assimilation is biased or unbiased. However, after some number of simulation steps (approximately 200 steps) the increase of density leads to an increase of the randomness output from *SimpleICF.* I that case, biased assimilation is able to cause a user to acquire more items of one label than another, resulting in simplification of user preference. Furthermore the greater randomness of *SimpleSALSA* causes a similar biased consumption effect resulting in simplification of individual users' opinions. This is further confirmed by the fact that the overall system sees no change in the label distribution as shown in Figure 2b under *SimpleSALSA*.

In summary, this section shows that the tendency of user preferences to simplify or diversify depends on a delicate interplay of the diversity of items suggested by the recommender system and the extent to which users exhibit biased assimilation.

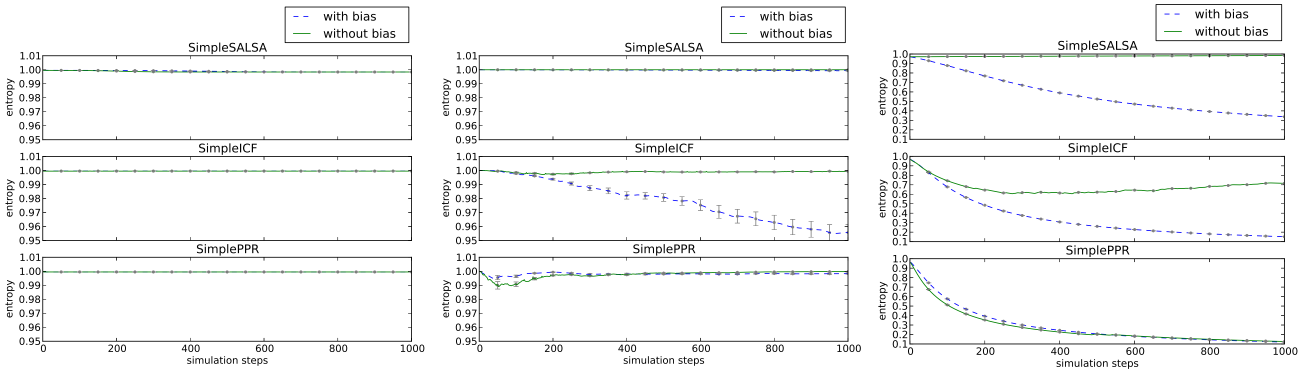


(a) Case Study 1

(b) Case Study 2

(c) Case Study 4

Figure 1: Average Entropy: User's Preference



(a) Case Study 1

(b) Case Study 2

(c) Case Study 4

Figure 2: Average Entropy: System' Preference

3.3.3 *Case Study 3: Variations.* To test whether our results in Case Study 2 are representative, we vary the simulation parameters from Study Case 2. In particular, we study the effects of varying;

(1) The percentage of items to which each user is initially linked. Previous cases used 1.08%, here we also explored 10%;
(2) The label-independent probability of accepting an item in the unbiased biased case. Previous results used $p = 1.0$, here we range p in $\{0.6, 0.8\}$.

None of the above variations affected significantly the trends results presented in Study Case 2, regardless of recommender algorithm or biased/unbiased assimilation. Rather than presenting all results in full, we summarize their similarity by presenting statistics in Table 1. The table shows the coefficients of linear regression lines fit to each individual algorithm and response variation. Specifically, Table 1 presents the slopes (noted as s) and intercept points (noted as i) for algorithms *SimpleSALSA*, *SimpleICF* and *SimplePPR* (noted respectively as $f = \{1, 2, 3\}$). In the Table the percentage of initial items is denoted as $q(\%)$, the number of users is n, the number of items is m, and the user response probability or biased response indicator is denoted as $g(p)$. Across almost all variants in Table 1 we observe similar slopes and intercept points.

Table 1: Case Study 3: Comparison over parameter variations.

n	m	$q(\%)$	f	$g(p)$	slope	intercept
4700	3700	1.08	1	0.6	8.4e-06	0.99
4700	3700	1.08	2	0.6	8.4e-06	0.99
4700	3700	1.08	3	0.6	6.8e-06	0.99
4700	3700	1.08	1	0.8	7.5e-06	0.99
4700	3700	1.08	2	0.8	7.8e-06	0.99
4700	3700	1.08	3	0.8	1.3e-05	0.99
4700	3700	1.08	1	1.0	6.6e-06	0.99
4700	3700	1.08	2	1.0	6.7e-06	0.99
4700	3700	1.08	3	1.0	1.1e-05	0.99
4700	3700	1.08	1	bias	-5.3e-06	0.98
4700	3700	1.08	2	bias	-3.2e-05	0.99
4700	3700	1.08	3	bias	7.4e-06	0.99
4700	3700	12	1	1.0	9.5e-07	0.99
4700	3700	12	2	1.0	5.7e-07	0.99
4700	3700	12	3	1.0	1.0e-06	0.99
4700	3700	12	1	bias	-2.1e-07	0.99
4700	3700	12	2	bias	-7.2e-07	0.99
4700	3700	12	3	bias	7.4e-07	0.99

3.3.4 *Case Study 4: non-uniform initialization.* The case studies above demonstrate how different algorithms and user responses evolve in a system in which links to items of different labels are initially uniformly-distributed. We undertake Case Study 4 in order to better understand the role played by the initial distribution of item labels. The setting of Case Study 4 is the same as Case Study 2 except for the initialization step. Each user is still linked on average to 40 items but on average 60% of those links are to items with the first label ("RED") and 40% are to items with the second label. Figure 1c and Figure 2c shows the simplification effects over time at the user and system level respectively when there is a uneven starting link distribution and no repeat items are allowed. The figures show first of all that the degree of simplification in this case is dramatically larger than in any of the balanced-link cases. This shows the very strong effect that the initial preferences of users have on the evolution of the system. Second, the figures show that the high levels of simplification are visible both at the user level and at the system level. Overall, the fraction of links that are to RED items grows dramatically at the system level. Third, with respect to the behavior of various algorithms, we note the following. Without biased assimilation *SimpleSALSA* still has a diversification effect on both user's and system level effect measured by the increase of average entropy. However when the user's response is biased the *SimpleSALSA* results in significant simplification at both the user and system level. On the other hand, *SimpleICF* and *SimplePPR* always result in very signficant simplification at the user and system level regardless of the user's response behavior.

The dramatic change of behavior of *SimpleICF* and *SimplePPR* when starting preferences are unbalanced is once more explained by the selection step (through the arg max computation at line 2). When there are initially more links to items of one label the most well connected items are also from that label, creating a cycle of same-label recommendations.

3.4 Discussion

The failure of the theoretical analysis to consistently match any set of results in our case studies reveals how sensitive the analysis of system dynamics in [8] is to assumptions and initial state. This can be seen intuitively as well. For example, the operation of *SimpleSALSA* is just to take three steps in a Markov Chain starting from a random state. Intuitively this results in an output vector that interpolates between the initial state and the steady state of the chain. The extreme symmetry of the initial system setup in [8] suggests that the steady state is the uniform distribution (considered separately over users and items). Creating a new link according to this distribution will move the system to an ever more uniformly connected state – hence the decrease in polarization shown in Figure 1a and Figure 2a which agrees with the analytical results in [8]. However, the same analysis suggests that the analytical results are strongly dependent on the initial linking pattern, as an imbalance in that pattern will result in an imbalanced steady state, and hence the addition of links in an imbalanced fashion as well.

Our analysis also shows the key role played by diversity of recommendations. Both *SimpleICF* and *SimplePPR* attempt to provide a notion of "best" recommendation through the use of the arg max step. On the other hand, *SimpleSALSA* only tries to provide a "good" recommendation (with high probability). While providing the "best" recommendation may seem more optimal in some sense, *SimpleSALSA* shows a much smaller tendency to simplify user preferences, particularly when user preferences start out in an imbalanced state (which seems the likely case in practice).

4 RATINGS BASED SYSTEMS

The link based systems considered in the previous section are relatively easy to analyze and interpret. However, in many recommender systems the relation between a user and an item goes beyond a simple binary connection and is expressed in some form of numerical rating. We term such systems as *ratings based systems*. A recommender algorithm in that scenario aims to predict the ratings of the unevaluated items – i.e., anticipate how the user will evaluate the remaining unevaluated items.

One of the main methods used by rating based systems is to provide recommendations using latent factor models [24]. Latent factor models estimate the ratings relations between users and items by modeling each in a latent space. The latent vectors are are learned from the data. Ratings are then estimated using inner product of user and item vector in the latent space. This estimation process can be also understood as *matrix completion*.

In a system where user preferences are real-valued, there are a wider array of metrics that are important and can be considered. Our goal in this section is to understand not just how simplification evolves (as in the last section) but also how intensity and diversity evolve in time.

4.1 Model

Let $\mathbf{M} \in \mathbb{R}^{n \times m}$ be a ratings matrix of n users over m items, λ_{MIN} be the minimum ratings value, λ_{MAX} be the maximum ratings value, and a the completion algorithm that decomposes $\mathbf{M}$ into the factors $\mathbf{X}$ and $\mathbf{Y}$ where $\mathbf{C} = \mathbf{X}^\mathbf{T}\mathbf{Y}$.

Once $\mathbf{C}$ is computed, a typical way to provide a recommendation r to a user i is using an algorithm f that recommends the unevaluated item $C(i, j)$ with highest predicted rating. We denote that algorithm as *RecBEST*; it is described below (Algorithm 4).

As a comparative case we define *RecRAN* (Algorithm 5) to be the recommender algorithm f that simply recommends a random unevaluated item.

Algorithm 4 RecBEST using **C**

```
Require: C, $\Omega$ and a user $i$ .
Compute $j = \arg\max_j C[i, j]$ such that $(i, j) \in \bar{\Omega}$.
Return $j$
```

Algorithm 5 RecRAN using **C**

```
Require: C, $\Omega$ and a user $i$ .
Choose randomly $j$ such that $(i, j) \in \bar{\Omega}$.
Return $j$
```

We seek to capture a range of possibilities for the response of a user to a recommendation. These essentially reflect how a user's opinion changes in response to evaluating an item (e.g., viewing a movie or reading a book). We model the user response $g_x(i, r)$ as a probabilistic function of $x \in [0, 1]$. The user signals her evaluation by providing either a rating of λ_{MAX} or λ_{MIN}. The parameter x determines how often a user tends to be positive about a recommended item. That is, for a given user i and a recommendation r, $\mathbf{M}$ will be updated accordingly to $g_x(i, r)$, i.e., $M(i, r) = g_x(i, r)$, where:

$$g_x(i,r) = \begin{cases} \lambda_{MAX}, & \text{with probability } x \\ \lambda_{MIN}, & \text{with probability } (1-x) \end{cases} \quad (1)$$

4.2 Datasets

The previous section showed that system dynamics can be strongly influenced by the initial system state. Hence we conclude that it is important to initialize the system in a realistic manner. As a result we use previously captured datasets as the system initialization in this section. We use three datasets that include ratings relations between users and items to define the starting point matrix **M** and initial Ω.

The **MovieLens** dataset is collected from a non-commercial web movie recommender [23]. We selected a relatively dense subset of this dataset, consisting of users that have at least 40 ratings, which we denote as `dataML`. In total `dataML` has 4736 users, 3706 movies, and 962682 ratings. Additionally we scaled the ratings to center them at zero, changing the initial range from [1:5] to [-2:2].

The **MovieTweetings** dataset is collected from well-structured movie evaluation tweets on Twitter from 2013 until 2015 [9]. We selected a relatively dense subset of this dataset of movies that have at least 10 ratings and users that have at least 40 ratings, which we denote `dataMT`. This dataset has 2604 users, 3703 movies and 218302 ratings. We centered the ratings in `dataMT` by rescaling them from [1:10] to [-5:5].

The **BookCrossing** was collected in a 4-week crawl from the Book-Crossing community [28]. We used a relatively subset of this data excluding null inferred ratings and selecting user with at least 40 ratings over books with at least 5 ratings, which we denote `dataBX`. This dataset has 294 users, 2764 books, and 20040 ratings rescaled from [1:10] to [-5:5].

4.3 Simulations

We again study the closed-loop dynamics between recommender system and users in simulation.

In our simulations all matrix completion procedures were performed using LMaFit [26] with a latent space $k = 20$. We define the user preference vector as the completed vector of item opinions ($\mathbf{u_i} = \mathbf{c_i}$) and observe how $\mathbf{u_i}$ evolves over time through an individual dynamic simulation, i.e., just one user evolves in time while other user ratings are not changed. We observe how the user preference evolves in the dynamical system at an individual level (through intensity and simplification) as well as at a system level (through divergence).

We analyzed the recommender algorithms $f \in \{\mathit{RecBEST}, \mathit{RecRAN}\}$. over the probabilistic users response $g_x(i,r)$ for $x \in \{0.0, 0.1, \ldots, 1.0\}$. Using those settings we evolved each user for $T = 400$ iterations where at each step a new recommendation was made and evaluated by the user according to rating policy g_x. Furthermore some individual and collective metrics were computed from $\mathbf{u_i}$ at each step. This framework for our analysis is described in Algorithm 6.

Algorithm 6 Ratings based dynamical system

```
Require: A partially-observed matrix of ratings $\mathbf{M}^{\Omega}$; a recommender system algorithm $f$; a user response function $g_x$; a preference space size $k$; and a number of iterations $T$.
for $i \in$ Users do
    $\mathbf{R} = \mathbf{M}$
    for (step=1:T) do
        $(\mathbf{X}, \mathbf{Y}) = \mathrm{LMaFit}(\mathbf{R}, k, \Omega)$
        $\mathbf{C} = \mathbf{X}^{\mathbf{T}}\mathbf{Y}$
        $j = f(\mathrm{C}, i, \Omega)$
        $\mathbf{R}(i, j) = g_x(i, j)$
        $\Omega = \Omega \cup (i, j)$
        Output Intensity and Simplification of $\mathbf{c_i}$
        $S(\mathrm{step, i}) = \mathbf{c_i}$
    end for
end for
for (step=1:T) do
    Output Divergence of $S(\mathrm{step})$
end for
```

4.3.1 Intensity. First we consider how the intensity of the user preference vector varies over time and how the choice of user response and recommender system influence in those changes.

Figure 3 captures the intensity measures computed for all of our simulations. Each square from each plot of Figure 3 represents one metric observation, while the color of the square indicates the metric value as indicated in the color map. Furthermore, the x-axis in each plot represents the simulation step (ranging from 1 to 400) and, the y-axis in each plot represents the probability with which the user response g_x was set. Thus each horizontal set of points represent one particular simulation. Each plot of Figure 3 represents a set of simulations results from a given algorithm and dataset. Different set of plots rows correspond to the dataset used to initialize M – respectively `dataML` , `dataMT` or `dataBX` . The first two columns of plots group simulation results from *RecBEST* and *RecRAN* respectively. The last column shows the pointwise subtraction of *RecRAN* from *RecBEST* for comparative purposes.

The reason for the last column of plots is that the *RecRAN* simulations measure the effects on user opinion when randomly chosen items are evaluated by the user according to the rating policy g_x. Thus, this case can be considered to capture the effect of a user viewing and rating items without the influence of a recommender system, but rather in completely random fashion. Therefore the subtraction of the simulations results (*RecBEST* - *RecRAN*) – noted in this work as *BEST-RAN* – is a measure of influence of the recommender system *algorithm* on user opinion.

All the intensity observations represented at Figure 3 were computed by averaging the norm of the users preference vectors for a given time step.

The figure show a number of results. First, for both algorithms (*RecBEST* and *RecRAN*) and for all datasets, regardless of the user's response, we can observe a general increase in intensity over time – marked by the vertical increase of average norm value when the time color indicators also increases.

Second, for *RecBEST* (first column) the increase in intensity of opinion is maximized for probabilities x intermediate between 0 and 1. The *RecBEST* system increases intensity the least when $x = 1$, the user always agrees with the system and rates the items presented highly.

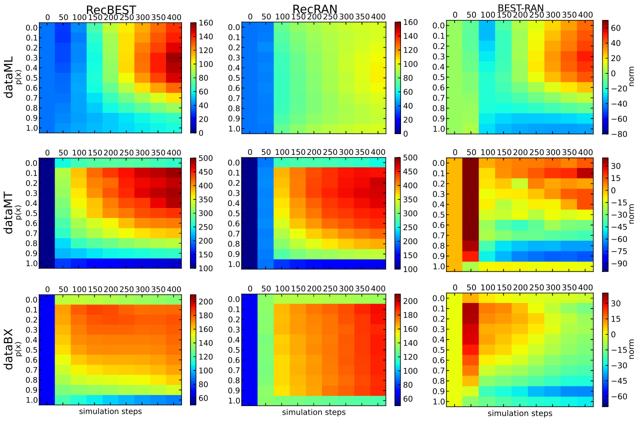


Figure 3: Average Norm of User's Preference

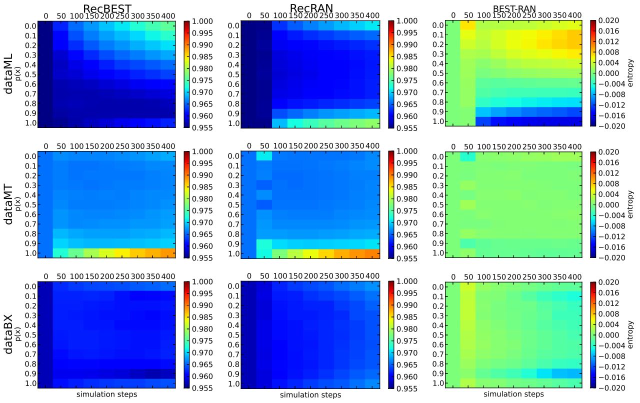


Figure 4: Average Entropy of User's Preference

These two effects together mean that for *BEST-RAN* (third column) there is a generally decreasing relationship between the user opinion intensity and the degree to which the user agrees with the recommender algorithm. In other words, when the user agrees with the recommendations made by the system, the user's opinion intensity increases, but the increase is lower than if there had not been a recommender system in the loop. On the other hand, when the user disagrees with the recommender system, the user's intensity of opinions increases more than if there had been no recommender system in the loop. We conclude from this that a recommender system does not necessarily increase the *intensity* of user opinions, as compared to random recommendations, as long as the system makes good recommendations to the user.

4.3.2 Simplification. Our second analysis concerns how the user's preference vector behaves over time as a distribution, which is captured as simplification (or diversification) of opinions. Figure 4 compiles the results from the simplification measures of the user preference vector. The figure uses the same representation for squares, colors, plots and plot positions as Figure 3.

The figure shows a number of effects in opinion dynamics. First of all, for both algorithms (*RecBEST* and *RecRAN*) and for all datasets regardless the user response, we can observe a general increase of entropy over time. In another words, user opinions get more diverse under the influence of either a recommender system or a random presentation of items.

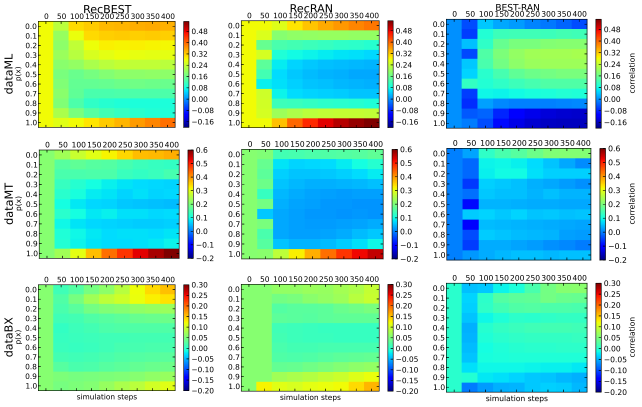


Figure 5: Average Correlation Coefficient of User's Preference

Second, the influence of dataset (i.e., system initialization) has a much stronger effect on diversity (as compared to, eg, intensity). The changes in diversity are highly varied across datasets (although diversity always increases).

However, the third column (*BEST-RAN*) shows that the relative influence of the recommender algorithm *RecBEST* over users for `dataML` and `dataMT` is that when y increases, the average entropy decreases. This means that, when the user agrees more with the recommendations its opinions become simpler, as compared to a random presentation of items. Even for `dataBX`, where there is no strong correlation between the influence of the recommender system and the user response g_x, user opinion always becomes simpler under the influence of the recommender system than under a random item presentation.

These results emphasize the importance of comparing the effect of a recommender system to an alternative. While a dynamical system involving a recommender system tends to result in diversified user opinions, the same is true of a dynamical system that does not involve a recommender system. Only by comparing the two do we see how the recommender system decreases the diversity of user opinion.

4.3.3 Divergence. Our last analysis is with regard to how closed loop dynamics shapes the user preference vectors as a set. We want to understand the conditions under which the user preference vectors become more similar to the others, i.e. we want to observe when there is loss or increase of individuality from the user preference vectors set.

Figure 5 presents the average correlation of user preference vectors, with plots using the same representation for squares, colors, plots and plot positions as in previous figures.

We note that for both algorithms (*RecBEST* and *RecRAN*) and for all datasets, the average correlation coefficient decreases for intermediate values of x, while it increases when x is close to either 0 or 1. Thus, unless the user does not fully agree or disagree with most recommendations, the set of user opinions tends to diverge. However, the influence of the recommender system as compared to random recommendations, as shown in the third column (*BEST-RAN*) is quite different. That column shows that as the user's agreement with recommendations increases, there is a consistent increase in opinion divergence. This effect is strong enough that opinions always increase in divergence, compared to random item presentations, when the user fully agrees with the recommendations made by the recommender system.

4.4 Discussion

The conclusions from this section and the previous section together show consistencies that reinforce a number of high level conclusions. The principal conclusion concerns the tendency of recommender systems to simplify user opinions. This tendency is observed generally over both link based and ratings based systems, although it comes with some caveats. First, recommender systems that recommend the "best" item at any given time (eg, *SimpleICF*, *SimplePPR*, and *RecBEST*) have a much stronger simplifying effect than systems that return "good" recommendations according to some probability (*SimpleSALSA*). Furthermore, the simplifying effect is strongest when the recommender system does a good job of predicting user preference (e.g., when $x = 1$ in *RecBEST*).

Second, the initial state of the system when the recommender system starts has a strong effect on the eventual outcome. In the case of *RecBEST*, opinion diversity is strongly affected by the dataset used, and in the cases of *SimpleICF* and *SimplePPR* the simplification effect is highly pronounced when the system does not state in a perfectly balanced configuration.

Turning to the other metrics, our results suggest that the ratings based system tends to cause user opinions to collectively diverge and to become individually less diverse, but *only* when the system accurately predicts the user's preferences. If the system is less effective in this regard, it can cause the opposite effect, namely, the to cause user opinions to become more similar to each other, and to become individually more diverse. On the one hand, the conclusion that opinions can diverge and become less diverse is potentially concerning, but the realization that the recommender system must be quite accurate for this to happen raises questions about whether this effect is likely to occur in practice.

5 RELATED WORK

Although a number of papers have addressed the individual components of the dynamics between recommender systems and user opinion formation or the social implications of that dynamic, few of them have addressed the problem in whole.

In this section we review three works that have addressed the dynamics of opinion formation by recommender systems, and we review key prior work that has explored individual components of that dynamics.

As discussed extensively in the body of the paper, the authors in [8] explore the causes of polarization through an opinion formation process based on averaging of users opinions. They show that the opinion in the group polarizes when users have biased assimilation responses. Additionally, they expand the opinion formation process analysis to a recommender system model analyzing the biased assimilation response. As we covered in Section 3 our results generalize that work and show that in many realistic settings, the conclusions therein do not apply.

Bakshy, Messing and Adamic carry out an empirical analysis of the ideological diversity of news exposure on Facebook in [3]. They measure the diversity of news (through political alignment) shared between friends and measure the potential for cross-cutting – i.e. how much a user from a given political alignment (conservative, moderate or liberal) is exposed to news of a different political alignment. They concluded that the individual's social network itself is the most important factor in limiting their exposure to diversity. However, is the user's choices about what to consume (i.e. which links to click) more than the newsfeed ranking algorithms that contribute to the news diversity consumed. Although [3] recognized the dynamics between news feed (recommender system) and users, their analysis focuses only on the user's exposure to diversity when user opinion (political affiliation) is held constant in their model.

Koren in [15] proposes a recommender system based in collaborative filtering that incorpores temporal dynamics. Using his recommender system he splits the prediction score between factors dependent or independent of the interaction of users and items and performs an analysis of rating drifting on a Netflix dataset. Although [15] studies the dynamics of recommender systems and how much the interaction of user and items can affects the prediction score over time, his analysis is focused in the recommender system itself and doesn't explore the user's opinion formation.

Opinion Formation. Next, we turn to work that addresses individual aspects of our study. Prior work in opinion formation has taken several approaches to understand how opinions can evolve in a network according to a given model of information propagation.

One line of work examines the direct influence of neighbors and self beliefs in opinion formation; this is studied in [6] where a repeated averaging model is used to analyze consensus. An alternative approach, taken by [5] uses game theory as opinion formation process. Both works present boundaries for the cost to reach consensus in their models. Finally, in [19] Mas and Flache propose a peer-to-peer interaction model that can explain polarization of opinion with homophily and without negative influence (disliking of dissimilar others). Although these models are concerned with some of the same phenomena as our study (eg, polarization) they do not include a recommender system as an external agent.

Filter Bubble. Finally, attention has recently been drawn to the situation in which there is a narrowing of access to information by users in online systems. The focus here has primarily been on personalization and recommender engines. This effect has been dubbed a "filter bubble" by Pariser in [22]. Following this line, researchers have analyzed interactions between users and recommender systems to detect filter bubbles. The authors in [11] and [13] explore the factors that trigger personalization on results of search queries. Furthermore online price discrimination was exposed by [12] and online ad delivery discrimination was exposed by Sweeney in [25]. These are examples in which information content is filtered and throttled in a fashion that is undesirable.

The authors in [29] investigate the different types of personalization in communication and they claim that at present, there is no empirical evidence that warrants any strong worries about filter bubbles however, they agree that this could be a future problem if personalization technology improves.

Another line of work related to the filter bubble phenomenon tries to understand the role played by recommender algorithms in decreasing recommendations diversity and thus potentially limiting user experience.

Knijenburg et al in [14] claims that for solve the Filter Bubble problem it is necessary to build a recommender system with a different goal than simply recommending good items. The authors of [14] then proposes a idea of a new recommender system – named "Recommender System for Self-Actualization" – that aims to support the users in developing, exploring and understand their unique taste.

Graells-Garrrido et al in [10] proposes a content recommender on Twitter that uses graphical tools and gap indicators to stimulate diversity and connect people of opposite views. Other researchers such as [18], [1], [27] and [20] suggest the inclusion of "serendipitous" recommendations in order to promote diversification. While these suggestions are consistent with our observation that "best" recommendations are not always the most beneficial for user opinion formation, they also do not consider how opinions and recommender systems evolve together over time.

6 CONCLUSIONS

In this paper we studied the closed-loop dynamics between recommender systems and users across a wide variety of system models and configurations. We proposed three metrics – intensity, simplification and divergence – to capture important properties of user opinions as they evolve in such a system. By studying a wider range of settings, we extend previous work [8] and show that its conclusions do not always generalize to more typical settings (eg., when already-linked items may not be recommended, or when the system starts with nonuniform user preferences). Further, comparing results for link based and ratings based systems, we identify common features of recommender systems that tend to simplify user opinions. We also show that under certain circumstances, recommender systems can act to cause user opinions to diverge and become less diverse, but this is not always the case, and more study is needed to determine whether in practice the conditions necessary for divergence and simplification of opinions do actually occur.

Our work has a number of limitations that suggest the need for future study; in particular, further theoretical analysis of both link based and ratings based dynamics seems worthwhile and potentially feasible. Nonetheless, our results show that the dynamics of recommender systems over time are complex and varied, and that there is potential for such systems to affect user opinions in subtle ways. Combined with the increasing prevalence of recommender systems, our results suggest that better understanding of closed loop opinion formation is an ongoing and important problem.

REFERENCES

[1] Zeinab Abbassi, Sihem Amer-Yahia, Laks V.S. Lakshmanan, Sergei Vassilvitskii, and Cong Yu. 2009. Getting Recommender Systems to Think Outside the Box. In *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. ACM, New York, NY, USA, 285–288.

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* 17, 6 (2005), 734–749.

[3] Eytan Bakshy, Solomon Messing, and Lada Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook. *Science* (07 May 2015), aaa1160+.

[4] Robert M. Bell and Yehuda Koren. 2007. Lessons from the Netflix Prize Challenge. *SIGKDD Explor. Newsl.* 9, 2 (Dec. 2007), 75–79.

[5] Kshipra Bhawalkar, Sreenivas Gollapudi, and Kamesh Munagala. 2013. Coevolutionary Opinion Formation Games. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing (STOC '13)*. ACM, New York, NY, USA, 41–50.

[6] David Bindel, Jon M. Kleinberg, and Sigal Oren. 2012. How Bad is Forming Your Own Opinion? *CoRR* abs/1203.2973 (2012).

[7] Colin Cooper, Sang Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. 2014. Random Walks in Recommender Systems: Exact Computation and Simulations. In *Proceedings of WWW*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 811–816.

[8] Pranav Dandekar, Ashish Goel, and David T. Lee. 2013. Biased assimilation, homophily, and the dynamics of polarization. *Proceedings of the National Academy of Sciences* 110, 15 (2013), 5791–5796. arXiv:http://www.pnas.org/content/110/15/5791.full.pdf+html

[9] Simon Dooms, Toon De Pessemier, and Luc Martens. 2013. MovieTweetings: a Movie Rating Dataset Collected From Twitter. In *CrowdRec at RecSys 2013*. https://github.com/sidooms/MovieTweetings

[10] Eduardo Graells-Garrido, Mounia Lalmas, and Daniele Quercia. 2013. Data Portraits: Connecting People of Opposing Views. *CoRR* abs/1311.4658 (2013). http://arxiv.org/abs/1311.4658

[11] Aniko Hannak, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. 2013. Measuring Personalization of Web Search. In *Proceedings of WWW*. ACM, New York, NY, USA, 527–538.

[12] Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. 2014. Measuring Price Discrimination and Steering on E-commerce Web Sites. In *Proceedings of the 2014 Conference on Internet Measurement Conference (IMC '14)*. ACM, New York, NY, USA, 305–318.

[13] Chloe Kliman-Silver, Aniko Hannak, David Lazer, Christo Wilson, and Alan Mislove. 2015. Location, Location, Location: The Impact of Geolocation on Web Search Personalization. In *Proceedings of ACM IMC*. ACM, New York, NY, USA, 121–127.

[14] Bart P Knijnenburg, Saadhika Sivakumar, and Daricia Wilkinson. 2016. Recommender systems for self-actualization. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 11–14.

[15] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.

[16] Ronny Lempel and Shlomo Moran. 2001. SALSA: the stochastic approach for link-structure analysis. *ACM Transactions on Information Systems (TOIS)* 19, 2 (2001), 131–160.

[17] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE* 7, 1 (Jan 2003), 76–80.

[18] Valentina Maccatrozzo. 2012. Burst the Filter Bubble: Using Semantic Web to Enable Serendipity.. In *International Semantic Web Conference (2) (Lecture Notes in Computer Science)*, Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist (Eds.), Vol. 7650. Springer, 391–398.

[19] Flache A Mäs M. 2013. Differentiation without Distancing. Explaining Bi-Polarization of Opinions without Negative Influence. *PLoS ONE* 11, 8 (2013).

[20] Kenta Oku and Fumio Hattori. 2011. Fusion-based Recommender System for Improving Serendipity. In *1st International Workshop on Novelty and Diversity in Recommender System (DiveRS 2011)*.

[21] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank citation ranking: bringing order to the web. (1999).

[22] E. Pariser. 2011. *The Filter Bubble: What the Internet Is Hiding from You.* Penguin Group USA.

[23] GroupLens Research Project. 2003. MovieLens dataset ml-1m. http://grouplens.org/datasets/movielens/

[24] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2010. *Recommender Systems Handbook* (1st ed.). Springer-Verlag New York, Inc., New York, NY, USA.

[25] Latanya Sweeney. 2013. Discrimination in Online Ad Delivery. *Queue* 11, 3, Article 10 (March 2013), 20 pages.

[26] Zaiwen Wen, Wotao Yin, and Yin Zhang. 2012. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation* 4, 4 (2012), 333–361.

[27] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: Introducing Serendipity into Music Recommendation. In *Proceedings of the ACM WSDM*. ACM, New York, NY, USA, 13–22.

[28] Cai-Nicolas Ziegler, Sean McNee, Joseph Konstan, and Georg Lausen. 2005. Improving Recommendation Lists Through Topic Diversification. In *14th WWW*. ACM, Chiba, Japan.

[29] Frederik J. Zuiderveen Borgesius, Damian Trilling, Judith Moeller, BalĂązs Bodó, Claes H. de Vreese, and Natali Helberger. 2016. Should We Worry About Filter Bubbles? *Internet Policy Review. Journal on Internet Regulation* 5, 1 (2016), 102–114.