Preference Amplification in Recommender Systems

Dimitris Kalimeris
kalimeris@g.harvard.edu
Harvard University

Smriti Bhagat
smr@fb.com
Facebook

Shankar Kalyanaraman
kshankar@fb.com
Facebook

Udi Weinsberg
udi@fb.com
Facebook

ABSTRACT

Recommender systems have become increasingly accurate in suggesting content to users, resulting in users primarily consuming content through recommendations. This can cause the user's interest to narrow toward the recommended content, something we refer to as *preference amplification*. While this can contribute to increased engagement, it can also lead to negative experiences such as lack of diversity and echo chambers. We propose a theoretical framework for studying such amplification in a matrix factorization based recommender system. We model the dynamics of the system, where users interact with the recommender systems and gradually "drift" toward the recommended content, with the recommender system adapting, based on user feedback, to the updated preferences. We study the conditions under which preference amplification manifests, and validate our results with simulations. Finally, we evaluate mitigation strategies that prevent the adverse effects of preference amplification and present experimental results using a real-world large-scale video recommender system showing that by reducing exposure to potentially objectionable content we can increase user engagement by up to 2%.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Computing methodologies →** *Learning from implicit feedback*.

KEYWORDS

Recommender systems; echo chambers; filter bubbles; fixed point

ACM Reference Format:
Dimitris Kalimeris, Smriti Bhagat, Shankar Kalyanaraman, and Udi Weinsberg. 2021. Preference Amplification in Recommender Systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467298

1 INTRODUCTION

Recommender systems have grown to dominate how people navigate through and consume the vast amounts of content that exists on the Internet. As users interact with the recommendations they are given, they provide feedback that is used by the system to improve its understanding of their preferences, and subsequently improve the quality of future recommendations. This iterative process of fine-tuning new recommendations based on user feedback results in a feedback loop. Such feedback loops may occasionally cause negative experiences for users.

For the purpose of illustration, consider a user who is curious about videos that might be considered objectionable. If a recommender system repeatedly presents them with such videos it will likely receive positive feedback and learn about the user's preference for this content. Hence, over time the system may converge to primarily showing them objectionable videos, and most importantly, the user also might end up actively seeking and liking such content. These phenomena of progressive reinforcement of one's own views as a result of the feedback loop as well as the narrowing exposure to different types of content, have been referred to as *echo chambers* and *filter bubbles* respectively, and have received much recent attention [18, 24, 31, 38, 39]. Given the impact that recommender systems have on our lives, it is important to understand when recommendation dynamics result in echo chambers or filter bubbles, and how to mitigate any negative experiences to the user.

In this work, we study echo chambers from a theoretical point of view. We consider a matrix factorization-based recommender system and discuss the formation of feedback loops, resulting from users exhibiting a *drift* (or *affinity*) toward certain item categories recommended to them. Essentially, drift captures the intuition that a user might become more interested in an item after exposure to it, for example after reading an article they really enjoyed they might want to read more similar articles showing a shift of their preferences towards that particular article content.

The most natural solution concept that allows us to predict the long-term behavior of the dynamical interaction between the recommender system and a user is that of a *stable fixed point*, i.e. a situation in which the user preferences do not change in response to the system's recommendations. We prove that under mild assumptions such a stable fixed point does not exist at all, implying that if users experience drift towards certain item categories, repeated exposure to content from those categories could result in increasing the user's preference for them, absent any intervention. Moreover, this is true even if the prevalence of these item categories is low. We then apply this theoretical formulation in simulations involving integrity scenarios, where the item categories could be representative of problematic content. We show using both synthetic and real-world datasets that if the user has initial affinity

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 '21, August 14–18, 2021, Virtual Event, Singapore.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467298

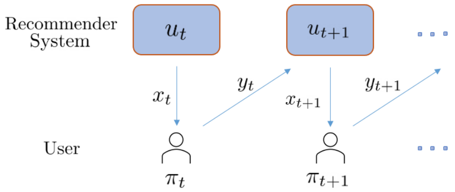

Figure 1: : Interaction between the user and the system. The system uses u_t to recommend item x_t to the user. The system then updates its parameters u_t to reflect on the feedback y_t. The user also slightly shifts their preferences π_t after interacting with the item that was recommended.

towards problematic content, the system will move towards showing more problematic content to them. We propose strategies for mitigating these outcomes, and present results from simulations as well as investigations on real-world datasets. Finally, we present supporting evidence of these strategies from a large user experiment run on a real world content recommender system and observe an improvement in metrics used to measure user engagement. In summary, our contributions are

(1) a theoretical characterization of the dynamic interactions between a user and the recommender system,
(2) evidence through simulations on synthetic and real-world data of the divergence in user preferences and the existence of echo chambers in matrix factorization systems,
(3) results from a large user experiment conducted at Facebook showing how applying personalized mitigation can increase user engagement.

1.1 Related work

The study of feedback loops and echo chambers in recommender systems has received a lot of attention [19, 23, 29] and has been catalogued in a survey by Chen et al. [8] who discuss feedback loops and present a taxonomy for biases that exist in recommender systems. In investigating how users interact with recommender systems, some earlier work has focused on the static (single snapshot) performance of the system [11, 36, 37], while others have modeled these interactions in a dynamic, sequential setting, seeking to understand the effect that the past ratings of a user have on the future content that is available to them [1, 5, 7, 12, 13, 25, 40, 41]. Along the same line of work, in [28, 32] the authors propose a general framework to predict the long-term behavior of machine learning models in cases where the model predictions directly influence the observed data. Recommender systems broadly fall in that category; however many real systems, for example matrix factorization ones, do not satisfy the requirements of their framework, thereby warranting a different approach. Echo chambers have also been extensively studied in recent years in a variety of contexts besides recommender systems, like online social networks [3, 6, 9, 10, 15] and opinion dynamics [4, 22, 33, 35], with content diversity being proposed most commonly to address user polarization [2, 21, 23, 27, 30].

Finally, the work that is conceptually closest to ours is that of Jiang et al. [20]. The authors provide a theoretical model to study echo chambers, which however, fails to capture core aspects of real recommender systems. First, they do not consider any systematic way through which the recommender system recommends items to the user (their assumption is that each item is recommended infinitely often). Second, in their model items are independent from each other in the sense that reacting to an item does not provide any information about the possible reaction to a similar item. This assumption violates the essence of how collaborative filtering works. In this work, we study similar phenomena in a more complete and realistic setting, where there is an underlying matrix factorization-based recommender system.

1.2 Outline

We formalize the model for the recommender system and the user behavior in Section 2. In Section 3, we present our main theoretical results showing the lack of stable fixed points in the user-system interaction, hinting at the existence of echo chambers in matrix-factorization systems. In Section 4, we evaluate our results through simulations on real and synthetic data. We discuss mitigation strategies in Section 4.2, and describe a real-world large user experiment applying these strategies conducted at Facebook in Section 4.3.

2 PROBLEM SETTING

Consider a collection of m items that are available in a platform. At each time step t, a recommender system presents an item to a user and receives feedback $y \in \{0, 1\}$ indicating whether they liked, clicked on, watched, etc. the item that was suggested. This feedback is utilized to improve subsequent recommendations. We provide an illustration of the user-system interaction in Figure 1 and a notation summary in Table 1.

Linear Scores. Our work focuses on stochastic matrix factorization recommender systems. In such systems, each item is assigned a linear score $s_x = u^\top x$, where $u, x \in \mathbb{R}^d$ are feature vectors for the user and for the item, respectively. Here $d \ll m$, i.e. the user and item representation is low-dimensional. Following standard logistic regression modeling, the system then predicts the probability of the user liking an item x at time t as $p_x^t = \sigma(s_x^t)$, where $\sigma(\cdot)$ denotes the sigmoid function. We assume that the item features are a static description of the item, i.e., they correspond to scores coming from some classifiers or represent fixed characteristics of the item that do not change after an interaction with the user. The user features on the other hand are dynamic and are updated based on their feedback, to reflect the true user preferences more accurately. For example, for a movie recommendation, x can describe the genre of the movie, the duration, the language, etc., while u describes how much does the user values each of these. As a result, the scores for items are time-varying: $s_x^t = u_t^\top x$ for each time step t.

Stochastic recommendation. We consider stochastic recommender systems, a relaxation of the classical `Top-k` recommender system [12]. Such a system maps the score of each item to a selection probability through the *softmax function*. Hence, higher scores correspond to higher selection probabilities. Specifically, at each iteration, an item x is chosen as $x \sim f_t$, where f_t:

$$f_t(x) = \frac{e^{\beta s_x^t}}{\sum_{x'} e^{\beta s_{x'}^t}} \tag{1}$$

Table 1: Notation Summary

Symbol	Meaning
m	number of items
$u_t \in \mathbb{R}^d$	inferred user feature vector at time t
$x \in \mathbb{R}^d$	item feature vector (static)
s_x^t	score of item x at time t: $s_x^t = u_t^\top x$
$p_x^t = \sigma(s_x^t)$	predicted probability that the user will like item x
$\pi_t(x)$	true probability that the user will like x at time t
f_t	softmax of item scores at time t
$f_t(x)$	probability of recommending x at time t
$\beta > 0$	system sensitivity, determines the "steepness" of f_t
$\gamma \in (0, 1)$	sensitivity parameter of the user
$\alpha_{t,x}$	rescaling coeff. for item x at time t (see Def. 1)

Notice that the selection probability of an item $f_t(x)$, which corresponds to the probability of presenting x to the user, is different than the probability p_x^t which is the probability that the user likes item x upon been presented with it. Using different values for the sensitivity parameter β in f_t we can induce different behaviors in the system[1]. In particular, high β results in a steep f_t where only the items with the highest scores have non-negligible probability of been recommended. In particular, if $\beta \to \infty$ then f_t is exactly a `Top-1` recommender system. On the other hand, small values of β lead to a more uniform behavior where all items have a similar probability of being presented to the user. See Section 4 for more details on the effect of β and the shape of f_t (e.g. Fig 6).

System update. Upon receiving feedback y_t for the item x_t that was recommended to the user, the system updates u_t to account for it, attempting to learn the *true user preferences* accurately. We assume that the update of u_t happens through gradient descent, a standard algorithm with strong performance guarantees [17] that is employed in practice for online learning with convex loss functions. Specifically, at iteration t, the expected update in the system is:

$$u_{t+1} = u_t - \eta_t \cdot \mathbb{E}_{x \sim f_t}\left[\nabla_u \, \ell(u_t, (x, y))\right] \tag{2}$$

where ℓ is the negative log-likelihood loss function[2] $\ell(u, (x, y)) = -y \ln \sigma(u^\top x) - (1 - y) \cdot \ln(1 - \sigma(u^\top x))$.

Remark. Notice that in (2), the gradient of the loss is inside the expectation and their order cannot be exchanged since the distribution f_t according to which x is chosen depends on u_t. This is one of the main reasons why the analysis of such a dynamical model differs from the analysis of the gradient descent dynamics in standard settings; the distribution of our data (the m items in the system in our case) is dynamically adapting and depends on the parameters of the model that we utilize in the respective iteration.

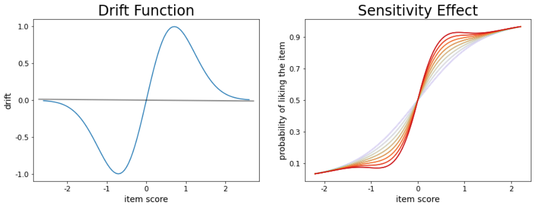

Figure 2: The shape of the drift function (on the left) indicates that the users do not reinforce their preferences for an item uniformly but rather, depending on how much they already like it. After some threshold (peak), this reinforcement fades since they already have a firm opinion for the particular item (and similar ones) and as a result it is unlikely that they will significantly alter their preferences for it. On the right we see the difference between the true and the predicted probability of liking an item, in red and blue respectively. The difference between these two is attributed to the drift function. More vivid red corresponds to more sensitive users that are more aggressive in changing their preference for an item.

Fixed points. The most natural solution concept to characterize the interaction between the user and a dynamically-adjusting recommender system is that of a fixed point, i.e., a user vector u^* such that, if the recommender system presents items according to the distribution f^* induced by it (according to (1)), there will be no more updates to the user vector u_t; the user and the system have reached equilibrium. Formally, u^* corresponds to a fixed point if:

$$u^* = \arg\min_u \mathbb{E}_{x \sim f^*, y}\left[\ell\big(u, (x, y)\big)\right]$$

Fixed points can be further categorized with respect to their long term behavior. Specifically, a fixed point u^* is called:

- *stable*, if for every neighborhood U of u^* there exists a neighborhood U' of u^* such that $u_t \in U$ for every t if $u_0 \in U'$, i.e., if we start close to u^* we will not move far from it.
- *attracting*, if there exists a neighborhood U of u^* for which $u_t \to u^*$ if $u_0 \in U$, i.e., we converge to u^* if we start close.
- *asymptotically stable*, if it is stable and attracting.

Naturally, asymptotically stable fixed points correspond to the most desirable solution since they allow us to characterize and predict the limit behavior of a time-evolving system easily. However, it is easy to see that some assumptions on the user behavior are required even for the user-system interaction to have a fixed point. Otherwise, if the user arbitrarily changes the way they interact with items in every iteration, the system will always try to adapt by updating u_t. In the next subsection, we formalize this idea by making specific assumptions on the true user preferences.

Fixed points, echo chambers and filter bubbles. Note that the notion of a fixed point is orthogonal to the ideas of an echo chamber or a filter bubble. Echo chambers correspond to a situation where the true preferences of a user are constantly reinforced leading to polarization: the user likes more of what they previously endorsed and less of what they did not. Similarly, in a filter bubble the system actively narrows down the categories of items it presents to a user. However, by definition at a fixed point there is no update on u^*, hence such reinforcement does not occur.

[1] We call β sensitivity parameter because it determines how "sensitive" will f_t be to changes of the score. The smaller the β, the more uniform its behavior.

[2] This loss is standard for classification problems with binary labels.

2.1 User behavior

The behavior of the user is determined by a preference function $\pi_t : \mathbb{R}^d \longrightarrow [0, 1]$, unknown to the recommender system. It denotes their real preference for each item at time t and is time-dependent, meaning that it can be modified based on the recommendations that the user receives or on exogenous factors. The user reacts positively (like, watch, etc.) to an item x that they are presented with at time t with probability π_t, i.e., $y \sim \texttt{Bernoulli}(\pi_t(x))$.

Our main assumption is that the likelihood of a user reacting to a particular item slightly increases if the score of the item is positive and slightly decreases if it is negative. Recall our example in Section 1, where users have a slight inclination to reinforce their opinion, i.e., increase their preference towards articles that they seem to correlate well with, and decrease it otherwise[3]. This deviation in user behavior is formally modeled by a *drift function*. Specifically, a drift function is a function $r : \mathbb{R} \longrightarrow [-1, 1]$ that relates the user's shift in the probability of liking an item, to its score s^t_x. The higher (resp. lower) the drift of a particular item, the more the user will positively (resp. negatively) reinforce their opinion for this item. In this work we will use $r(z) = K \cdot z \cdot e^{-z^2}$, where $K \approx 2.33$ is just a normalizing constant to ensure that $r(z) \in [-1, 1]$. The shape of this function along with its effect on the user preferences can be seen in Figure 2. It captures core properties of the user dynamics as explained in the caption. Similar functions have been used to model opinion dynamics for example regarding agreement with political parties [35].

Finally, we define user sensitivity to formalize the intuition that in order to reach an equilibrium, the behavior of the user cannot be decoupled from the recommender system. Essentially, sensitivity constitutes a bound on how much the true preference function of a user can deviate from the recommender system's prediction. The magnitude of this deviation depends on the drift function. The name "sensitivity" refers to the fact that a sensitive user will significantly reinforce their opinion after being presented with an item, i.e., they are not firm in their preferences and are prone to influence by the system/environment; the more sensitive they are (larger γ as defined below), the higher the influence.

Definition 1 (γ-sensitive user). *A user is γ-sensitive with respect to a drift function r, for $\gamma \in (0, 1)$, if for all time steps t and items x the difference between the true user preferences and the perceived preferences by the recommender system is bounded as follows:*

$$\pi_t(x) - \sigma(s^t_x) = \gamma \cdot \alpha_{t,x} \cdot r(s^t_x)$$

where:

$$\alpha_{t,x} = \begin{cases} 1 - \sigma(s^t_x) & , \textit{if } s^t_x \geq 0 \\ \sigma(s^t_x) & , \textit{if } s^t_x < 0 \end{cases}$$

is a scaling factor to ensure that $\pi_t(x) \in [0, 1]$ for all items x.

In the following, we will slightly abuse the notation and refer to the γ function associated with a γ-sensitive user as: $\gamma(s_x) = \gamma \cdot \alpha_{t,x} \cdot r(s_x)$ for all items x.

3 EVOLUTION OF USER PREFERENCES

In the previous section, we defined asymptotically stable fixed points and argued that they constitute the gold standard in the long-term behavior of a dynamically evolving recommender system, ruling out echo chamber formation. Hence, the natural question is:

Are there asymptotically stable fixed points
in the user-system interaction?

In this section, we present the main theoretical results of our paper. We answer the above question negatively and formally prove that the matrix factorization recommender system we defined will not reach a fixed point, hinting at the existence of echo chambers. We start by proving that under a mild condition on the items to be recommended there is a unique fixed point of the system dynamics. We defer the proof to Appendix A. All the results in this section hold for any choice of learning rates $\{\eta_t\}_{t=1}^{\infty}$.

Lemma 1 (Existence of Fixed Points.). *Let $X \in \mathbb{R}^{d \times m}$ be the matrix of the items available for recommendation and $XX^\top = \sum_x xx^\top$ the respective covariance matrix. If $rank(XX^\top) = d$, i.e., the covariance matrix has full rank, then $u^* = 0$ is the unique FP of (2).*

In the following lemma, we answer the second part of the question regarding the stability of that fixed point. Specifically, we prove that the norm of the user vector u_t always increases, ruling out the possibility for $u^* = 0$ to be a stable equilibrium between the user and the recommender system (Corollary 1).

Lemma 2. *If $rank(XX^\top) = d$ and $u_0 \neq 0$, then $\|u_{t+1}\| > \|u_t\|$ for all t. That is, the norm of the user preferences vector u_t strictly increases in every iteration.*

Proof. Following similar calculations to those in the proof of Lemma 1 (shown analytically in Appendix A) we get

$$\begin{aligned} u_{t+1} &= u_t - \eta_t \cdot \mathbb{E}_{x \sim f_t}\left[\nabla_u \ell(u_t, (x, y))\right] \\ &= u_t + \eta_t \cdot \sum_x x \cdot \mathbb{P}[x] \cdot \gamma(u_t^\top x) \\ &= u_t + \eta_t \cdot \sum_x x \cdot \mathbb{P}[x] \cdot \gamma \cdot \alpha_{t,x} \cdot K \cdot (u^\top x) \cdot e^{-(u_t^\top x)^2} \end{aligned}$$

The last equality follows by the definition of the γ function with $r(z) = K \cdot z \cdot e^{-z^2}$. To simplify notation, let $c_{t,x} = \eta_t \cdot \gamma \cdot K \cdot \mathbb{P}[x] \cdot \alpha_{t,x} \cdot e^{-(u_t^\top x)^2}$. Notice that $c_{t,x} > 0$ for all t, x. Hence, we have:

$$\begin{aligned} u_{t+1} &= u_t + \big(\sum_x c_{t,x} \cdot xx^\top\big) u_t \\ &= \big(I_d + \sum_x c_{t,x} \cdot xx^\top\big) u_t \end{aligned}$$

Consider the $m \times m$ diagonal matrix with values $c_{t,x}$ in the diagonal: $C_t = diag([c_{t,x_i}]_{i=1}^m)$. Then, we can rewrite $\sum_x c_{t,x} \cdot xx^\top = XC_tX^\top$. Also, notice that all the terms $c_{t,x} > 0$ for all t, x, hence C_t is positive definite. Now if $rank(XX^\top) = d$ we have that for any $v \in \mathbb{R}^d$ s.t. $v \neq 0$:

$$v^\top XC_tX^\top v = (X^\top v)^\top C_t (X^\top v) > 0,$$

since $X^\top v = 0$ has a unique solution $v = 0$. That is because $rank(XX^\top) = d \Rightarrow rank(X^\top) = d$. This implies that $XC_tX^\top$ is positive definite and thus all of its eigenvalues are strictly positive.

Let $A_t := I_d + XC_tX^\top$, which is symmetric and hence diagonalizable as $A_t = P_t \Lambda_t P_t^\top$, where P_t: orthogonal matrix. Moreover, all its eigenvalues, i.e. the diagonal elements of Λ_t, are strictly greater

[3] We measure this correlation using cosine similarity between u_t and x.

than 1 for every t, since they correspond to the eigenvalues of I_d (all 1) plus the eigenvalues of $XC_tX^\top$ (all strictly positive).

Now, consider the transformation

$$z_{t+1} = P_t^\top u_{t+1}, \text{for t} \geq 1, \text{with } z_0 = u_0$$

$$\Rightarrow z_{t+1} = P_t^\top u_{t+1} = P_t^\top A_t u_t = P_t^\top P_t \Lambda_t P_t^\top u_t = \Lambda_t z_t$$

Starting with $u_0 = z_0 \neq 0$ yields $z_t \neq 0$ for all t since Λ_t: diagonal matrix with non-zero diagonal elements. Moreover, since all the diagonal elements of Λ_t are strictly greater than 1 we have $\|z_{t+1}\| > \|z_t\|$. Now, since P_t, P_{t-1}: orthogonal matrices we have that $\|z_{t+1}\| = \|P_t^\top u_{t+1}\| = \|u_{t+1}\|$ and similarly $\|z_t\| = \|u_t\|$, completing the proof of the lemma.

□

Corollary 1 (Stability of FPs.). *Assume that $rank(XX^\top) = d$. Then, the unique fixed point $u^* = 0$ is not asymptotically stable.*

The two lemmas above essentially show that the unique fixed point is irrelevant as a solution concept of the user system interaction, hinting at the existence of echo chambers in matrix factorization recommender systems. Moreover, in the case where the features of the items in the system are transformed to be *decorrelated*, i.e. have identity covariance matrix[4] following the proof of Lemma 2, we can formally prove the amplification of the item scores. The proof is deferred to Appendix A.

Lemma 3 (Score amplification and Echo Chambers). *If $XX^\top = I_d$, then and for every item x s.t. $s_x^0 \neq 0$ it holds $|s_x^{t+1}| > |s_x^t|$ and $s_x^{t+1} \cdot s_x^t > 0$. That is, each score gets amplified with time.*

In the next section we provide simulations to show that even if the data is not decorrelated, i.e. $XX^\top \neq I_d$, the average absolute value of the score increases with time, essentially creating two well-separated groups of items in the system, the ones the user likes and the ones that they do not.

4 SIMULATIONS

In the previous section we proved that there is no asymptotically stable fixed point in the user-system interaction, giving evidence towards the existence of echo chambers and filter bubbles even in simple matrix factorization recommender systems. In this section, we conduct simulations on real and synthetic datasets to further investigate these phenomena by looking at the evolution of the user preferences u_t and the shape of the recommender system f_t, under different model sizes d, system parameters β, user sensitivities γ, and item distributions.

Metrics. We use the following metrics to evaluate the behavior of the recommender system and the evolution of the user preferences:

1. *Norm of u_t.* We expect that the ℓ_2 norm of the user vector will be driven away from 0, leading to a shift in the user preferences and a divergence of the item scores, as indicated by Lemma 2.
2. *Average probability per item type.* We partition the items into "likable" and "not likable" based on whether their assigned score at time t, s_x^t is positive or negative. We measure the average p_x^t for likable and not likable items.
3. *Probability mass on items correlating well with the initial preference vector u_0.* We consider the top 5% of items having the highest cosine similarity with u_0. These are the items the system initially believes the user will like. We report the total probability mass assigned to them by f_t for each t.

Experimental Setup. We fix the dimension of the system to be $d = 15$ but we found consistent results across a range of model sizes (relevant plots can be found in Appendix B.1). We run each simulation for 200 iterations for the synthetic datasets and for 750 for the real ones, where the user vector u_t is updated as in equation (2) with learning rate $\eta_t = 1$. In Appendix B.2 we include similar experiments using Stochastic Gradient Descent (SGD) updates, which essentially corresponds to the case where the user receives k recommendations at each time step. We consider a range of different β values $\beta \in \{0.5, 1, 1.5, 2\}$ that lead to different behavior of the recommender system, as well as different user sensitivities $\gamma \in \{0.2, 0.4, 0.6, 0.8\}$. Below, we describe the datasets that we use.

Synthetic datasets. We synthetically generate 100,000 items. The features of each item are generated from a fixed distribution f_{item}. We consider the following distributions:

- *Uniform in the hypercube.* Each item feature is generated independently and uniformly in $[-1, 1]$: $x \sim \texttt{Uniform}([-1, 1]^\text{d})$.
- *Mixture of two uniforms.* In many real platforms there is a dichotomy in the items present in the system, for example, regular content vs. content that violates some of the platform's policies. Also, some of the item features often correspond to classifier scores, with higher values indicating that the content is problematic (e.g. such classifiers can predict the probability of violent content, nudity, etc.). We model this effect using a mixture distribution. Specifically, with probability $1 - \alpha$, $x \sim \texttt{Uniform}([-1, \text{b}]^\text{d})$ (regular items) while with probability α, $x \sim \texttt{Uniform}([\text{b}, 1]^\text{d})$ (problematic items with high feature values). Here, α is the mixture coefficient and b is a threshold that we set to 0.8 for the simulations. We set $\alpha = 1\%$ which corresponds to the estimated prevalence of problematic content in real platforms, see e.g. [14].

For the synthetic datasets we initialize the user preference vector $u_0 \sim \texttt{Uniform}([-\epsilon, \epsilon]^\text{d})$ for $\epsilon = 0.3$. Note that for small $\|u_0\|$ we have $f_0 \approx f_{\text{item}}$, i.e., the recommender systems select items almost uniformly at random to present to the user.

Real datasets. We also consider the more realistic scenario of obtaining the initial user and the item features from real datasets. We use the *MovieLens 10M* [16] and the *Yahoo* [42] datasets. Movielens contains movie ratings for 69,878 users and 10,677 different movies, while Yahoo song ratings for 5,400 surveyed users and 1,000 songs. We selected 50 users with the highest number of ratings in each case, corresponding to 10,208 movies for Movielens and 982 songs for Yahoo. Since we use binary labels in our model, we mark each movie/song that was ranked with at least 3.5/5 as 1, indicating that the user liked it and the rest of them as 0. We used a Python library [26] that employs kernel matrix factorization [34] to obtain

[4] This is a common data preprocessing procedure, often applied for efficient optimization and can be easily done by applying the transformation $\tilde{x} = S^{-1}(x - \mu)$ to each item x, where $\mu = \frac{1}{m}\sum_x x$ and S is any matrix for which it holds $SS^\top = XX^\top$. Such a matrix exists if $rank(XX^\top) = d$.

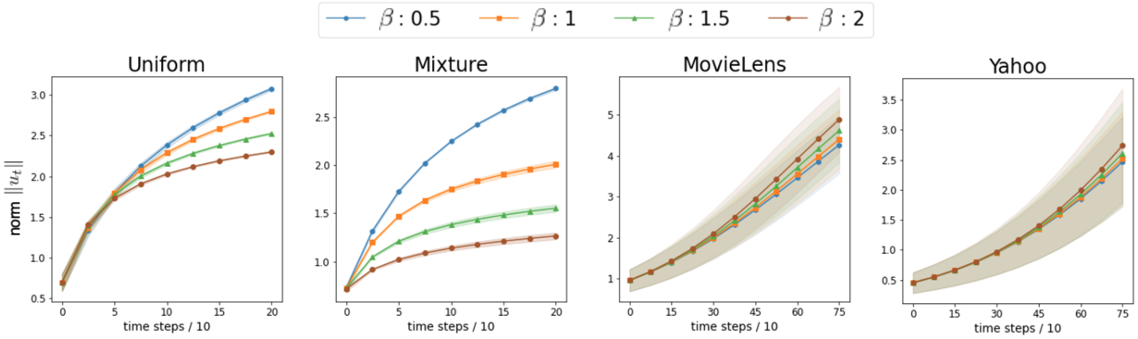

Figure 3: *Evolution of* $\|u_t\|$ *with* t. In all cases the user preferences diverge as predicted by Lemma 2.

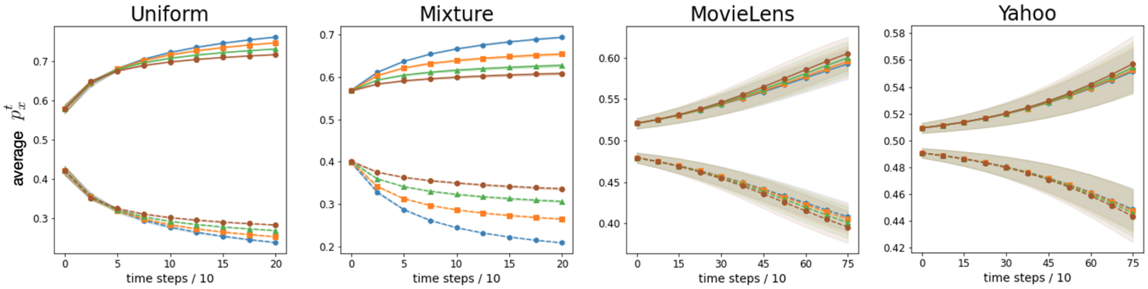

Figure 4: *Evolution of the average probability of a likable vs a non-likable item with* t. Solid lines indicate the likable items, whose probability of receiving a positive reaction from the user, p_x^t, is above 0.5, while the dashed lines indicate the non-likable items. The model size is $d = 15$ and $\gamma = 0.5$. The probability of clicking on an item moves towards 1 if its score is positive and towards 0 otherwise.

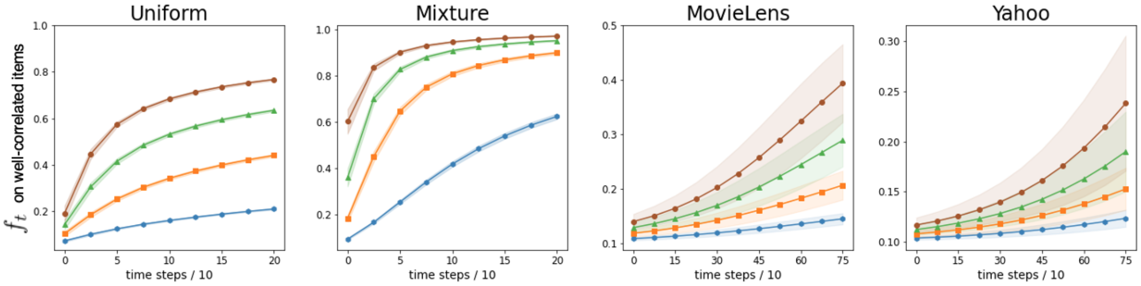

Figure 5: *Total probability mass assigned by* f_t *to the items with the highest cosine similarity with* u_0. The model size is $d = 15$ and $\gamma = 0.5$. There is a clear tendency to recommend items that the system believes that the user already likes and this tendency becomes more pronounced for higher values of β, making f_t more steep and approximating a Top-1 recommender system.

the user and item features as $R = U \cdot X$, where R is the binarized ratings matrix, U is the matrix of initial features for each user and X is the matrix of the item features.

4.1 Results

We evaluate the proposed metrics on the real and synthetic datasets. For the synthetic datasets we repeat each simulation 10 times with different vectors u_0, while for the real ones we iterate over each of the 50 different subsampled users. We report the means and the standard deviations of the run in Figs 3- 5, 7. Additional plots for different model sizes and SGD updates are presented in Appendix B.

Echo chambers and filter bubbles. First, we observe that the norm of u_t clearly diverges with time for different models and distributions as predicted by Lemma 2, indicating that a fixed point is never reached. As a result, the scores for each item are pushed towards the extremes as can be seen in Figure 4. Hence, every item tends to be *deterministically liked or not liked by the user*, suggesting an echo chamber. Secondly, in Figure 5 we demonstrate the filter

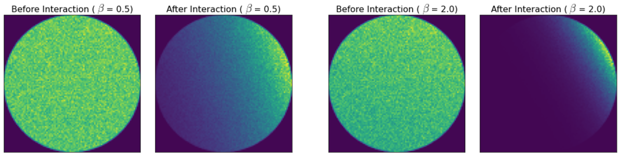

Figure 6: *The effect of different β on the recommender system f_t*. Consider the uniform distribution on the unit circle and $u_0 > 0$ coordinate-wise. The left two pictures show the distribution f_t in the beginning and after 100 steps of interaction with the user for $\beta = 0.5$. Brighter colors indicate higher probability mass. The two pictures on the right correspond to $\beta = 2$. It is evident that for larger β the transformation of f_t is much stronger with all of its mass concentrated around items with the highest possible scores that correlate well with u_0. In other words, the system tries to recommend items that have the highest probability of been liked. For smaller β the behavior is more explorative, and items with smaller scores have non-trivial probability of been recommended.

bubble effect that can occur in a matrix factorization recommender system. Specifically, we see that a small subset of items that have high cosine similarity with u_0, i.e., the system initially believes that the user will like them, are the ones that will end up being recommended with overwhelmingly high probability, narrowing down the exposure to any other type of content.

The effect of model sensitivity β. In Figures 3, 4 and 5 we plot our metrics for different values of β and fixed $\gamma = 0.5$. For the synthetic distributions, we see that higher values of β lead to slower divergence of the objectionable user preferences u_t while the opposite is true for the real ones. However, preference amplification, and the convergence of f_t to the items that correlate well with u_0 are faster. This is explained by the fact that for higher values of β, the stochastic recommender system of (1) turns to a `Top-1` recommender system, and is therefore more likely to present the users with items that they already liked. This leads to a stronger reinforcement of their preferences and increases the filter bubble effect. Visually, this difference in the recommender system for different values of β can be seen in Figure 6 (see also Fig. 14 in Appendix B) where we plot f_t, for two-dimensional items, for $t = 0$ and $t = 100$. We use two different values of β: $\beta = 0.5$ and $\beta = 2$ and we initialize with $u_0 > 0$ (coordinate-wise positive). It is evident that higher β leads to significantly higher concentration in the areas corresponding to items with high initial scores, i.e., items with large positive values in both coordinates (since $u_0 > 0$).

Effect of user-sensitivity γ. We consider $\beta = 1$ and varying values for γ. In Figure 7 we include the relevant plots for the uniform distribution and defer the results for the rest of the datasets to Figure 13 in Appendix B. Users with higher γ, i.e., the ones that are more prone to reinforce their previous opinion after been presented with an item, are the ones for whom the norm of their preferences vector diverges the fastest and they also experience the larger echo chamber and filter bubble effects. This implies that while there is a vulnerability in the recommender system, how much these effects will manifest is ultimately up to the user, matching findings in [3].

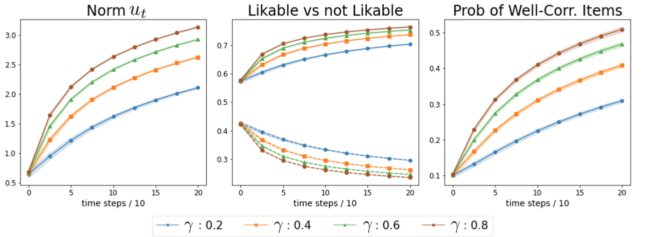

Figure 7: *Effect of user sensitivity γ*. On the left we plot $||u_t||$, in the middle the average probability for likable (solid) and non-likable items (dashed) and on the right the probability mass on the items that correlate well with u_0. The model size is $d = 15$ and $\beta = 1$. More sensitive users experience larger movements in f_t.

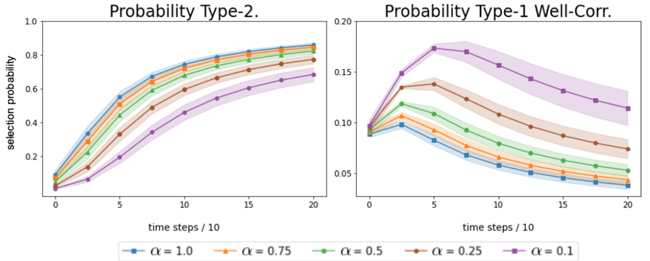

Figure 8: On the left we see the total probability that is assigned to the Type-2, low-prevalence items. The different lines correspond to different prevalence of such items in the system ranging from 1% to 0.1%. On the right we plot the probability on the Type-1 items that have high cosine similarity, i.e., they correlate well, with the initial preference vector u_0. These items act as a "bridge" for the probability to concentrate to the low-prevalence items.

4.2 Mitigation strategies

Consider the synthetic mixture distribution that we defined in the beginning of the section to model two different types of content (e.g. regular vs problematic). We will show experimentally, that even if the prevalence of one of the two types is very low, the recommender system is able to discover items of this type efficiently and present them to a user that shows interest for them. While this is generally a positive outcome, it can become an issue if the system contains problematic content. In this section, we will discuss two strategies, global and personalized demotion, to alleviate preference amplification, and in Section 4.3, we show the effect of such mitigation on the user's experience for borderline problematic content in a real large-scale recommender system.

Experimental Setup. Our basic mixture distribution consists of 100,000 items. They can be of *Type-1* sampled as $x \sim \text{Uniform}([-1, 0.8]^d)$ with probability $\alpha = 0.99$, or *Type-2* sampled as $x \sim \text{Uniform}([0.8, 1]^d)$ with probability $\alpha = 0.01$. We initialize $u_0 > 0$ (coordinate-wise positive), to focus on users that have high correlation with the low-prevalence Type-2 content in the system, hence they are interested in these items the most. We repeat each simulation 10 times.

Global Action. In this case, we actively remove items to reduce the prevalence of the Type-2 content in the system. We use two metrics to evaluate how the recommender system adapts to that:

(1) *Probability mass on Type-2 items.*
(2) *Probability mass on Type-1 items that have high cosine similarity with the initial preference vector u_0.* These are Type-1 items that the system initially believes that the user likes.

In Figure 8 we adjust the count of Type-2 items in the system so that their prevalence, would be α% for $\alpha \in \{1, 0.75, 0.5, 0.25, 0.1\}$. It can be seen that the probability mass of having a low-prevalence item presented to the user keeps increasing as the user interacts with the system more, *no matter how low the actual prevalence of that content is.* This indicates that matrix factorization recommender systems are able to surface items if they are relevant to the user. Additionally, we focus on Type-1 items that appear attractive to the user in the beginning and observe that they act as a "bridge" for the probability mass of f_t to transfer to the low-prevalence Type-2 items. Intuitively, in a real platform these could correspond to borderline problematic items that introduce the user to more problematic content.

Personalized Action. A recommender system applying personalized demotion avoids showing the user excessive amounts of a particular type of content without explicitly removing it from the system. In other words, it specifically targets the personalized recommendation system f_t of a user, and demotes the selection probability of certain item types. A way to achieve this is to use more reluctant updates of the preference vector u_t upon receiving user feedback. For example, a decaying learning rate schedule can be employed in (2) to slow down the preference adjustment and discourage "overfitting" to the users interests, hence preventing the creation of a filter bubble, or even a different learning rate for each particular feature i whose magnitude is inversely proportional to $(u_t)_i$. In Figure 9 we consider four different learning rate schedules for the system update (2). In each of these cases the η_t is adjusted every $t_{adj} = 20$ iterations. We use $t//t_{adj}$ to denote integer division:

(1) constant learning rate: $\eta_t = 1$,
(2) decaying learning rate: $1/(1 + t//t_{adj})$
(3) decaying learning rate: $1/\left(1 + \sqrt{t//t_{adj}}\right)$
(4) feature specific learning rate: In this case the learning rate corresponds to the matrix $\eta_t = diag\left[\left(\frac{1}{(u_t)_i}\right)_{i=1}^{d}\right]$, yielding different update for each coordinate. In particular the values for large coordinates are modified at a slower rate.

It can be seen that more reluctant updates in the preferences vector u_t mitigate the echo chamber and filter bubble effects, since the system is not actively changing the content that presents to the user, hence keeping it more diverse despite feedback loops.

4.3 Mitigation in Facebook recommendions

We now present an experiment that we conducted to study mitigation in a real-world large-scale video recommender system in Facebook. While this recommender system is complex and beyond the scope of this work, we describe a few relevant details here. Users can engage with videos in several ways including liking, commenting and flagging them as being "problematic". Videos have several features. The most relevant to our experiment is the probability that the video contains borderline nudity. We note that this content is mild and does not violate any enforcement standards; however, it may be considered problematic especially after repeated exposure.

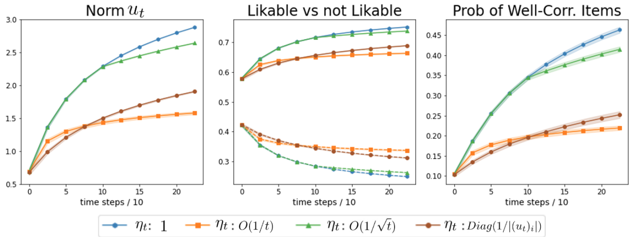

Figure 9: Evolution of the user preferences vector u_t and the recommender system for different learning rates. More reluctant updates in the preferences vector u_t mitigate the echo chamber and filter bubble effects.

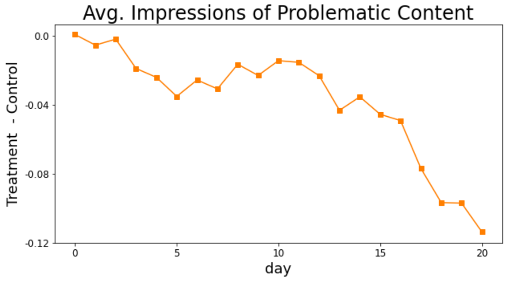

Figure 10: The difference in the average impressions of borderline problematic content viewed by treatment and control groups. As time progresses, users that received the treatment see progressively less problematic content.

We first identify users that are repeatedly exposed to such content. Specifically, using a classifier, we compute, for each user, the percentage of problematic videos (classifier output above a threshold) that they were exposed to. If it averages over 25% for a week, we reduce their exposure to this content for future sessions, hence attempting to reduce the echo chamber effect proactively and increase the diversity of the content recommended to them.

In the previous section, we illustrated personalized mitigation by altering the learning rate of the recommender system. However, changing the parameters of the recommender system is infeasible for a system in production. Instead, we adopt a simple personalized mitigation by showing lesser borderline content. This is achieved by lowering the score of borderline problematic content (an instantiation of $f_t(x)$) by multiplying it with a factor of 0.1 for users in the experiment. The factor of 0.1 was chosen after careful consideration of the item score distribution.

We ran this experiment for three weeks, with 37K users in each of the treatment and control groups, where the users in treatment received the mitigation. Figure 10 shows the difference in the average impressions of borderline problematic content viewed by users in control and those in treatment. At the start of the experiment, the two groups viewed about the same amount of problematic content, however, over time, the treatment group saw less, with the difference being statistically significant. Moreover, we observed that the overall interaction with content (likes, comments, etc.) by users in the treatment group increased by up to 2%. In conclusion, this experiment shows that echo chambers related to specific item features can create a negative user experience. Explicitly mitigating such effects can have an overall positive impact.

5 DISCUSSION AND FUTURE WORK

In this paper we examined the ability of stochastic matrix factorization recommender systems to reinforce the user preferences creating echo chambers and filter bubbles of content, from a theoretical point of view. This work serves as a step towards understanding and addressing such vulnerabilities with several future directions to explore further. A key question is whether we can prove quantitative rates for the divergence of $\|u_t\|$, hence understanding the exact dependence of echo chamber formation on the parameters of the system. Secondly, a natural question is what is the largest class of true preference functions for the user $\pi_t(\cdot)$ for which there is no asymptotically stable fixed point? Finally, an interesting effect that we need to take into account when applying mitigation strategies, is whether there are trade-offs in user engagement with the system. In Section 4.3, we provide preliminary evidence that this is not the case in real systems but further investigation is required.

REFERENCES

[1] Guy Aridor, Duarte Goncalves, and Shan Sikdar. 2020. Deconstructing the Filter Bubble: User Decision-Making And Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20).*

[2] Çigdem Aslay, Antonis Matakos, Esther Galbrun, and Aristides Gionis. 2018. Maximizing the Diversity of Exposure in a Social Network. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018.*

[3] E. Bakshy, S. Messing, and L. A. Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook.. In *Science.* 348:1130–1132.

[4] S. Banisch and E. Olbrich. 2019. Opinion polarization by learning from social feedback. *The Journal of Mathematical Sociology* 43, 2 (2019), 76–103.

[5] Andrea Barraza-Urbina and Dorota Glowacka. 2020. Introduction to Bandits in Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20).* Association for Computing Machinery, New York, NY, USA, 748–750.

[6] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018.*

[7] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Gançarski. 2012. A Contextual Bandit Algorithm for Mobile Context-Aware Recommender System. *Neural Information Processing Systems.*

[8] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *CoRR* abs/2010.03240 (2020). arXiv:2010.03240

[9] Uthsav Chitra and Christopher Musco. 2020. Analyzing the Impact of Filter Bubbles on Social Network Polarization. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20).*

[10] Matteo Cinelli, Gianmarco De Francisci Morales, Alessandro Galeazzi, Walter Quattrociocchi, and Michele Starnini. 2020. Echo Chambers on Social Media: A comparative analysis. *CoRR* abs/2004.09603 (2020). arXiv:2004.09603

[11] Mark A. Davenport, Yaniv Plan, Ewout van den Berg, and Mary Wootters. 2012. 1-Bit Matrix Completion. *CoRR* abs/1209.3672 (2012). arXiv:1209.3672

[12] Sarah Dean, Mihaela Curmei, and Benjamin Recht. 2020. Designing Recommender Systems with Reachability in Mind. In *Participatory Approaches to Machine Learning workshop at ICML.*

[13] Sarah Dean, Sarah Rich, and Benjamin Recht. 2020. Recommendations and user agency: the reachability of collaboratively-filtered information. In *FAT* '20: ACM Conference on Fairness, Accountability, and Transparency, Barcelona, Spain.*

[14] Facebook. 2020. Community Standards Enforcement Report. (2020). https://transparency.facebook.com/community-standards-enforcement

[15] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political Discourse on Social Media: Echo Chambers, Gatekeepers, and the Price of Bipartisanship. In *Proceedings of the 2018 World Wide Web Conference, WWW 2018, Lyon, France.*

[16] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. 5, 4 (2015).

[17] Elad Hazan. 2019. Introduction to Online Convex Optimization. *CoRR* abs/1909.05207 (2019). arXiv:1909.05207

[18] Homa Hosseinmardi, Amir Ghasemian, Aaron Clauset, David Rothschild, Marine Mobius, and Duncan Watts. 2020. Evaluating the scale, growth, and origins of right-wing echo chambers on YouTube. https://arxiv.org/abs/2011.12843

[19] Olivier Jeunen. 2019. Revisiting offline evaluation for implicit-feedback recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems.* 596–600.

[20] Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. 2019. Degenerate Feedback Loops in Recommender Systems. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA.*

[21] Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.*, Article 2 (Dec. 2016).

[22] Tyll Krueger, Janusz Szwabiński, and Tomasz Weron. 2017. Conformity, Anticonformity and Polarization of Opinions: Insights from a Mathematical Model of Opinion Dynamics. *Entropy* 19, 7 (Jul 2017), 371.

[23] Matevz Kunaver and Tomaz Pozrl. 2017. Diversity in recommender systems - A survey. *Knowl. Based Syst.* 123 (2017), 154–162.

[24] Mark Ledwich and Anna Zaitsev. 2020. Algorithmic extremism: Examining YouTube's rabbit hole of radicalization. *First Monday* (02 2020).

[25] Y. Lei and W. Li. 2019. When Collaborative Filtering Meets Reinforcement Learning. *ArXiv* abs/1902.00715 (2019).

[26] Kernel Matrix Factorization Library. 2020. https://pypi.org/project/matrix-factorization/

[27] Antonis Matakos and Aristides Gionis. 2018. Tell me Something My Friends do not Know: Diversity Maximization in Social Networks. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018.* IEEE Computer Society, 327–336.

[28] Celestine Mendler-Dünner, Juan C. Perdomo, Tijana Zrnic, and Moritz Hardt. 2020. Stochastic Optimization for Performative Prediction. In *Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, virtual.*

[29] Judith Möller, Damian Trilling, Natali Helberger, and Bram van Es. 2018. Do not blame it on the algorithm: an empirical assessment of multiple recommender systems and their impact on content diversity. *Information, Communication & Society* 21, 7 (2018), 959–977.

[30] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. 2014. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14).*

[31] Jack Nicas. 2018. How Youtube drives people to the internet's darkest corners. *Wall Street Journal, February 2018* (2018). https://www.wsj.com/articles/how-youtube-drives-viewers-to-the-\internets-darkest-corners-1518020478

[32] Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. [n.d.]. Performative Prediction. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event.*

[33] Rocha L.E.C. Perra, N. 2019. Modelling opinion dynamics in the age of algorithmic personalisation. *Sci Rep* (2019). https://doi.org/10.1038/s41598-019-43830-2

[34] Steffen Rendle and Lars Schmidt-Thieme. 2008. Online-Updating Regularized Kernel Matrix Factorization Models for Large-Scale Recommender Systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems* (Lausanne, Switzerland) *(RecSys '08).*

[35] David Sabin-Miller and Daniel M. Abrams. 2020. When pull turns to shove: A continuous-time model for opinion dynamics. *Phys. Rev. Research* 2 (Oct 2020), 043001. Issue 4. https://doi.org/10.1103/PhysRevResearch.2.043001

[36] J. Ben Schafer, Dan Frankowski, Jonathan L. Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. In *The Adaptive Web, Methods and Strategies of Web Personalization (Lecture Notes in Computer Science).* Springer.

[37] J. Ben Schafer, Joseph A. Konstan, and John Riedl. 2001. E-Commerce Recommendation Applications. *Data Min. Knowl. Discov.* 5, 1/2 (2001), 115–153.

[38] Joan E. Solsman. 2018. Youtube's AI is the puppet master over most of what you watch. *CNET* (2018). www.cnet.com/news/youtube-ces-2018-neal-mohan/

[39] Zeynep Tufekci. 2018. Youtube, the great radicalizer. *The New York Times, March* (2018). www.nytimes.com/2018/03/10/opinion/sunday/youtube-politics-radical

[40] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable Recourse in Linear Classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19).* Association for Computing Machinery, New York, NY, USA, 10–19. https://doi.org/10.1145/3287560.3287566

[41] Zeng Wei, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2017. Reinforcement Learning to Rank with Markov Decision Process.

[42] Yahoo! 2006. R3 – Music ratings for User Selected and Randomly Selected songs. https://webscope.sandbox.yahoo.com/catalog.php?datatype=r

A OMITTED PROOFS

Proof of Lemma 1. In order for u_t to be at an exact fixed point we need

$$u_{t+1} = u_t \Rightarrow \mathbb{E}_{x,y}\left[\nabla \ell\big(u_t, (x, y)\big)\right] = 0$$

In the following, we write $\ell_u(x, y)$ instead of $\ell\big(u_t, (x, y)\big)$ to simplify notation. Using the definition of the negative log-likelihood function ℓ, we have that

$$\begin{aligned}\mathbb{E}_{x,y}\left[\nabla \ell_u(x, y)\right] &= \sum_{x,y} \mathbb{P}[x]\mathbb{P}[y|x]\nabla \ell_u(x, y)\\ &= \sum_{x} \mathbb{P}[x]\big(\mathbb{P}[y = 1|x]\nabla \ell_u(x, 1)\\ +\mathbb{P}[y = 0|x]\nabla \ell_u(x, 0)\big)&\\ &= \sum_{x} x \cdot \mathbb{P}[x] \cdot \big(\sigma(u_t^\top x) - \pi(u_t, x)\big)\\ &= -\sum_{x} x \cdot \mathbb{P}[x] \cdot \gamma(u_t^\top x)\end{aligned}$$

where the last line follows by the definition of a γ-sensitive user. Based on the definition of the function γ we can expand the equation as follows:

$$\sum_{x} x \cdot \mathbb{P}[x] \cdot \gamma \cdot \alpha_{u_t,x} \cdot (u_t^\top x) \cdot e^{-(u_t^\top x)^2} = 0$$

Letting $c_{t,x} = \gamma \cdot \mathbb{P}[x] \cdot \alpha_{u_t,x} \cdot e^{-(u_t^\top x)^2}$ we have:

$$\Big(\sum_{x} c_{u,x} \cdot xx^\top\Big)u_t = 0$$

The above system has unique solution $u_t = 0$ iff $\sum_x c_{t,x} \cdot xx^\top$ has full rank. Consider the $m \times m$ diagonal matrix with values $c_{t,x}$ in the diagonal: $C_t = diag([c_{t,x_i}]_{i=1}^m)$. We can rewrite $\sum_x c_{t,x} \cdot xx^\top = XC_tX^\top$. Also, notice that all the terms $c_{t,x} > 0$ for all t, x, hence C_t is positive definite.

Now if $rank(XX^\top) = d$ we have that for any $v \in \mathbb{R}^d$ s.t. $v \neq 0$:

$$v^\top XC_tX^\top v = (X^\top v)^\top C_t(X^\top v) > 0,$$

since $X^\top v = 0$ has a unique solution $v = 0$ because $rank(X^\top) = d$ as well. This implies that A_t is positive definite and hence, also full rank: $rank(XC_tX^\top) = d$, completing the proof of the lemma. □

Proof of Lemma 3. Let $s_t = (s_x^t)_x$ be the vector of the scores of all the items at time t. Using the notation and the derivations of the proof of Lemma 1 we have:

$$\begin{aligned}s_{t+1} &= X^\top u_{t+1}\\ &= X^\top \Big(I_d + \sum_{x} c_{t,x}xx^\top\Big)u_t\\ &= X^\top \big(I_d + XC_tX^\top\big)u_t\\ &= X^\top X\big(I_m + C_t\big)X^\top u_t = \big(I_m + C_t\big)s_t\end{aligned}$$

where we used the fact that $XX^\top = I_d$. Now, since C_t is diagonal with positive diagonal elements each score gets amplified, meaning that it keeps the same sign and its absolute value increases, and the lemma follows. □

B ADDITIONAL PLOTS

B.1 Model Size d

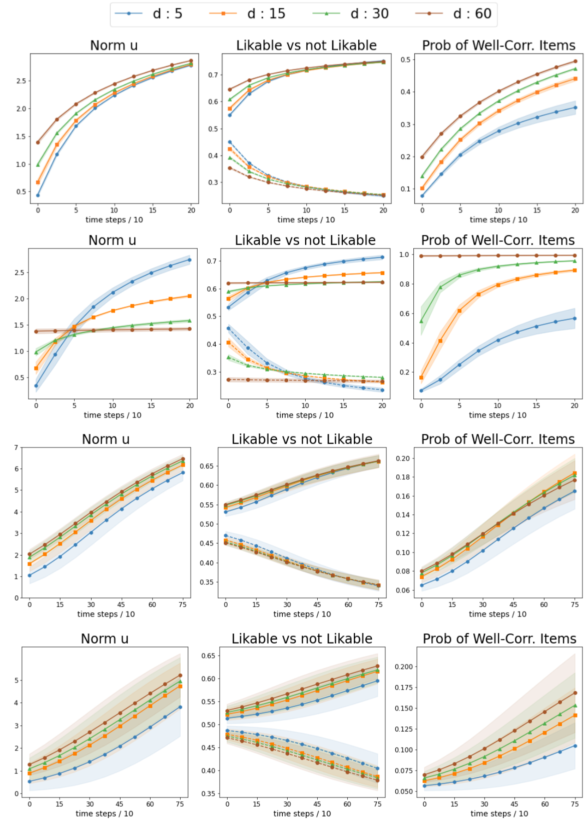

Figure 11: Effect of different model sizes d for the uniform (first row), and mixture distribution (second row), the Movielens dataset (third row) and the Yahoo dataset (fourth row).

We experiment with different model sizes $d \in \{5, 15, 30, 60\}$ for fixed $\beta = 1$ and $\gamma = 0.5$. We repeat the simulation for each model size 10 times and we report the mean and the standard deviations of our metrics for all four datasets in Figure 11. It is evident from the plots, the effect of model size heavily depends on the distribution of the items in the system. If the distribution is not heavily skewed towards some particular direction, then higher model sizes lead to faster divergence in the user preferences but the effect is not particularly strong. However, in the case of the mixture distribution where by construction we initialize $u_0 > 0$ the initial scores of the Type-2 items are very high. As a result all the mass is already concentrated there leading to essentially no movement in u_t.

B.2 Stochastic Gradient Descent

By analyzing the gradient descent update in (2) we can study the expected trajectory of the vector u_t and user-system interaction. However, this update is difficult to implement and in practice the common approach is to substitute it with SGD updates. In that case,

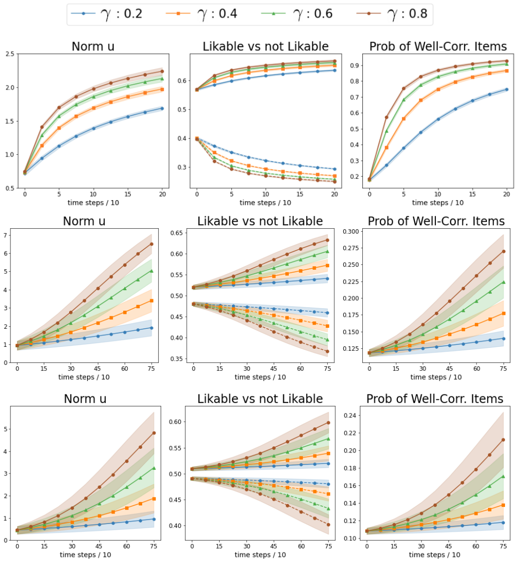

Figure 13: Effect of gamma for the mixture distribution (first row), the Movielens dataset (second row) and the Yahoo dataset (third row). With the γ parameter controlling the amount of preference reinforcement that a user exhibits, it is expected that in all cases higher values for γ lead to faster divergence of the user preferences as well as higher separation between likable and not likable items.

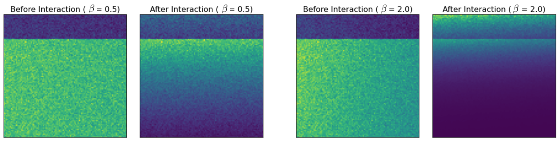

Figure 14: *Evolution of the recommender system f_t for a user that is interested in low-prevalence content.* Consider the two-dimensional item distribution where the first feature is uniform in [-1, 1], while the second feature is uniform in [-1, 0.8] with probability 90% and uniform in (0.8, 1] with the remaining 10%. Hence, all item for which $x_2 > 0.8$ are significantly more rare in the system. We consider a user with u_0 whose second coordinate is positive. The two pictures on the left show the distribution f_t in the beginning and after 100 steps of interaction with the user for $\beta = 0.5$. Brighter colors indicate higher probability mass. The two pictures on the right correspond to $\beta = 2$. In the case where $\beta = 0.5$ the initial surface is virtually identical to the actual distribution of items in the system, while in the case where $\beta = 2$ there is noticeable difference. At $t = 100$, in both cases the probability mass concentrates on the items for which x_2 is large, and the effect is much more pronounced with higher β. In particular, for $\beta = 2$, already after 100 iterations the mass starts concentrating in the low-prevalence items, while for $\beta = 0.5$ the mass concentrates on the barrier between the high prevalence and the low prevalence items, without being able to efficiently surface the low prevalence items yet.

the vector u_t evolves as:

$$u_{t+1} = u_t - \eta_t \cdot \frac{1}{k} \sum_{i=1}^{k} \nabla \ell(u_t, (x_i^t, y_i^t))$$

where k can be thought as the number of items that are presented to the user at each iteration. The user responds with feedback $y_i^t \in \{0, 1\}$ to each of them. We set $k = 30$ for the experiments. For the synthetic distributions we run the experiments with 20 different initializations and for each initialization we run it for 15 times. For the real datasets we just run SGD updates for the 50 selected users as before. We report the mean and the standard deviations of the run for diffrent values of the parameter β in Figure 12. As before we run the simulation for 200 different iterations for the synthetic and for 750 for the real datasets and we use $\eta = 0.1$.

As in the case of the the expected loss (Figures 3 - 5) the norm of u_t diverges in all cases and the user's preference get polarized, with the noise of SGD having limited effect. On the other hand, we see that the filter-bubble phenomenon, i.e. how fast does the recommender system converge into showing the user only items that they like is less evident here and the respective standard deviations are very large, indicating that the trajectory of f_t is very sensitive to the initial conditions u_0.

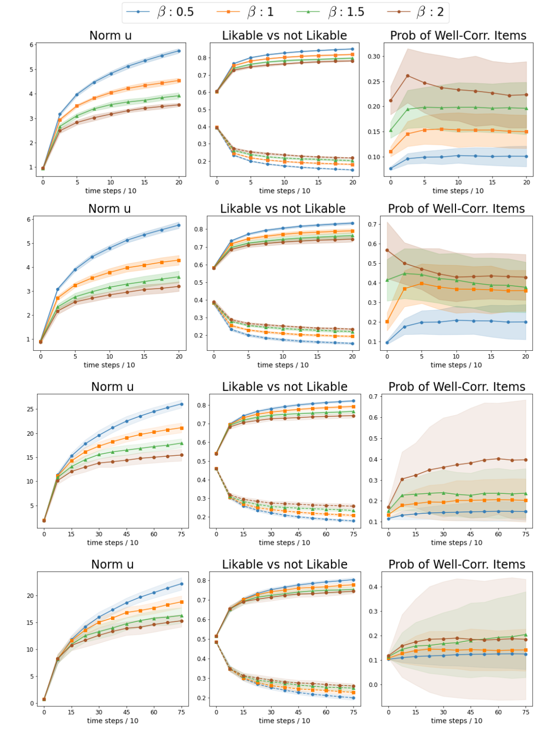

Figure 12: SGD with different β parameters for the uniform distribution (first row), the mixture distribution (second row), Movielens (third row) and Yahoo (fourth row).