

Recommendations and User Agency: The Reachability of Collaboratively-Filtered Information

Sarah Dean
dean_sarah@berkeley.edu
UC Berkeley

Sarah Rich
sarah@canopy.cr
Canopy Crest

Benjamin Recht
brecht@berkeley.edu
UC Berkeley

ABSTRACT

Recommender systems often rely on models which are trained to maximize accuracy in predicting user preferences. When the systems are deployed, these models determine the availability of content and information to different users. The gap between these objectives gives rise to a potential for unintended consequences, contributing to phenomena such as filter bubbles and polarization. In this work, we consider directly the information availability problem through the lens of user recourse. Using ideas of reachability, we propose a computationally efficient audit for top- N linear recommender models. Furthermore, we describe the relationship between model complexity and the effort necessary for users to exert control over their recommendations. We use this insight to provide a novel perspective on the user cold-start problem. Finally, we demonstrate these concepts with an empirical investigation of a state-of-the-art model trained on a widely used movie ratings dataset.

ACM Reference Format:

Sarah Dean, Sarah Rich, and Benjamin Recht. 2020. Recommendations and User Agency: The Reachability of Collaboratively-Filtered Information. In *Conference on Fairness, Accountability, and Transparency (FAT* '20)*, January 27–30, 2020, Barcelona, Spain. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3351095.3372866>

1 INTRODUCTION

Recommendation systems influence the way information is presented to individuals for a wide variety of domains including music, videos, dating, shopping, and advertising. On one hand, the near-ubiquitous practice of filtering content by predicted preferences makes the digital information overload possible for individuals to navigate. By exploiting the patterns in ratings or consumption across users, preference predictions are useful in surfacing relevant and interesting content. On the other hand, this personalized curation is a potential mechanism for social segmentation and polarization. The exploited patterns across users may in fact encode undesirable biases which become self-reinforcing when used in feedback to make recommendations.

Recent empirical work shows that personalization on the Internet has a limited effect on political polarization [15], and in

fact it can increase the diversity of content consumed by individuals [31]. However, these observations follow by comparison to non-personalized baselines of cable news or well known publishers. In a digital world where all content is algorithmically sorted by default, how do we articulate the tradeoffs involved? In the past year, YouTube has come under fire for promoting disturbing children's content and working as an engine of radicalization [7, 32, 46]. This comes a push on algorithm development towards reaching 1 billion hours of watchtime per day; over 70% of views now come from the recommended videos [42].

The Youtube controversy is an illustrative example of potential pitfalls when putting large scale machine learning-based systems in feedback with people, and highlights the importance of creating analytical tools to anticipate and prevent undesirable behavior. Such tools should seek to quantify the degree to which a recommender system will meet the information needs of its users or of society as a whole, where these “information needs” must be carefully defined to include notions like relevance, coverage, and diversity. An important approach involves the empirical evaluation of these metrics by simulating recommendations made by models once they are trained [13]. In this work we develop a complementary approach which differs in two major ways: First, we directly analyze the predictive model, making it possible to understand underlying mechanisms. Second, our evaluation considers a range of possible user behaviors rather than a static snapshot.

Drawing conclusions about the likely effects of recommendations involves treating humans as a component within the system, and the validity of these conclusions hinges on modeling human behavior. We propose an alternative evaluation that favors the agency of individuals over the limited perspective offered by behavioral predictions. Our main focus is on questions of *possibility*: to what extent can someone be pigeonholed by their viewing history? What videos may they never see, even after a drastic change in viewing behavior? And how might a recommender system encode biases in a way that effectively limits the available library of content?

This perspective brings user agency into the center, prioritizing the the ability for models to be as adaptable as they are accurate, able to accommodate arbitrary changes in the interests of individuals. Studies find positive effects of allowing users to exert greater control in recommendation systems [21, 51]. While there are many system-level or post-hoc approaches to incorporating user feedback, we focus directly on the machine learning model that powers recommendations.

Contributions. In this paper, we propose a definition of *user recourse* and *item availability* for recommender systems. This perspective extends the notion of recourse proposed by Ustun et al. [47] to multiclass classification settings and specializes to concerns most relevant for information retrieval systems. We focus our analysis

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.
FAT* '20, January 27–30, 2020, Barcelona, Spain

© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6936-7/20/02...\$15.00
<https://doi.org/10.1145/3351095.3372866>

on top- N recommendations made using linear predictions, a broad class including matrix factorization models. In Section 3 we show how properties of latent user and item representations interact to limit or ensure recourse and availability. This yields a novel perspective on user cold-start problems, where a user with no rating history is introduced to a system. In Section 4, we propose a computationally efficient model audit. Finally, in Section 5, we demonstrate how the proposed analysis can be used as a tool to interpret how learned models will interact with users when deployed.

1.1 Related Work

Recommendation models that incorporate user feedback for online updates have been considered from several different angles. The computational perspective focuses on ensuring that model updates are efficient and fast [22]. The statistical perspective articulates the sampling bias induced by recommendation [5], while practical perspectives identify ways to discard user interactions that are not informative for model updates [8]. Another body of work focuses on the learning problem, seeking to improve the predictive accuracy of models by exploiting the sequential nature of information. This includes strategies like Thompson sampling [24], upper confidence bound approximations for contextual bandits [6, 28], and reinforcement learning [27, 50].

This body of work, and indeed much work on recommender systems, focuses on the accuracy of the model. This encodes an implicit assumption that the primary information needs of users or society are described by predictive performance. Alternative measures proposed in the literature include concepts related to diversity or novelty of recommendations [9]. Directly incorporating these objectives into a recommender system might include further predictive models of users, e.g. to determine whether they are “challenge averse” or “diversity seeking” [45]. Further alternative criteria arise from concerns of fairness and bias, and recent work has sought to empirically quantify parity metrics on recommendations [13, 14]. In this work, we focus more directly on agency rather than predictive models or observations.

Most similar to our work is a handful of papers focusing on decision systems through the lens of the agency of individuals. We are most directly inspired by the work of Ustun et al. [47] on actionable recourse for binary decisions, where users seek to change negative classification through modifications to their features. This work has connections to concepts in explainability and transparency via the idea of *counterfactual explanations* [40, 48], which provide statements of the form: if a user had features X , then they would have been assigned alternate outcome Y . Work in strategic manipulation studies nearly the same problem with the goal of creating a decision system that is robust to malicious changes in features [19, 30].

Applying these ideas to recommender systems is complex because while they can be viewed as classifiers or decision systems, there are as many outcomes as pieces of content. Computing precise action sets for recourse for every user-item pair is unrealistic; we don’t expect a user to even become aware of the majority of items. Instead, we consider the “reachability” of items by users, drawing philosophically from the fields of formal verification and dynamical system analysis [3, 34].

2 PROBLEM SETTING

A recommender system considers a population of users and a collection of items. We denote a “rating” by user u of item i as $r_{ui} \in \mathcal{R} \subseteq \mathbb{R}$. This value can be either explicit (e.g. star-ratings for movies) or implicit (e.g. number of listens). Let n denote the number of users in the system and m denote the number of items. Though these are both generally quite large, the number of *observed ratings* is much smaller. Let Ω_u denote the set of items whose ratings by user u have been observed. We collect these observed ratings into a sparse vector $\mathbf{r}_u \in \mathcal{R}^m$ whose values are defined at Ω_u and 0 elsewhere. Then a system makes recommendations with a policy $\pi(\mathbf{r}_u)$ which returns a subset of items.¹ We will denote the size of the returned subset as N , which is a parameter of the system.

We are now ready to define the reachability sub-problem for a recommender system. We say that a user u can reach item i if there is some allowable modification to their rating history \mathbf{r}_u that causes item i to be recommended. The reachability problem for user u and item i is defined as

$$\begin{aligned} & \underset{\mathbf{r} \in \mathcal{M}(\mathbf{r}_u)}{\text{minimize}} && \text{cost}(\mathbf{r}; \mathbf{r}_u) \\ & \text{subject to} && i \in \pi(\mathbf{r}) \end{aligned} \quad (1)$$

where the modification set $\mathcal{M}(\mathbf{r}_u) \subseteq \mathcal{R}$ describes how users are allowed to modify their rating history and $\text{cost}(\mathbf{r}; \mathbf{r}_u)$ describes how “difficult” or “unlikely” it is for a user to make this change. This notion of difficulty might relate discretely to the total number of changes, or to the amount that these changes deviate from the existing preferences of the user. By defining the cost with respect to user behavior, the reachability problem encodes both the *possibilities* of recommendations through its feasibility, as well as the relative *likelihood* of different outcomes as modeled by the cost.

The ways that users can change their rating histories, described by the modification set $\mathcal{M}(\mathbf{r}_u)$, depends on the design of user input to the system. We consider a single round of user reactions to N recommendations and focus on two models of user behavior: changes to existing ratings, which we will refer to as “history edits,” and reaction to a batch of recommended items, which we will refer to as “reactions.” In the first case, $\mathcal{M}(\mathbf{r}_u)$ consists of all possible ratings on the support Ω_u . In the second, it consists of all new ratings on the support $\pi(\mathbf{r}_u)$ combined with the existing rating history.

The reachability problem (1) defines a quantity for each user and item in the system. To use this problem as a metric for evaluating recommender systems, we consider both user- and item-centric perspectives. For users, this is a notion of *recourse*.

Definition 2.1. The *amount* of recourse available to a user u is defined as the percentage of unseen items that are reachable, i.e. for which (1) is feasible. The *difficulty* of recourse is defined by the average value of the recourse problem over all reachable items i .

On the other hand, the item-centric perspective centers around notions of *availability and representation*.

¹ While we focus on the case of deterministic policies, the analyses can be extended to randomized policies which sample from a subset of items based on their ratings. It is only necessary to define reachability with respect to probabilities of seeing an item, and then to carry through terms related to the sampling distribution.

Definition 2.2. The *availability* of items in a recommender system is defined as the percentage of items that are reachable by some user.

These definitions are important from the perspective of guaranteeing fair representation of content within recommender systems. This is significant for users – for example, to what extent have their previously expressed preferences limited the content that is currently reachable? It is equally important to content creators, for whom the ability to build an audience depends on the availability of their content in the recommender system overall.

In what follows, we turn our attention to specific classes of preference models, for which the reachability problem is analytically tractable. However, we note that estimation via sampling can provide lower bounds on availability and recourse even for black-box models, and further explore this observation in Section 4.1. We keep the main focus on analysis rather than sampling because it allows us to crisply distinguish between unlikely and impossible.

2.1 Linear Preference Models

While many different approaches to recommender systems exist [39], ranging from classical neighborhood models [17] to more recent deep neural networks [11], we focus our attention on *linear preference models*. These are models which predict user rating as the dot product between user and item vectors plus bias terms:

$$\widehat{r}_{ui} = \mathbf{q}_i^\top \mathbf{p}_u + b_i + c_u + \mu.$$

We will refer to \mathbf{q}_i and \mathbf{p}_u as item and user representations. This broad class includes both item- and user-based neighborhood models, sparse approaches like SLIM [33], and matrix factorization, which differ only in how the user and item representations are determined from the rating data. For ease of exposition, we drop the bias terms for the body of the paper and focus on *matrix factorization*, deferring our general results and explanation to Appendix A. Matrix factorization is a classical approach to recommendation which become prominent during the Netflix Prize competition [49]. It can be specialized to different assumptions about data and user behavior, including constrained approaches like non-negative matrix factorization [16] or augmentations like inclusion of implicit information about preferences and additional features [22, 23, 25, 26, 35, 37]. Due to its power and simplicity, the matrix factorization approach is still widely used; indeed it has recently been shown to be capable of attaining state-of-the-art results [12, 38].

In this setting, the item and user representations are referred to as factors, lying in a latent space of specified dimension d which controls the complexity of the model. The factors can be collected into matrices $P \in \mathbb{R}^{n \times d}$ and $Q \in \mathbb{R}^{m \times d}$. Fitting the model most commonly entails solving the nonconvex minimization:

$$\underset{P, Q}{\text{minimize}} \sum_u \sum_{i \in \Omega_u} (r_{ui} - \mathbf{p}_u^\top \mathbf{q}_i)^2 + \Gamma(P, Q) \quad (2)$$

where Γ is a regularizer.

The predicted ratings of unseen items are used to make recommendations. Specifically, we consider top- N recommenders which return

$$\{i : \widehat{r}_{ui} > \widehat{r}_{uj} \text{ all but at most } N \text{ unseen items } j\}.$$

For linear preference models, the condition $\widehat{r}_{ui} > \widehat{r}_{uj}$ reduces to

$$\mathbf{q}_i^\top \mathbf{p}_u > \mathbf{q}_j^\top \mathbf{p}_u \iff (\mathbf{q}_i - \mathbf{q}_j)^\top \mathbf{p}_u > 0.$$

Thus, for fixed item vectors, a user's recommendations are determined by their representation along with a list of unseen items. In a slight abuse of notation, we will use this fact to write the recommender policy $\pi(\mathbf{p}; \Omega)$ instead of $\pi(\mathbf{r})$.

When users' ratings change, their representations change as well. While there are a variety of possible strategies for performing online updates, we focus on the least squares approach, where

$$\mathbf{p}_u = \arg \min_{\mathbf{p}} \|\mathbf{r}_{u, \Omega_u} - Q_{\Omega_u} \mathbf{p}\|_2^2 + \Gamma_u(\mathbf{p}).$$

This is similar to continuing an alternating least-squares (ALS) minimization of (2), a common strategy [52]. Because we analyze a single round of recommendations, we do not consider a simultaneous updates to the item representations in $\{\mathbf{q}_i\}$.

In what follows, we focus on the canonical case of ℓ_2 regularization on user and item factors, with $\Gamma_u(\mathbf{x}) = \Gamma_i(\mathbf{x}) = \lambda \|\mathbf{x}\|_2^2$. With this, the user factor calculation is given by:

$$\mathbf{p}_u = (Q_{\Omega_u}^\top Q_{\Omega_u} + \lambda I)^{-1} Q_{\Omega_u}^\top \mathbf{r}_u, \quad (3)$$

which is linear in the rating vector. In Appendix A, we show that several other linear preference models have similar user updates.

3 RECOURSE AND AVAILABILITY

We begin by reformulating the reachability problem to the case of recommendations made by matrix factorization models (the general reformulation for linear models is presented in Appendix A). We focus this initial exposition on the simplifying case that $N = 1$ and make direct connections between between model factors and the recourse and availability provided by the recommender system.

First, we consider what needs to be true for an item i to be recommended. For top-1, the constraint $i \in \pi(\mathbf{p}, \Omega)$ is equivalent to requiring that

$$(\mathbf{q}_i - \mathbf{q}_j)^\top \mathbf{p} > 0 \quad \forall j \notin \Omega \iff G_i \mathbf{p} > 0.$$

where we define G_i to be a $m - |\Omega| \times d$ matrix with rows given by $(\mathbf{q}_i - \mathbf{q}_j)$ for $j \notin \Omega$. This is a linear constraint on the user factor \mathbf{p} , and the set of user factors which satisfy this constraint make up an open convex polytopic cone. We refer to this set as the *item-region* for item i , since any user whose latent representation falls within this region will be recommended item i . The top-1 regions partition the latent space, as illustrated by Figure 1 for a toy example with latent dimension $d = 2$.

If item factors define regions within the latent space, user factors are points that may move between regions. The constraints on user actions are described by the modification set $\mathcal{M}(\mathbf{r}_u)$. We will distinguish between *mutable* and *immutable* ratings of items within a rating vector \mathbf{r}_u . Let Ω_0 denote the set of items with immutable ratings and let $\mathbf{r}_0 \in \mathbb{R}^{|\Omega_0|}$ denote the corresponding ratings. Then let Ω_m denote the set of items with mutable ratings. In what follows, we will write the full set of observed ratings $\Omega = \Omega_0 \cup \Omega_m$. Then the modification set is all rating vectors $\mathbf{r} \in \mathcal{R}$ with:

- (1) fixed immutable ratings, $\mathbf{r}_{\Omega_0} = \mathbf{r}_0$
- (2) mutable ratings $\mathbf{r}_{\Omega_m} = \mathbf{a}$ for some value $\mathbf{a} \in \mathbb{R}^{|\Omega_m|}$,
- (3) unseen items with no rating, $\mathbf{r}_{\Omega^c} = 0$

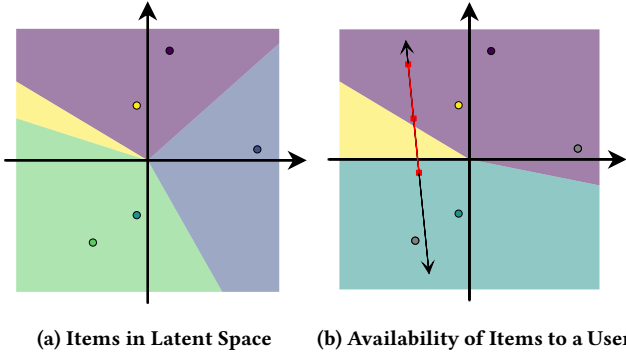


Figure 1: An example of item factors (indicted by colored points) in $d = 2$. In (a), the top-1 regions are indicated by shaded colors. The teal item is unavailable, and though the yellow item is reachable, it is not aligned-reachable. In (b), the availability of items for a user who has seen the blue and the green items (now in grey) with the blue item's rating fixed. The black line indicates how the user's representation can change depending on their rating of the green item. The red region indicates the constraining effect of requiring bounded and integer-valued ratings, which affect the reachability of the yellow region.

The variable \mathbf{a} is the decision variable in the reachability problem.

Then a user's latent factor can change as

$$\mathbf{p} = (Q_\Omega^\top Q_\Omega + \lambda I)^{-1} (Q_\Omega^\top \mathbf{r}_0 + Q_{\Omega_m}^\top \mathbf{a}) = \mathbf{v}_0 + B\mathbf{a}$$

where we define

$$W = (Q_\Omega^\top Q_\Omega + \lambda I)^{-1}, \quad B = W Q_{\Omega_m}^\top, \quad \mathbf{v}_0 = W Q_\Omega^\top \mathbf{r}_0.$$

It is thus clear that this latent factor lies in an affine subspace. This space is anchored at \mathbf{v}_0 by the immutable ratings, while the mutable ratings determine the directions of possible movement. This idea is illustrated in Figure 1b, which further demonstrates the limitations due to bounded or discrete ratings, as encoded in the rating set \mathcal{R} .

We are now able to specialize reachability problem (1) for matrix factorization models:

$$\begin{aligned} & \underset{\mathbf{a} \in \mathcal{R}^{|\Omega_m|}}{\text{minimize}} && \text{cost}([\mathbf{r}_0; \mathbf{a}]; \mathbf{r}_u) \\ & \text{subject to} && G_i(\mathbf{v}_0 + B\mathbf{a}) > 0 \end{aligned} \quad (4)$$

If the cost is a convex function and \mathcal{R} is a convex set, this is a convex optimization problem which can be solved efficiently. If \mathcal{R} is a discrete set or if the cost function incorporates nonconvex phenomena like sparsity, then this problem can be formulated as a mixed-integer program (MIP). Despite bad worst-case complexity, MIP can generally be solved quickly with modern software [18, 47].

3.1 Item Availability

Beyond defining the reachability problem, we seek to derive properties of recommender systems based on their underlying preference models. We begin by considering the feasibility of (4) with respect to its linear inequality constraints. For now, we focus on the item-regions, ignoring the effects of user history Ω , anchor point \mathbf{v}_0 , and control matrix B . In the following result, we consider the *convex*

hull of unseen item factors, which is the the smallest convex set that contains the item factors. This can be formally written as,

$$\text{conv}(\{\mathbf{q}_j\}_j) = \left\{ \sum_j \lambda_j \mathbf{q}_j : \sum_j \lambda_j \leq 1 \text{ and } \lambda \geq 0 \right\}.$$

Furthermore, we will consider *vertices* of the convex hull, which are item factors that are not contained in the convex hull of the other factors, i.e. $\mathbf{q}_i \notin \text{conv}(\{\mathbf{q}_j\}_{j \neq i})$.

RESULT 1. *In a top-1 recommender system, the available items are those whose factors are vertices on the convex hull of all item factors.*

As a result, the availability of items in a top-1 recommender system is determined by the way the item factors are distributed in space: it is simply the percentage of item factors that are vertices of their convex hull. The proof is provided in Appendix A, along with proofs of all results to follow.

We can further understand the effect of limited user movement in the case that ratings are real-valued, i.e. $\mathcal{R} = \mathbb{R}$. In this case, we consider both the control matrix B and the anchor point \mathbf{v}_0 . For a fixed i , this anchor point determines the set of items j necessary for comparison: $\mathbf{q}_j^\top \mathbf{v}_0 \geq \mathbf{q}_i^\top \mathbf{v}_0$, i.e. those that are more similar to the anchor point than item i is. We will refer to these items as the *anchor-similar items*. Furthermore, we consider multiplication of item factors by the transpose of the control matrix, the *multiplied factors* $B^\top \mathbf{q}_i$.

RESULT 2. *In a top-1 recommender system, a user can reach any item i whose multiplied factor is a vertex of the convex hull of all unseen anchor-similar multiplied item factors.*

Furthermore, if the factors of the items with mutable ratings are full rank, i.e. Q_{Ω_m} has rank equal to the latent dimension of the model d , then item availability implies user recourse.

The second statement in this result means that for a model with 100% item availability, having as many mutable ratings as latent dimensions is sufficient for ensuring that users have full recourse (so long as the associated item factors are linearly independent). This observation highlights that increased model complexity calls for more comprehensive user controls to maintain the same level of recourse.

Of course, this conclusion follows only from considering the *possibilities* of user action – to consider a notion of *likelihood* for various outcomes we need to consider the cost.

3.2 Bound on Difficulty of Recourse

We now propose a simple model for the cost of user actions, and use this to show a bound on the difficulty of recourse for users. The cost of user actions can be modeled as a penalty on *change from existing ratings*. For items whose ratings have not already been observed, we penalize instead the *change from predicted ratings*. For simplicity, we will represent this penalty as the norm of the difference.

For history edits, all mutable items have been observed, so we simply have

$$\text{cost}_{\text{hist}}(\mathbf{r}; \mathbf{r}_u) = \|\mathbf{r} - \mathbf{r}_u\|.$$

Additionally, all existing ratings are mutable so mutable set $\Omega_m = \Omega_u$ and immutable set $\Omega_0 = \emptyset$. For reactions, the ratings for the

new recommended items have not been observed, so

$$\text{cost}_{\text{react}}(\mathbf{r}; \mathbf{r}_u) = \|\mathbf{r}_{\pi(\mathbf{r}_u)} - \widehat{\mathbf{r}}_{\pi(\mathbf{r}_u)}\|.$$

Additionally, the rating history is immutable so $\Omega_0 = \Omega_u$ while the mutable ratings are the recommendations with $\Omega_m = \pi(\mathbf{r}_u)$.

We note briefly that our choice of handling the cost of unobserved ratings as “change from predicted ratings” assumes a level of model validity. While it does allow us to avoid any external behavioral modeling, it represents a simple case that perhaps over-emphasizes the role of the model. Exploring alternative cost functions is important for future work.

Under this model, we provide an upper bound on the difficulty of recourse. This result holds for the case that ratings are real-valued, i.e. $\mathcal{R} = \mathbb{R}$ and that the reachable items satisfy an *alignment* condition (defined in (8) of Appendix A).

RESULT 3. *Let \mathbf{p}_u indicate the user’s latent factor as in (3) before any actions are taken or the next set of recommendations are added to the user history. Then both in the case of full history edits and reactions,*

$$\text{difficulty of recourse for user } u \leq \|B^\dagger\| \cdot \frac{1}{|\Omega_r|} \sum_{i \in \Omega_r} \|\mathbf{q}_i - \mathbf{p}_u\|,$$

where $\Omega_r \subseteq \Omega^c$ is the set of reachable items.

This bound depends how far item factors are from the initial latent representation of the user. When latent representations are close together, recourse is easier or more likely—an intuitive relationship. This quantity will be large in situations where a user is in an isolated niche, far from most of the items in latent space. The bound also depends on the conditioning of the user control matrix B , which is related to the similarity between mutable items: the right hand side of the bound will be larger for sets of mutable items that are more similar to each other.

The proof of this result hinges on showing the existence of a specific feasible point to the optimization problem in (4). In the Section 4 we will further explore this idea of feasibility to develop lower bounds on availability and recourse when $N > 1$. First, we described how the presented results can be used to evaluate solutions to the user cold-start problem.

3.3 User Cold-Start

The amount and difficulty of recourse for a user yields a novel perspective on how to incorporate new users into a recommender system. The user cold-start problem is the challenge of selecting items to show a user who enters a system with no rating history from which to predict their preferences. This is a major issue with collaboratively filtered recommendations, and systems often rely on incorporating extraneous information [41]. These strategies focus on presenting items which are most likely to be rated highly or to be most informative about user preferences [4].

The idea of recourse offers an alternative point of view. Rather than evaluating a potential “onboarding set” only for its contribution to model accuracy, we can choose a set which additionally ensures some amount of recourse. Looking to Result 2, we can evaluate an onboarding set by the geometry of the *multiplied factors* in latent space. In the case of onboarding, $\mathbf{v}_0 = 0$ and $B = WQ_\Omega^\top$,

so the recourse evaluation involves considering the vertices of the convex hull of the columns of the matrix $\{Q_\Omega W Q_{\Omega^c}^\top\}$.

An additional perspective is offered by considering the difficulty of recourse. In this case, we focus on $\|B^\dagger\|$. If we consider an ℓ_2 norm, then it reduces to

$$\|B^\dagger\| = \max_i \frac{\sigma_i^2 + \lambda}{\sigma_i}$$

where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ are the nonzero singular values of Q_Ω . Minimizing this quantity is hard [10], though the hardness of selecting informative item sets is unsurprising as it has been discussed in related settings [4]. Due to their computational challenges, we primarily propose that these metrics be used to distinguish between candidate onboarding sets, based on the ways these sets provide user control. We leave to future work the task of generating candidate sets based on these recourse properties.

4 SUFFICIENT CONDITIONS FOR TOP-N

In the previous section, we developed a characterization of reachability for top-1 recommender systems. However, most real world applications involve serving several items at once. Furthermore, using $N > 1$ can approximate the availability of items to a user over time, as they see more items and increase the size of the set Ω which is excluded from the selection. In this section, we focus on sufficient conditions to develop a computationally efficient model audit that provides lower bounds on the *availability* of items in a model. We further provide approximations for computing a lower bound on the *recourse* available to users.

We can define an item-region for the top- N case, when $i \in \pi(\mathbf{p}; \Omega)$ for any user factors in the set

$$\mathcal{P}_i = \{\mathbf{p} : (\mathbf{q}_i - \mathbf{q}_j)^\top \mathbf{p} > 0 \text{ all but at most } N \text{ items } j \notin \Omega\}.$$

As in the previous section, this region is contained within the latent space, which is generally of relatively small dimension. However, its description depends on the number of items, which will generally be quite large. In the case of $N = 1$, this dependence is linear and therefore manageable. For $N > 1$, the item region is the union over polytopic cones for subsets describing “all but at most N items.” Therefore, the description of each item region requires $O(m^N)$ linear inequalities. For systems with tens of thousands of items, even considering $N = 5$ becomes prohibitively expensive.

To ease the notational burden of discussing the ranking logic around top- N selection in what follows, we define the operator $\max^{(N)}$, which selects the N th largest value from a set. This allows us to write, for example,

$$\widehat{r}_{ui} > \widehat{r}_{uj} \text{ all but at most } N \text{ items } j \iff \widehat{r}_{ui} > \max_{j \neq i}^{(N)} \widehat{r}_{uj}.$$

4.1 Sufficient Condition for Availability

To bypass the computational concerns, we focus on a sufficient condition for item availability. The full description of the region \mathcal{P}_i is not necessary to verify non-emptiness; rather, showing the existence of any point in the latent space $\mathbf{v} \in \mathbb{R}^d$ that satisfies $\mathbf{v} \in \mathcal{P}_i$ is sufficient. Using this insight, we propose a sampling approach to determining the availability of an item. For a fixed \mathbf{v} and any N , it is necessary only to compute and sort $Q_{\Omega^c} \mathbf{v}$, which is an operation of complexity $O(m^2 d \log(m))$.

While this sampling approach could make use of gridding, randomness, or empirical user factor distributions, we propose choosing the sample point $\mathbf{v} = \mathbf{q}_i$.

RESULT 4. *The item-region \mathcal{P}_i is nonempty if*

$$\delta_i = \|\mathbf{q}_i\|_2^2 - \max_{j \notin \Omega \cup \{i\}}^{(N)} \mathbf{q}_j^\top \mathbf{q}_i > 0. \quad (5)$$

When this condition holds, we say that item i is aligned-reachable. The percent of items that are aligned-reachable is a lower bound on the availability of items.

Note that this is a sufficient rather than a necessary condition; it is possible to have $\mathbf{q}_i \notin \mathcal{P}_i$ for a nonempty \mathcal{P}_i . Figure 1 illustrates an example where this is the case: the yellow latent factor lies within $\mathcal{P}_{\text{purple}}$ even though $\mathcal{P}_{\text{yellow}}$ is nonempty. As a result, aligned-reachability yields an underestimate of the availability of items in a system.

4.2 Model Audit

To use the aligned-reachable condition (5) as a generic model audit, we need to sidestep the specificity of the set of seen items Ω . We propose an audit based on this condition with $\Omega = \emptyset$ and an increased value for N , where increasing N compensates for discarding the effect of the items which have been seen. This audit is described in Algorithm 1. If we set $N' = N + N_h$ where N is the number of items recommended by the system, then item availability has the following interpretation: if an item is not top- N' available, then that item will never be recommended to a user who has seen fewer than N_h items.

Algorithm 1: Item-Based Model Audit

Result: Lower bound on item availability $1 - \frac{m_{\text{unavailable}}}{m}$

```

1 initialize  $m_{\text{unavailable}} = 0$ ,  $N' = N + N_h$ ;
2 for  $i \in \{1, \dots, m\}$  do
3   compute  $\delta_i = \|\mathbf{q}_i\|_2^2 - \max_{j \neq i}^{(N')} \mathbf{q}_j^\top \mathbf{q}_i$ ;
4   if  $\delta_i \leq 0$  then
5      $m_{\text{unavailable}}^+ = 1$ 
6   end
7 end
```

If we consider the set of all possible users to be users with a history of at most N_h , this model audit counts the number of aligned-unreachable items, returning lower bound on the overall availability of items. We can further use this model audit to propose constraints or penalties on the model during training. Ensuring aligned-reachability is equivalent to imposing linear constraints on the matrix $A = QQ^\top$,

$$A_{ii} \geq \max_{j \neq i}^{(N)} A_{ij}.$$

While this constraint is not convex, relaxed versions of it could be incorporated into the optimization problem (2) to ensure reachability during training. We point this out as a potential avenue for future work.

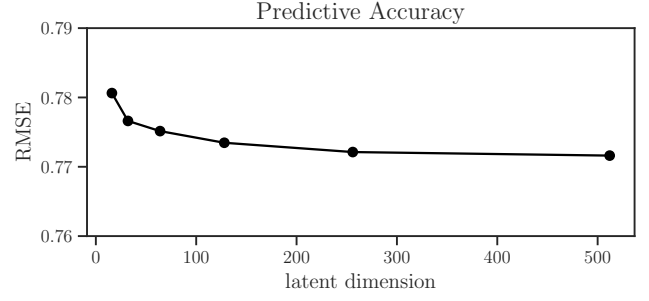


Figure 2: The test RMSE of the matrix factorization models on the MovieLens dataset.

4.3 Sufficient Condition for Recourse

User recourse inherits the computational problems described above for $N > 1$. We note that the region \mathcal{P}_i is not necessarily convex, though it is the union of convex regions. While the problem could be solved by first minimizing within each region and then choosing the minimum value over all regions, this would not be practical for large values of N . Therefore we continue with the sampling perspective to develop an efficient sufficient condition for verifying the feasibility of (4). We propose testing feasibility with the condition

$$\mathbf{v}_0 + B\mathbf{a}_i \in \mathcal{P}_i \text{ for } \mathbf{a}_i \in \arg \min_{\mathcal{R}^k} \|\mathbf{B}\mathbf{a} + \mathbf{v}_0 - \mathbf{q}_i\|_2^2. \quad (6)$$

By checking feasibility for each i , we verify a lower bound on the amount of recourse available to a user, considering their specific rating history and the allowable actions.

Note that if the control matrix B is full rank, then we can find a point \mathbf{a}_i such that $\mathbf{v}_0 + B\mathbf{a}_i = \mathbf{q}_i$, meaning that items who are aligned-reachable are also reachable by users. The rank of B is equal to the rank of Q_{Ω_m} , so as previously observed, item availability implies recourse for any user with control over at least d ratings whose corresponding item factors are linearly independent.

Even users with incomplete control have some level of recourse. For the following result, we define Π_B as the projection matrix onto the subspace spanned by B . Then let $\mathbf{q}_{B,i} = \Pi_B \mathbf{q}_i$ be the component of the target item factor that lies in the space spanned by the control matrix B , and $\mathbf{v}_\perp = \mathbf{v}_0 - \Pi_B \mathbf{v}_0$ be the component of the anchor point that cannot be affected by user control.

RESULT 5. *When $\mathcal{R} = \mathbb{R}$, a lower bound on the amount of recourse for a user u is given by the percent of unseen items that satisfy:*

$$\|\mathbf{q}_{B,i}\|_2^2 + \mathbf{q}_i^\top \mathbf{v}_\perp > \max_{j \neq i}^{(N)} (\mathbf{q}_j^\top \mathbf{q}_{B,i} + \mathbf{q}_j^\top \mathbf{v}_\perp).$$

Note that this statement mirrors the sufficient condition for items, with modifications relating both to the directions of user control and the anchor point. In short, user recourse follows from the ability to modify ratings for a set of diverse items, and immutable ratings ensure the reachability of some items, potentially at the expense of others.

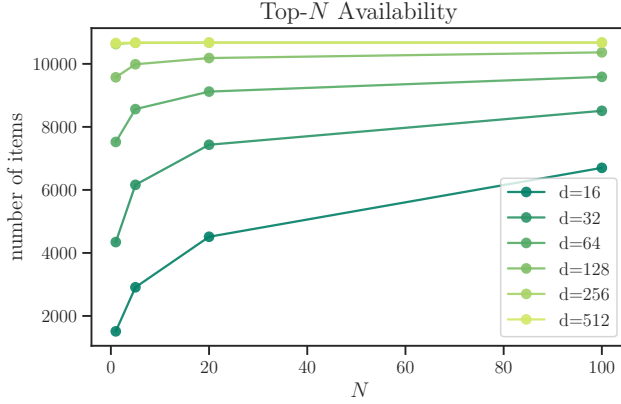


Figure 3: Only some of the 10,681 total movies are aligned-reachable, especially for models with smaller complexity and for smaller recommendation set sizes N .

5 EXPERIMENTAL DEMONSTRATION

In this section, we demonstrate how our proposed analyses can be used as a tool to audit and interpret characteristics of a matrix factorization model. We use the MovieLens 10M dataset, which comes from an online movie recommender service called MovieLens [20]. The dataset² contains approximately 10 million ratings applied to 10,681 movies by 71,567 users. The ratings fall between 0 and 5 in 0.5 increments.

We chose this dataset because it is a common benchmark for evaluating rating predictions. Using the method described by Rendle et al. [38] in their recent work on baselines for recommender systems, we train a regularized matrix factorization model. This model incorporates item, user, and overall bias terms. (Appendix A includes full description of adapting our proposed audits to this model).

We examine models of a variety of latent dimension ranging from $d = 16$ to $d = 512$. The models were trained using the libfm³ library [36]. We use the regression objective and optimize using SGD with regularization parameter $\lambda = 0.04$ and step size 0.003 for 128 epochs on 90% of the data, verifying accuracy on the remaining 10% with a random global test/train split. These methods match those presented by Rendle et al. [38] and reproduce their reported accuracies (Figure 2).

In Appendix B, we present a similar set of experiments on the LastFM dataset.

5.1 Item-Based Audit

We begin by performing the item-based audit as described in Section 4.2. Figure 3 displays the total number of aligned-reachable movies for various parameters N . It is immediately clear that all items are baseline-reachable in only the models with the largest latent dimension. Indeed, we can conclude that the model with $d = 16$ has only about 60% availability for users with a history of under 100 movies. On the other hand, the models with the highest

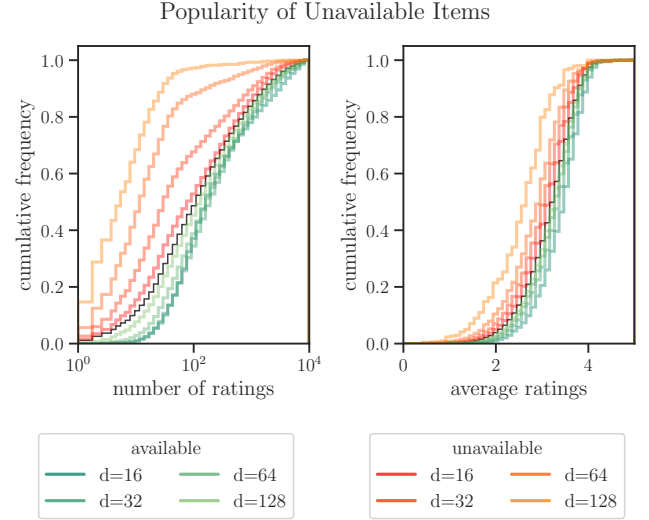


Figure 4: Unavailable items are systematically less popular than available items: they are rated less frequently and have lower average ratings in the training data. Each curve represents the cumulative density function (CDF) of the popularity measure within the available (green) and unavailable (red) items. The black line represents the CDF of the combined population. This trend is true for models of varying complexity.

complexity $d \in \{256, 512\}$ have about 99% availability for even the smallest values of N .

We further examine the characteristics of the items that are unavailable compared with those that are available. We examine two notions of popularity: total number of ratings and average rating. In Figure 4, we compare the distributions of the available and unavailable items (for $N = 5$) in the training set on these measures. The unavailable items have systematically lower popularity for various latent dimensions. This observation has implications for the outcome of putting these models in feedback with users. If unavailable items are never recommended, they will be less likely to be rated, which may exacerbate their unavailability if models are updated online. We highlight this phenomenon as a potential avenue for future work.

While the difference in popularity is true across all models, it is important to note that there is still overlap in the support of both distributions. For a given number of ratings or average rating, some items will be available while others will not, meaning that popularity alone does not determine reachability.

5.2 System Recourse for Users

Next, we examine the users in this dataset. We use the combined testing and training data to determine user ratings \mathbf{r}_u and histories Ω_u . For this section, we examine 100 randomly selected users and only the 1,000 most rated items. Sub-selecting items and especially choosing them based on popularity means that these experimental results are an overestimation of the amount of recourse available

² <http://grouplens.org/datasets/movielens/10m/>

³ <http://www.libfm.org/>

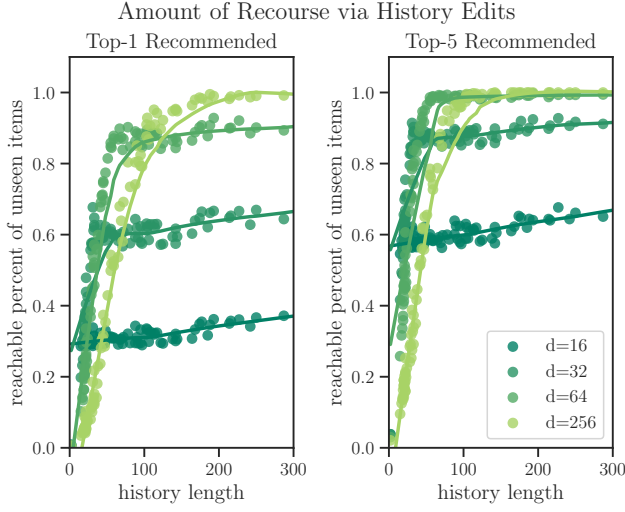


Figure 5: The proportion of unseen items reachable by users varies with their history length. A LOESS regressed curve illustrates the trend. Less complex models are better for shorter history lengths, while more complex models reach higher overall values.

to users. Additionally, we allow ratings on the continuous interval $\mathcal{R} = [0, 5]$ rather than enforcing integer constraints, meaning that our results represent the recourse available to users if they were able to precisely rate items on a continuous scale. Despite these two approximations, several interesting trends on the limits of recourse appear.

We begin with history edits, and compute the amount of recourse that the system provides to users using the sufficient condition in Result 5. Figure 5 shows the relationship with the length of user history for several different latent dimensions. First, note the shape of the curve: a fast increase and then leveling off for each dimension d . For short histories, we see the limiting effect of projection onto the control matrix Π_B . For longer histories, as the rank of Q_{Ω_u} approaches or exceeds d , the baseline item-reachability determines the effect. The transition between these two regimes differs depending on the latent dimension of the model. Smaller models reach their maximum quickly, while models of higher complexity provide a larger amount of recourse to users with long histories. This is an interesting distinction that connects to the idea of “power users.”

Next, we consider reactions, where user input comes only through reaction to a new set of items while the existing ratings are fixed. Figure 6 displays the amount of recourse for two different types of new items: first, the case that users are shown a completely random set of 5 unseen items and second, the case that they are shown the 5 items with the highest predicted ratings. The top panel displays the amount of recourse provided by each model and each type of recommendation. There are two important trends. First, smaller models offer larger amounts of recourse—this is because we are in the regime of few mutable ratings, analogous to the availability of items to users with short histories in the previous figure. Second, for each model size, the random recommendations provide

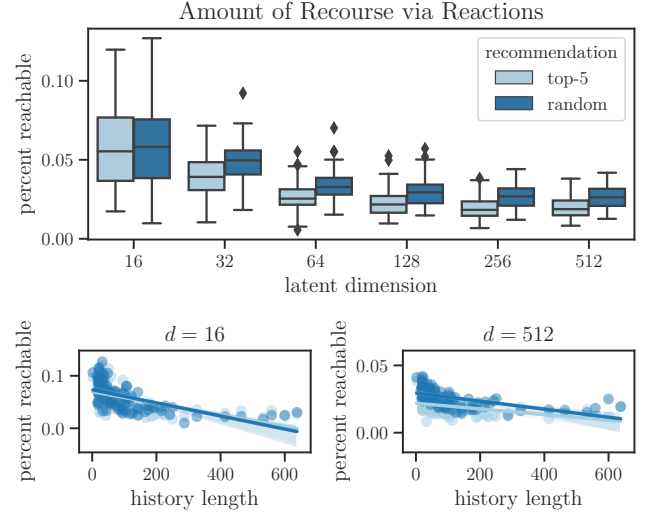


Figure 6: When actions are constrained to reaction to a set of items, lower complexity models provide higher reachability. A random set of items provides slightly more recourse to users than if the set is selected based on predicted user preferences. Furthermore, there is a slight trend that users with smaller history lengths have more available recourse.

more recourse than the top-5, and though the gap is not large it is consistent.

In the bottom panel of Figure 6, we further examine how the length of user history interacts with this model of user behavior. For both the smallest and the largest latent dimensions, there is a downwards trend between reachability and history length. This does not contradict the trend displayed in Figure 5: in the reactions setting, the rating history manifests as the anchor point v_0 rather than additional degrees of freedom in the control matrix B . It is interesting in light of recent works examining the usefulness of recency bias in recommender systems [29].

Finally, we investigate the difficulty of recourse over all users and a single item. In this case, we consider top-1 recommendations to reduce the computational burden of computing the exact set \mathcal{P}_i . We pose the cost as the size of the difference between the user input a and the predicted ratings in the ℓ_1 norm. Figure 7 shows the difficulty of recourse via reaction for the two types of new items: a completely random set of 20 unseen items and 20 items with the highest predicted ratings. We note two interesting trends. First, the difficulty of recourse does not increase with model size (even though the amount of recourse is lower). Second, difficulty is lower for the random set of items than for the top-20 items. Along with the trend in availability, this suggests a benefit of suggesting items to users based on metrics other than predicted rating. Future work should more carefully examine methods for constructing recommended sets that trade-off predicted ratings with measures like diversity under the lens of user recourse.

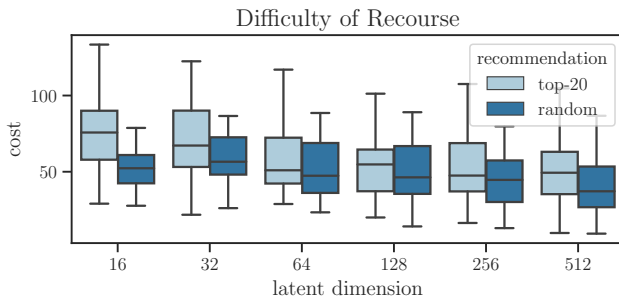


Figure 7: The difficulty of reaching a single item across 100 users for different sets of new times. The difficulty of recourse does not increase for the larger models, despite the previously observed decrease in availability. Furthermore, we note that random items have lower difficulty.

6 DISCUSSION

In this paper, we consider the effects of using predicted user preferences to recommend content, a practice prevalent throughout the Internet today. By defining a reachability problem for top- N recommenders, we provide a way to evaluate the impact of using these predictive models to mediate the discovery of content. In applying these insights to linear preference models, we see several interesting phenomena. The first is simple but worth stating: good predictive models, when used to moderate information, can unintentionally make portions of the content library inaccessible to users. This is illustrated in practice in our study of the MovieLens and LastFM datasets.

To some extent, the differences in the availability of items are related to their unpopularity within training data. Popularity bias is a well known phenomenon in which systems fail to personalize [43], and instead over-recommend the most popular pieces of content. Empirical work shows connections between popularity bias and undesirable demographic biases, including the under-recommendation of female authors [14]. YouTube was long known to have a popularity bias problem (known as the “Gangnam Style Problem”), until the recommendations began optimizing for predicted “watch time” over “number of views.” Their new model has been criticized for its radicalization and political polarization [32, 46]. The choice of prediction target can have a large effect on the types of content users can or are likely to discover, motivating the use of analytic tools like the ones proposed here to reason about these trade-offs before deployment.

While the reachability criteria proposed in this work form an important basis for reasoning about the availability of content within a recommender system, they do not guarantee less biased behavior on their own. Many of the audits consider the feasibility of the recourse problem rather than its cost; thus confirming possible outcomes rather than distinguishing probable ones. Furthermore, the existence of recourse does not fix problems of filter bubbles or toxic content. Rather, it illuminates limitations inherent in recommender systems for organizing and promoting content. There is an important distinction between technically providing recourse and the likelihood that people will actually avail themselves of it. If the

cost function is not commensurate with actual user behavior this analysis may lend an appearance of fairness without substance.

With these limitations in mind, we mention several ways to extend the ideas presented in this work. On the technical side, there are different models for rating predictions, especially those that incorporate implicit feedback or perform different online update strategies for users. Not all simple models are linear—for example, subset based recommendations offer greater scrutability and thus user agency by design [2]. Further more, top- N recommendation is not the only option. Post-processing approaches to the recommender policy π could work with existing models to modify their reachability properties. For example, Steck [44] proposed a method to ensure that the distribution of recommendations over genres remains the same despite model predictions.

One avenue for addressing more generic preference models and recommender policies is to extend the sampling perspective introduced in Section 4.1 to develop a general framework for black-box recommender evaluation. By sampling with respect to a user transition model, the evaluation could incorporate notions of dynamics and user agency similar to those presented in this work.

Further future work could push the scope of the problem setting to understand the interactions between users and models over time. Analyzing connections between training data and the resulting reachability properties of the model would give context to empirical work showing how biases can be reproduced in the way items are recommended [13, 14]. Similarly, directly considering multiple rounds of interactions between users and the recommendation systems would shed light on how these models evolve over time. This is a path towards understanding phenomena like filter bubbles and polarization.

More broadly, we emphasize the importance of auditing systems with learning-based components in ways that directly consider the models’ behavior when put into feedback with humans. In the field of formal verification, making guarantees about the behavior of complex dynamical systems over time has a long history. There are many existing tools [1], though they are generally specialized to the case of physical systems and suffer from the curse of dimensionality. We accentuate the abundance of opportunity for developing novel approximations and strategies for evaluating large scale machine learning systems.

ACKNOWLEDGMENTS

Thanks to everyone at Canopy for feedback and support. SD is supported by an NSF Graduate Research Fellowship under Grant No. DGE 175281. BR is generously supported in part by ONR awards N00014-17-1-2191, N00014-17-1-2401, and N00014-18-1-2833, the DARPA Assured Autonomy (FA8750-18-C-0101) and LaGrange (W911NF-16-1-0552) programs, and an Amazon AWS AI Research Award.

REFERENCES

- [1] Eugene Asarin, Olivier Bournez, Thao Dang, and Oded Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 20–31. Springer, 2000.
- [2] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. Transparent, scrutable and explainable user models for personalized recommendation. 2019.

- [3] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.
- [4] Sampoorna Biswas, Laks VS Lakshmanan, and Senjuti Basu Ray. Combating the cold start user problem in model based collaborative filtering. *arXiv preprint arXiv:1703.00397*, 2017.
- [5] Stephen Bonner and Flavian Vasile. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 104–112. ACM, 2018.
- [6] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In *International Conference on Neural Information Processing*, pages 324–331. Springer, 2012.
- [7] James Bridle. Something is wrong on the internet. *Medium*, November 2017. URL <https://medium.com/@jamesbridle/something-is-wrong-on-the-internet-c39c471271d2>.
- [8] Alexandra Burashnikova, Yury Maximov, and Massih-Reza Amini. Sequential learning over implicit feedback for robust large-scale recommender systems. *arXiv preprint arXiv:1902.08495*, 2019.
- [9] Pablo Castells, Saül Vargas, and Jun Wang. Novelty and diversity metrics for recommender systems: choice, discovery and relevance. 2011.
- [10] Ali Çivril and Malik Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49): 4801–4811, 2009.
- [11] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198. ACM, 2016.
- [12] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. *arXiv preprint arXiv:1907.06902*, 2019.
- [13] Michael D Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In *Conference on Fairness, Accountability and Transparency*, pages 172–186, 2018.
- [14] Michael D Ekstrand, Mucun Tian, Mohammed R Imran Kazi, Hoda Mehrpouyan, and Daniel Kluger. Exploring author gender in book rating and recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 242–250. ACM, 2018.
- [15] Seth Flaxman, Sharad Goel, and Justin M Rao. Filter bubbles, echo chambers, and online news consumption. *Public opinion quarterly*, 80(S1):298–320, 2016.
- [16] Nicolas Gillis. The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, 12(257), 2014.
- [17] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–71, 1992.
- [18] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2019. URL <http://www.gurobi.com>.
- [19] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122. ACM, 2016.
- [20] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19, 2016.
- [21] F Maxwell Harper, Funing Xu, Harmanpreet Kaur, Kyle Condiff, Shuo Chang, and Loren Terveen. Putting users in control of their recommendations. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 3–10. ACM, 2015.
- [22] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 549–558. ACM, 2016.
- [23] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, volume 8, pages 263–272. Citeseer, 2008.
- [24] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient thompson sampling for online matrix factorization recommendation. In *Advances in neural information processing systems*, pages 1297–1305, 2015.
- [25] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009):1–10, 2009.
- [26] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.
- [27] Yu Lei and Wenjie Li. When collaborative filtering meets reinforcement learning. *arXiv preprint arXiv:1902.00715*, 2019.
- [28] Jérémie Mary, Romaric Gaudel, and Philippe Preux. Bandits and recommender systems. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 325–336. Springer, 2015.
- [29] Paweł Matuszyk, João Vinagre, Myra Spiliopoulou, Alípio Mário Jorge, and João Gama. Forgetting methods for incremental matrix factorization in recommender systems. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 947–953. ACM, 2015.
- [30] Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. *arXiv preprint arXiv:1808.08460*, 2018.
- [31] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, pages 677–686. ACM, 2014.
- [32] Jack Nicas. How youtube drives people to the internet’s darkest corners. *The Wall Street Journal*, February 2018. URL <https://www.wsj.com/articles/how-youtube-drives-viewers-to-the-internets-darkest-corners-1518020478>.
- [33] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *2011 IEEE 11th International Conference on Data Mining*, pages 497–506. IEEE, 2011.
- [34] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.
- [35] Arkadiusz Paterk. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8, 2007.
- [36] Steffen Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012. ISSN 2157-6904.
- [37] Steffen Rendle. Scaling factorization machines to relational data. In *Proceedings of the VLDB Endowment*, volume 6, pages 337–348. VLDB Endowment, 2013.
- [38] Steffen Rendle, Li Zhang, and Yehuda Koren. On the difficulty of evaluating baselines: A study on recommender systems. *arXiv preprint arXiv:1905.01395*, 2019.
- [39] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.
- [40] Chris Russell. Efficient search for diverse coherent explanations. *arXiv preprint arXiv:1901.04909*, 2019.
- [41] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.
- [42] Joan E. Solman. Youtube’s ai is the puppet master over most of what you watch. *CNET*. URL <https://www.cnet.com/news/youtube-ces-2018-neal-mohan/>.
- [43] Harald Steck. Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 125–132. ACM, 2011.
- [44] Harald Steck. Calibrated recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 154–162. ACM, 2018.
- [45] Nava Tintarev. Presenting diversity aware recommendations: Making challenging news acceptable. 2017.
- [46] Zeynep Tufekci. Youtube, the great radicalizer. *The New York Times*, March 2018. URL <https://www.nytimes.com/2018/03/10/opinion/sunday/youtube-politics-radical.html>.
- [47] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 10–19. ACM, 2019.
- [48] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31(2):2018, 2017.
- [49] B. Webb. Netflix update: Try this at home. 2006. URL <http://sifter.org/simon/journal/20061211.html>, 2006.
- [50] Zeng Wei, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. Reinforcement learning to rank with markov decision process. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 945–948. ACM, 2017.
- [51] Longqi Yang, Michael Sobolev, Yu Wang, Jenny Chen, Drew Dunne, Christina Tsangouri, Nicola Dell, Mor Naaman, and Deborah Estrin. How intention informed recommendations modulate choices: A field study of spoken word content. In *The World Wide Web Conference*, pages 2169–2180. ACM, 2019.
- [52] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management*, pages 337–348. Springer, 2008.