Quantifying Availability and Discovery in Recommender Systems via Stochastic Reachability

Mihaela Curmei [* 1] **Sarah Dean** [* 1] **Benjamin Recht** [1]

Abstract

In this work, we consider how preference models in interactive recommendation systems determine the availability of content and users' opportunities for discovery. We propose an evaluation procedure based on stochastic reachability to quantify the maximum probability of recommending a target piece of content to an user for a set of allowable strategic modifications. This framework allows us to compute an upper bound on the likelihood of recommendation with minimal assumptions about user behavior. Stochastic reachability can be used to detect biases in the availability of content and diagnose limitations in the opportunities for discovery granted to users. We show that this metric can be computed efficiently as a convex program for a variety of practical settings, and further argue that reachability is not inherently at odds with accuracy. We demonstrate evaluations of recommendation algorithms trained on large datasets of explicit and implicit ratings. Our results illustrate how preference models, selection rules, and user interventions impact reachability and how these effects can be distributed unevenly.

1. Introduction

Through recommendation systems, personalized preference models mediate access to many types of information on the internet. Aiming to surface content that will be consumed, enjoyed, and highly rated, these models are primarily designed to accurately predict individuals' preferences. However, it is important to look beyond measures of accuracy towards notions of access. The focus on improving recommender model accuracy favors systems in which human behavior becomes as predictable as possible—effects which have been implicated in unintended consequences like polarization or radicalization.

We focus on questions of access and agency by adopting an *interventional* lens, which considers arbitrary and strategic user actions. We expand upon the notion of reachability first proposed by Dean et al. (2020), which measures the ability of an individual to influence a recommender model to select a certain piece of content. We define a notion of stochastic reachability which quantifies the maximum achievable likelihood of a given recommendation in the presence of strategic interventions. This metric provides an upper bound on the ability of individuals to discover specific content, thus isolating unavoidable biases within preference models from those due to user behavior.

Our primary contribution is the definition of metrics based on stochastic reachability which capture the possible outcomes of a round of system interactions, including the *availability* of content and *discovery* possibilities for individuals. In Section 3, we show that they can be computed by solving a convex optimization problem for a class of relevant recommenders. In Section 4, we draw connections between the stochastic and deterministic settings. This perspective allows us to describe the relationship between agency and stochasticity and further to argue that there is not an inherent trade-off between reachability and model accuracy. Finally, we present an audit of recommendation systems using a variety of datasets and preference models. We explore how design decisions influence reachability and the extent to which biases in the training datasets are propagated.

1.1. Related Work

The recommender systems literature has long proposed a variety of other metrics for evaluation, including notions of novelty, serendipity, diversity, and coverage (Herlocker et al., 2004; Castells et al., 2011). There is a long history of measuring and mitigating bias in recommendation systems (Chen et al., 2020). Empirical investigations have found evidence of popularity and demographic bias in domains including movies, music, books, and hotels (Abdollahpouri et al., 2019; Ekstrand et al., 2018a;b; Jannach et al., 2015). Alternative metrics are useful both for diagnosing biases and as objectives for post-hoc mitigating techniques

---

[*]Equal contribution [1]Department of Electical Engineering and Computer Sciences, University of California, Berkeley, USA. Correspondence to: Mihaela Curmei <mcurmei@berkeley.edu>, Sarah Dean <sarahdean@eecs.berkeley.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

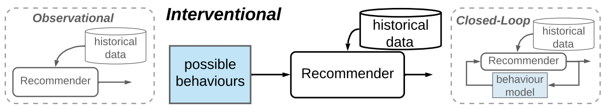

Figure 1. Conceptual framings of recommendation systems consider user behaviors to varying degrees. In this work we focus on evaluating interventional properties.

such as calibration (Steck, 2018) and re-ranking (Singh & Joachims, 2018). A inherent limitation of these approaches is that they focus on *observational* bias induced by preference models, i.e. examining the result of a single round of recommendations without considering individuals' behaviors. While certainly useful, they fall short of providing further understanding into the interactive nature of recommendation systems.

The behavior of recommendation systems over time and in *closed-loop* is still an open area of study. It is difficult to definitively link observational evidence of radicalization (Ribeiro et al., 2020; Faddoul et al., 2020) to proprietary recommendation algorithms. Empirical studies of human behavior find mixed results on the relationship between recommendation and content diversity (Nguyen et al., 2014; Flaxman et al., 2016). Simulation studies (Chaney et al., 2018; Yao et al., 2021; Krauth et al., 2020) and theoretical investigations (Dandekar et al., 2013) shed light on phenomena in simplified settings, showing how homogenization, popularity bias, performance, and polarization depend on assumed user behavior models. Even ensuring accuracy in sequential dynamic settings requires contending with closed-loop behaviors. Recommendation algorithms must mitigate biased sampling in order to learn underlying user preference models, using causal inference based techniques (Schnabel et al., 2016; Yang et al., 2018) or by balancing exploitation and exploration (Kawale et al., 2015; Mary et al., 2015). Reinforcement Learning algorithms contend with these challenges while considering a longer time horizon (Chen et al., 2019; Ie et al., 2019), implicitly using data to exploit user behavior.

Our work eschews behavior models in favor of an *interventional* framing which considers a variety of possible user actions. Giving users control over their recommendations has been found to have positive effects, while reducing agency has negative effects (Harper et al., 2015; Lukoff et al., 2021). The formal perspective we take on agency and access in recommender systems was first introduced by Dean et al. (2020), and is closely related to a body on work on recourse in consequential decision making (Ustun et al., 2019; Karimi et al., 2020). We build on this work to consider stochastic recommendation policies.

2. Metrics Based on Reachability

2.1. Stochastic Recommender Setting

We consider systems composed of n individuals as well as a collection of m pieces of content. For consistency with the recommender systems literature, we refer to individuals as users, pieces of content as items, and expressed preferences as ratings. We will denote a rating by user u of item i as $r_{ui} \in \mathcal{R}$, where $\mathcal{R} \subseteq \mathbb{R}$ denotes the space of values which ratings can take. For example, ratings corresponding to the percentage of a video watched would have $\mathcal{R} = [0, 1]$ while discrete star ratings would have $\mathcal{R} = \{1, 2, 3, 4, 5\}$. The number of *observed ratings* will generally be much smaller than the total number of possible ratings, and we denote by $\Omega_u \subseteq \{1, \dots, m\}$ the set of items seen by the user u. The goal of a recommendation system is to understand the preferences of users and recommend relevant content.

In this work, we focus on the common setting in which recommenders are the composition of a *scoring function* ϕ with *selection rule* π (Figure 2). The scoring function models the preferences of users. It is constructed based on historical data (e.g. observed ratings, user/item features) and returns a score for each user and item pair. For a given user u and item i, we denote $s_{ui} \in \mathbb{R}$ to be the associated score, and for user u we will denote by $\mathbf{s}_u \in \mathbb{R}^m$ the vector of scores for all items. A common example of a scoring function is a machine learning model which predicts future ratings based on historical data.

We will focus on the way that scores are updated after a round of user interaction. For example, if a user consumes and rates several new items, the recommender system should update the scores in response. Therefore, we parameterize the score function by an update rule, so that the new score vector is $\mathbf{s}_u^+ = \phi_u(\mathbf{a})$, where $\mathbf{a} \in \mathcal{A}_u$ represents actions taken by user u and $\mathcal{A}_u$ represents the set of all possible actions. Thus ϕ_u encodes the historical data, the preference model class, and the update algorithm. The action space $\mathcal{A}_u$ represents possibilities for system interaction, encoding for example limitations due to user interface design. We define the form of the score update function and discuss the action space in more detail in Section 3.

The selection rule π is a policy which, for given user u and scores $\mathbf{s}_u$, selects one or more items from a set of specified *target items* $\Omega_u^t \subseteq \{1, \dots, m\}$ as the next recommendation. The simplest selection rule is a top-1 policy, which is a deterministic rule that selects the item with the highest score for each user. A simple stochastic rule is the ϵ-greedy policy which with probability $1 - \epsilon$ selects the top scoring item and with probability ϵ chooses uniformly from the remaining items. Many additional approaches to recommendation can be viewed as the composition of a score function with a selection policy. This setting also encompasses implicit

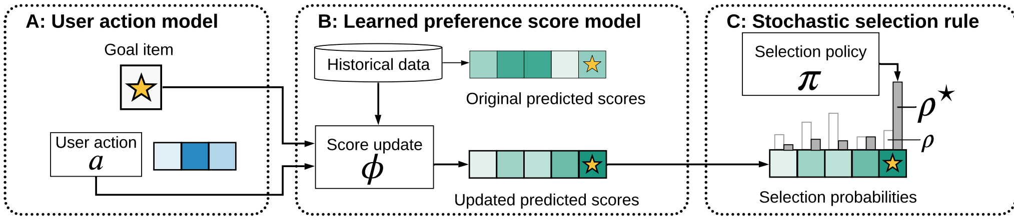

Figure 2. We audit recommender systems under a user action model (A), learned preference model (B), and stochastic selection rule (C).

feedback scenarios, where clicks or other behaviors are defined as or aggregated into "ratings." Many recommendation algorithms, even those not specifically motivated by regression, include an intermediate score prediction step, e.g. point-wise approaches to ranking. Further assumptions in Section 3 will not capture the full complexity of other techniques such as pairwise ranking and slate-based recommendations. We leave such extensions to future work.

In this work, we are primarily interested in stochastic policies which select items according to a probability distribution on the scores $\mathbf{s}_u$ parametrized by a exploration parameter. Policies of this form are often used to balance exploration and exploration in online or sequential learning settings. A stochastic selection rule recommends an item i according to $\mathbb{P}\left(\pi(\mathbf{s}_u, \Omega_u^t) = i\right)$, which is 0 for all non-target items $i \notin \Omega_u^t$. For example, to select among items that have not yet been seen by the user, the target items are set as $\Omega_u^t = \Omega_u^{\mathsf{c}}$ (recalling that Ω_u denotes the set of items seen by the user u). Deterministic policies are a special case of stochastic policies, with a degenerate distribution.

Stochastic policies have been proposed in the recommender system literature to improve diversity (Christoffel et al., 2015) or efficiently explore in a sequential setting (Kawale et al., 2015). By balancing exploitation of items with high predicted ratings against explorations of items with lower predictions, preferences can be estimated so that future predicted ratings are more accurate. However, our work decidedly does not take a perspective based on accuracy. Rather than supposing that users' reactions are predictable, we consider a perspective centered on agency and access.

2.2. Reachability

First defined in the context of recommendations by Dean et al. (2020), an item i is *deterministically reachable* by a user u if there is some allowable modification to the user's ratings $\mathbf{r}_u$ that causes item to be recommended. Allowable modifications can include history edits, such as removing or changing ratings of previously rated items. They can also include future looking modifications which assign ratings to a subset of unseen items.

In the setting where recommendations are made stochastically, we define an item i to be ρ *reachable* by a user u if there is some allowable action $\mathbf{a}$ such that the updated probability that item i is recommended after applying action $\mathbf{a}$; $\mathbb{P}(\pi(\phi_u(\mathbf{a}), \Omega_u^t) = i)$ is larger than ρ. The maximum ρ reachability for a user-item pair is defined as the solution to the following optimization problem:

$$\rho^\star(u, i) = \max_{\mathbf{a} \in \mathcal{A}_u} \; P(\pi(\phi_u(\mathbf{a}), \Omega_u^t) = i). \tag{1}$$

We will also refer to $\rho^\star(u, i)$ as "max reachability."

For example, in the case of ε-greedy policy, $\rho^\star(u, i) = 1 - \varepsilon$ if item i is deterministically reachable by user u, and is $\varepsilon/(|\Omega_u^t| - 1)$ otherwise.

By measuring the maximum achievable probability of recommending an item to a user, we are characterizing a granular metric of *access* within the recommender system. It can also be viewed as an upper bound on the likelihood of recommendation with minimal assumptions about user behavior. It may be illuminating to contrast this measure with a notion of expected reachability. Computing expected reachability would require specifying the distribution over user actions, which would amount to modelling human behavior. In contrast, max reachability requires specifying only the constraints arising from system design choices to define $\mathcal{A}_u$ (e.g. the user interface). By computing max reachability, we focus our analysis on the design of the recommender system, and avoid conclusions which are dependent on behavioral modelling choices.

Two related notions of user agency with respect to a target item i are *lift* and *rank gain*. The lift measures the ratio between the maximum achievable probability of recommendation and the baseline:

$$\lambda^\star(u, i) = \frac{\rho^\star(u, i)}{\rho_0(u, i)} \tag{2}$$

where the baseline $\rho_0(u, i)$ is defined to capture the default probability of recommendation in the absence of strategic behavior, e.g. $P\left(\pi\left(\mathbf{s}_u, \Omega_u^t\right) = i\right)$.

The rank gain for an item i is the difference in the ranked position of the item within the original list of scores $\mathbf{s}_u$ and its rank within the updated list of scores $\mathbf{s}_u^+$.

Lift and rank gain are related concepts, but ranked position is combinatorial in nature and thus difficult to optimize for directly. They both measure agency because they compare the default behavior of a system to its behavior under a strategic intervention by the user. Given that recommenders are designed with personalization in mind, we view the ability of users to influence the model in a positive light. This is in contrast to much recent work in robust machine literature where strategic manipulation is undesirable.

2.3. Diagnosing System Limitations

The analysis of stochastic reachability can be used to audit recommender systems and diagnose systemic biases from an interventional perspective (Figure 1). Unlike studies of observational bias, these analyses take into account system interactivity. Unlike studies of closed-loop bias, there is no dependence on a behavior model. Because max reachability considers the best case over possible actions, it isolates structural biases from those caused in part by user behavior.

Max reachability is a metric defined for each user-item pair, and disparities across users and items can be detected through aggregations. Aggregating over target items gives insight into a user's ability to discover content, thus detecting users who have been "pigeonholed" by the algorithm. Aggregations over users can be used to compare how the system makes items available for recommendation.

We define the following user- and item-based aggregations:

$$D_u = \sum_{i \in \Omega_u^t} \frac{\mathbf{1}\{\rho_{ui} > \rho_t\}}{|\Omega_u^t|}, \quad A_i = \frac{\sum_u \rho_{ui} \mathbf{1}\{i \in \Omega_u^t\}}{\sum_u \mathbf{1}\{i \in \Omega_u^t\}} \quad (3)$$

The discovery D_u is the proportion of target items that have a high chance of being recommended, as determined by the threshold ρ_t. A natural threshold is the better-than-uniform threshold, $\rho_t = 1/|\Omega_u^t|$, recalling that Ω_u^t is the set of target items. When $\rho_{ui} = \rho_0(u, i)$, baseline discovery counts the number of items that will be recommended with better-than-uniform probability and is determined by the spread of the recommendation distribution. When $\rho_{ui} = \rho^\star(u, i)$, discovery counts the number of items that a user *could* be recommended with better-than-uniform probability in the best case. Low best-case discovery means that the recommender system inherently limits user access to content.

The item availability A_i is the average likelihood of recommendation over all users who have item i as a target. It can be thought of as the chance that a uniformly selected user will be recommended item i. When $\rho_{ui} = \rho_0(u, i)$, the baseline availability measures the prevalence of the item in the recommendations. When $\rho_{ui} = \rho^\star(u, i)$, availability measures the prevalence of an item in the best case. Low best-case availability means that the recommender system inherently limits the distribution of a given item.

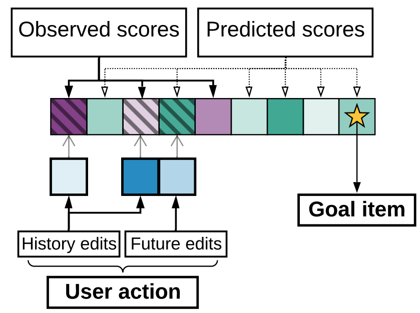

Figure 3. User action space: The shade represents the magnitude of historical (purple) or predicted (green) rating. The *action items* are marked with diagonal lines; they can be strategically modified to maximize the recommendation probability of the *goal item* (star). The value of the user action is shaded in blue.

3. Computing Reachability

3.1. Affine Recommendation

In this section, we consider a restricted class of recommender systems for which the max reachability problem can be efficiently solved via convex optimization.

User action model We suppose that users interact with the system through expressed preferences, and thus actions are updates to the vector $\mathbf{r}_u \in \mathcal{R}^m$, a sparse vector of observed ratings. For each user, the action model is based on distinguishing between *action* and *immutable* items.

Let $\Omega_u^{\mathcal{A}}$ denote the set of action items for which the ratings can be strategically modified by the user u. Then the action set $\mathcal{A}_u = \mathcal{R}^{|\Omega_u^{\mathcal{A}}|}$ corresponds to changing or setting the value of these ratings. Figure 3 provides an illustration. The action set should be defined to correspond to the interface through which a user interacts with the recommender system. For example, it could correspond to a display panel of "previously viewed" or "up next" items.

The updated rating vector $\mathbf{r}_u^+ \in \mathcal{R}^m$ is equal to $\mathbf{r}_u$ at the indices corresponding to immutable items and equal to the action $\mathbf{a}$ at the action items. Note the partition into action and immutable is distinct from earlier partition of items into observed and unobserved; action items can be both seen (history edits) and unseen (future reactions), as illustrated in Figure 2 (A). For the reachability problem, we will consider a set of target items Ω_u^t that does not intersect with the action items $\Omega_u^{\mathcal{A}}$. Depending on the specifics of the recommendation setting, we may also require that it does not intersect with the previously rated items Ω_u.

We remark that additional user or item features used for scoring and thus recommendations could be incorporated into this framework as either mutable or immutable features. The only computational difficulty arises when mutable features are discrete or categorical.

Recommender model The recommender model is composed of a scoring function ϕ and a selection function π, which we now specify. We consider *affine score update functions* where for each user, scores are determined by an affine function of the action: $\mathbf{s}_u^+ = \phi_u(\mathbf{a}) = B_u\mathbf{a} + \mathbf{c}_u$ where $B_u \in \mathbb{R}^{m\times|\Omega_u^{\mathcal{A}}|}$ and $\mathbf{c}_u \in \mathbb{R}^m$ are model parameters determined in part by historical data. Such a scoring model arises from a variety of preference models, as shown in the examples in Section 3.3.

We now turn to the selection component of the recommender, which translates the score $\mathbf{s}_u$ into a probability distribution over target items. The stochastic policy we consider is:

Definition 1. Soft-max selection
For $i \in \Omega_u^t$, the probability of item selection is given by

$$P(\pi_\beta(\mathbf{s}_u, \Omega_u^t) = i) = \frac{e^{\beta s_{ui}}}{\sum_{j\in\Omega_u^t} e^{\beta s_{uj}}}\ .$$

This form of stochastic policy samples an item according to a Boltzmann distribution defined by the predicted scores (Figure 2C). Distributions of this form are common in machine learning applications, and are known as Boltzmann sampling in reinforcement learning or online learning settings (Wei et al., 2017; Cesa-Bianchi et al., 2017).

3.2. Convex Optimization

We now show that under affine score update models and soft-max selection rules, the maximum stochastic reachability problem can be solved by an equivalent convex problem. First notice that for a soft-max selection rule with parameter β, we have that

$$\log\left(P(\pi_\beta(\mathbf{s}_u, \Omega_u^t) = i)\right) = \beta s_{ui} - \underset{j\in\Omega_u^t}{\mathrm{LSE}}\left(\beta s_{uj}\right)$$

where LSE is the log-sum-exp function.

Maximizing stochastic reachability is equivalent to minimizing its negative log-likelihood. Letting $\mathbf{b}_{ui}$ denote the ith row of the action matrix B_u and substituting the form of the score update rule, we have the equivalent optimization problem:

$$\min_{\mathbf{a}\in\mathcal{A}_u} \underset{j\in\Omega_u^t}{\mathrm{LSE}}\left(\beta(\mathbf{b}_{uj}^\top\mathbf{a} + c_{uj})\right) - \beta(\mathbf{b}_{ui}^\top\mathbf{a} + c_{ui}) \tag{4}$$

If the optimal value to (4) is $\gamma^\star(u,i)$, then the optimal value for (1) is given by $\rho^\star(u,i) = e^{-\gamma^\star(u,i)}$.

The objective in (4) is convex because log-sum-exp is a convex function, affine functions are convex, and the composition of a convex and an affine function is convex. Therefore, whenever the action space $\mathcal{A}_u$ is convex, so is the optimization problem. The size of the decision variable scales with the dimension of the action, while the objective function relies on a matrix-vector product of size $|\Omega_u^t| \times |\mathcal{A}_u|$. Being able to solve the maximum reachability problem quickly is of interest, since auditing an entire system requires computing $\rho^\star$ for many user and item pairs.

3.3. Examples

In this section we review examples of common preference models and show how the score updates have an affine form.

Example 1. Matrix factorization models compute scores as rating predictions so that $S = PQ^\top$, where $P \in \mathbb{R}^{n\times d}$ and $Q \in \mathbb{R}^{m\times d}$ are respectively user and item factors for some latent dimension d. They are learned via the optimization

$$\min_{P,Q} \sum_u \sum_{i\in\Omega_u} \|\mathbf{p}_u^\top \mathbf{q}_i - r_{ui}\|_2^2\ .$$

Under a stochastic gradient descent minimization scheme with step size α, the one-step update rule for a user factor is

$$\mathbf{p}_u^+ = \mathbf{p}_u - \alpha \sum_{i\in\Omega_u^{\mathcal{A}}} \left(\mathbf{q}_i\mathbf{q}_i^\top\mathbf{p}_u - \mathbf{q}_i r_{ui}\right),$$

Notice that this expression is affine in the action items. Therefore, we have an affine score function:

$$\phi_u(\mathbf{a}) = Q\mathbf{p}_u^+ = Q\left(\mathbf{p}_u - \alpha Q_{\mathcal{A}}^\top Q_{\mathcal{A}}\mathbf{p}_u - \alpha Q_{\mathcal{A}}^\top\mathbf{a}\right)$$

where we define $Q_{\mathcal{A}} = Q_{\Omega_u^{\mathcal{A}}} \in \mathbb{R}^{|\Omega_u^{\mathcal{A}}|\times d}$. Therefore,

$$B_u = -\alpha Q Q_{\mathcal{A}}^\top, \quad \mathbf{c}_u = Q\left(\mathbf{p}_u - \alpha Q_{\mathcal{A}}^\top Q_{\mathcal{A}}\mathbf{p}_u\right)\ .$$

Example 2. Neighborhood models compute scores as rating predictions by a weighted average, with:

$$s_{ui} = \frac{\sum_{j\in\mathcal{N}_i} w_{ij} r_{uj}}{\sum_{j\in\mathcal{N}_i} |w_{ij}|}$$

where w_{ij} are weights representing similarities between items and $\mathcal{N}_i$ is a set of indices of previously rated items in the neighborhood of item i. Regardless of the details of how these parameters are computed, the predicted scores are a linear function of observed scores: $\mathbf{s}_u = W\mathbf{r}_u$.

Therefore, the score updates take the form

$$\phi_u(\mathbf{a}) = W\mathbf{r}_u^+ = \underbrace{W\mathbf{r}_u}_{\mathbf{c}_u} + \underbrace{WE_{\Omega_u^{\mathcal{A}}}}_{B_u}\mathbf{a}$$

where $E_{\Omega_u^{\mathcal{A}}}$ selects rows of W corresponding to action items.

In both examples, the action matrices can be decomposed into two terms. The first is a term that depends only on the preference model (e.g. item factors Q or weights W), while the second is dependent on the user action model (e.g. action item factors $Q_{\mathcal{A}}$ or action selector $E_{\Omega_u^{\mathcal{A}}}$).

For simplicity of presentation, the examples above leave out model bias terms, which are common in practice. Incorporating these model biases changes only the definition of the affine term in the score update expression. We include the full action model derivation with biases in Appendix A, along with additional examples.

4. Geometry of Reachability

In this section, we explore the connection between stochastic and deterministic reachability to illustrate how both randomness and agency contribute to discovery as defined by the max reachability metric. We then argue by example that it is possible to design preference models that guarantee deterministic reachability, and that doing so does not induce accuracy trade-offs.

4.1. Connection to Deterministic Recommendation

We now explore how the softmax style selection rule is a relaxation of top-1 recommendation. For larger values of β, the selection rule distribution becomes closer to the deterministic top-1 rule. This also means that the stochastic reachability problem can be viewed as a relaxation of the top-1 reachability problem.

In stochastic settings it is relevant to inquire the extent to which randomness impacts discovery and availability. In the deterministic setting, the reachability of an item to a user is closely tied to agency—the ability of a user to influence their outcomes. The addition of randomness induces exploration, but not in a way that is controllable by users. In the following result, we show how this trade-off manifests in the max reachability metric itself. The proof, as well as proofs of results to follow, are in Appendix B.

Proposition 1. *Consider the stochastic reachability problem for a β-softmax selection rule as $\beta \to \infty$. Then if an item i is top-1 reachable by user u, $\rho^\star(u,i) \to 1$. In the opposite case that item i* is not *top-1 reachable, we have that $\rho^\star(u,i) \to 0$.*

This connection yields insight into the relationship between max reachability, randomness, and agency in stochastic recommender systems. For items which are top-1 reachable, larger values of β result in larger $\rho^\star$, and in fact the largest possible max reachability is attained as $\beta \to \infty$, i.e. there is no randomness. On the other hand, if β is too large, then items which are not top-1 reachable will have small $\rho^\star$. There is some optimal finite $\beta \geq 0$ that maximizes $\rho^\star$ for top-1 unreachable items. Therefore, we see a delicate balance when it comes to ensuring access with randomness. Viewed in another light, this result says that for a fixed $\beta \gg 1$, deterministic top-1 reachability ensures that $\rho^\star$ will be close to 1. We explore this perspective in the next section.

4.2. Reachability Without Sacrificing Accuracy

Specializing to affine score update models, we now highlight how parameters of the preference and action models play a role in determining max reachability. Building on the connection to deterministic reachability, we make use of results about model and action space geometry from Dean et al. (2020).

Proposition 2. *If $\mathbf{b}_{ui}$ is a vertex on the convex hull of $\{\mathbf{b}_{uj}\}_{j\in\Omega_u^t}$ and actions are real-valued, then $\rho_{ui}^\star \to 1$ as $\beta \to \infty$.*

This result highlights how the geometry of the score model determines when it is preferable for the system to have minimal exploration, from the perspective of reachability.

We now consider whether relevant geometric properties of the model are predetermined by the goal of accurate prediction. Is there a tension between ensuring reachability and accuracy? We answer in the negative by presenting a construction for the case of matrix factorization models. Our result shows that the item and user factors (P and Q) can be slightly altered such that all items become top-1 reachable at no loss of predictive accuracy. The construction expands the latent dimension of the user and item factors by one and relies on sufficiently rich action items; we make this notion of richness precise in Appendix B.

Proposition 3. *Consider the MF model with user factors $P \in \mathbb{R}^{n\times d}$ and item factors $Q \in \mathbb{R}^{m\times d}$. Further consider any user u with a sufficiently rich set of at least $d+1$ action items and real-valued actions. Then there exist $\tilde{P} \in \mathbb{R}^{n\times d+1}$ and $\tilde{Q} \in \mathbb{R}^{m\times d+1}$ such that $PQ^\top = \tilde{P}\tilde{Q}^\top$ and under this model, $\rho^\star(u,i) \to 1$ as $\beta \to \infty$ for all target items $i \in \Omega_u^t$.*

The existence of such a construction demonstrates that there is not an unavoidable trade-off between accuracy and reachability in recommender systems.

5. Audit Demonstration

5.1. Experimental Setup

Datasets We evaluate[1] max ρ reachability in settings based on three popular recommendation datasets: MovieLens 1M (ML-1M) (Harper & Konstan, 2015), LastFM 360K (Celma, 2010) and MIcrosoft News Dataset (MIND) (Wu et al., 2020). ML-1M is a dataset of 1 through 5 explicit ratings of movies, containing over one million recorded ratings; we do not perform any additional pre-processing. LastFM is

[1]Reproduction code available at `github.com/modestyachts/stochastic-rec-reachability`

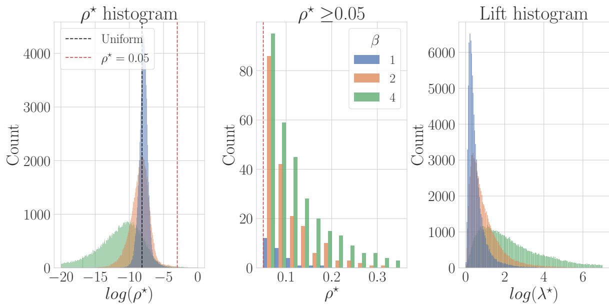

Figure 4. Left: Histogram of log max reachability values for $\beta = [1, 2, 4]$. Black dotted line denotes $\rho^\star$ for uniformly random recommender. Center: Histogram of $\rho^\star > 0.05$ (red dotted line). Right: Histogram of log-lifts. Reachability evaluated on ML-1M for $K = 5$ Random Future action space and a LibFM model.

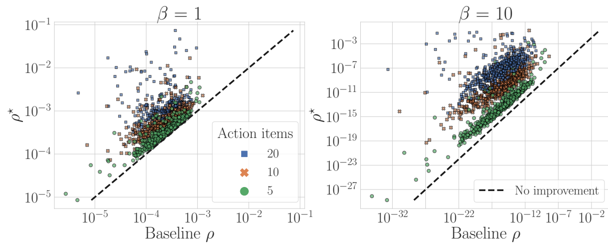

Figure 5. Log scale scatterplot of $\rho^\star$ values against baseline ρ for $K \in [5, 10, 20]$. Colors indicate action space size K. We compare low (left) and high (right) stochasticity. Reachability evaluated on ML-1M for Random Future action space and a LibFM model.

an implicit rating dataset containing the number of times a user has listened to songs of an artist. We used the version of the LastFM dataset preprocessed by Shakespeare et al. (2020). For computational tractability, we select a random subset of 10% of users and 10% artists and define ratings as $r_{ui} = \log(\#\text{listens}(u, i) + 1)$ to ensure that rating matrices are well conditioned. MIND is an implicit rating dataset containing clicks and impressions data. We use a subset of 50K users and 40K news articles spanning 17 categories and 247 subcategories. We transform news level click data into subcategory level aggregation and define the rating associated with a user-subcategory pair as a function of the number of times that the user clicked on news from that subcategory: $r_{ui} = \log(\#\text{clicks}(u, i) + 1)$. Appendix C.1 contains further details.

Preference models We consider two preference models: one based on matrix factorization (MF) as well as a neighborhood based model (KNN). We use the LibFM SGD implementation (Rendle, 2012) for the MF model and use the item-based k-nearest neighbors model implemented by Krauth et al. (2020). For each dataset and recommender model we perform hyper-parameter tuning using a 10%-90% test-train split. We report test performance in Table 1. See Appendix C.2 for details about tuning. Prior to performing the audit, we retrain the recommender models with the full dataset.

Reachability experiments To compute reachability, it is further necessary to specify additional elements of the recommendation pipeline: the user action model, the set of target items, and the soft-max selection parameter.

We consider three types of user action spaces: *History Edits*, *Future Edits*, and *Next K* in which users can strategically modify the ratings associated to K randomly chosen items from their history, K randomly chosen unobserved items, or the top-K items according to the baseline scores of the preference model. For each of the action spaces we consider a range of K values. We further constrain actions to lie in an interval corresponding to the rating range, using $[1, 5]$ for movies and $[0, 10]$ for music and news.

In the case of movies (ML-1M) we consider target items to be all items that are neither seen nor action items. In the case of music and news recommendations (LastFM & MIND), the target items are all the items with the exception of action items. This reflects an assumption that music created by a given artist or news within a particular subcategory can be consumed repeatedly, while movies are viewed once.

For each dataset and recommendation pipeline, we compute max reachability for soft-max selection rules parametrized by a range of β values. Due to the computational burden of large dense matrices, we compute metrics for a subset of users and target items sampled uniformly at random. For details about runtime, see Appendix C.3.

5.2. Impact of Recommender Pipeline

We begin by examining the role of recommender pipeline components: stochasticity of item selection, user action models, and choice of preference model. All presented experiments in this section use the ML-1M dataset.

These experiments show that more stochastic recommendations correspond to higher average max reachability values, whereas more deterministic recommenders have a more disparate impact, with a small number of items achieving higher $\rho^\star$. We also see that the impact of the user action space differs depending on the preference model. For neighborhood based preference models, strategic manipulations to the history are most effective at maximizing reachability, whereas manipulations of the items most likely to be recommended next are ineffective.

Role of stochasticity We investigate the role of the β parameter in the item selection policy. Figure 4 illustrates the relationship between the stochasticity of the selection policy and max reachability. There are significantly more target items with better than random reachability for low values of

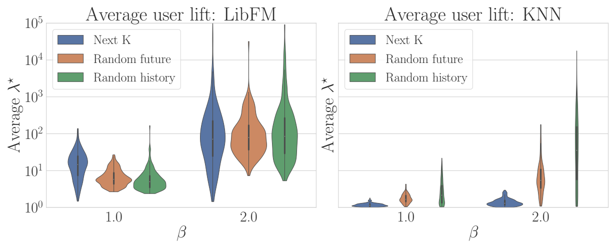

Figure 6. The distribution of average lifts (a notion of agency) over users. Colors indicate different user action spaces for LibFM (left) and KNN (right) on ML-1M.

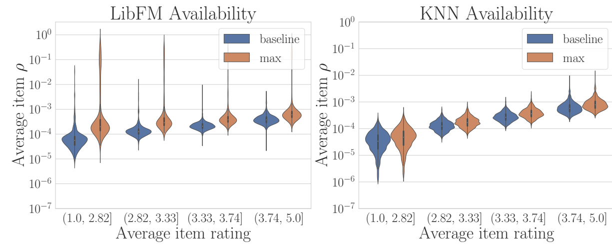

Figure 7. Comparison of baseline and best case availability of content, across four popularity categories for LibFM (left) and KNN (right) preference models. Reachability evaluated on ML-1M for *Next 10* action space with $\beta = 2$.

β. However, higher values of β yield more items with high reachability potential ($> 5\%$ likelihood of recommendation). These items are typically items that are top-1 or close to top-1 reachable. While lower β values provide better reachability on average and higher β values provide better reachability at the "top", higher β uniformly out-performs lower β values in terms of the lift metric. This suggests that larger β corresponds to more user agency, since the relative effect of strategic behavior is larger. However, note that for very large values of β, high lift values are not so much the effect of improved reachability as they are due to very low baseline recommendation probabilities.

Role of user action model We now consider different action space sizes. In Figure 5 we plot max reachability for target items of a particular user over varying levels of selection rule stochasticity and varying action space sizes. Larger action spaces correspond to improved item reachability for all values of β. However, increases in the number of action items have a more pronounced effect for larger β values.

While increasing the size of the action space uniformly improves reachability, the same cannot be said about the type of action space. For each user, we compute the average lift over target items as a metric for user agency in a recommender (Figure 6). For LibFM, the choice of action space does not strongly impact the average user lift, though *Next K* displays more variance across users than the other two. However, for Item KNN, there is a stark difference between *Next K* and and random action spaces.

Role of preference model As Figure 6 illustrates, a system using LibFM provides more agency on average than one using KNN. We now consider how this relates to properties of the preference models. First, consider the fact that for LibFM, there is higher variance among user-level average lifts observed for *Next K* action space compared with random action spaces. This can be understood as resulting from the user-specific nature of *Next K* recommended items. On the other hand, random action spaces are user independent, so it is not surprising that there is less variation across users.

In a neighborhood-based model users have leverage to increase the ρ reachability only for target items in the neighborhood of action items. In the case of KNN, the next items up for recommendation are in close geometrical proximity to each other. This limits the opportunity for discovery of more distant items for *Next K* action space. On the other hand, the action items are more uniformly over space of item ratings in random action models, thus contributing to much higher opportunities for discovery. Additionally, we see that *History Edits* displays higher average lift values than *Future Edits*. We posit that this is due to the fact that editing K items from the history leads to a larger ratio of strategic to non-strategic items.

5.3. Bias in Movie, Music, and News Recommendation

We futher compare aggregated stochastic reachability against properties of user and items to investigate bias. We aggregate baseline and max reachability to compute user-level metrics of discovery and item-level metrics of availability. The audit demonstrates popularity bias for items with respect to baseline availability. This bias persists in the best case for neighborhood based recommenders and is thus unavoidable, whereas it could be mitigated for MF recommenders. User discovery aggregation reveals inconclusive results with weak correlations between the length of users' experience and their ability to access content.

Popularity bias In Figure 7, we plot the baseline and best case item availability (as in (3)) to investigate popularity bias. We consider popularity defined by the average rating of an item in a dataset. Another possible definition of popularity is rating frequency, but for this definition we did not observe any discernable bias. For both LibFM and KNN models, the baseline availability displays a correlation with item popularity, with Spearman's rank-order correlations of $r_s = 0.87$ and $r_s = 0.95$. This suggests that as recommendations are made and consumed, more popular items will be recommended at disproportionate rates.

Furthermore, the best case availability for KNN displays a similar trend ($r_s = 0.94$), indicating that the propagation of

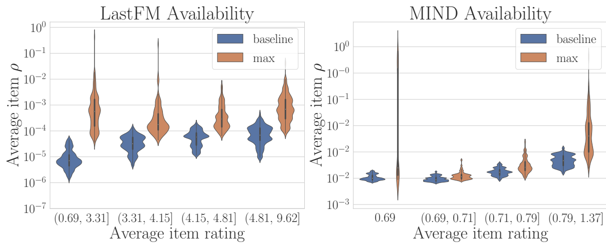

Figure 8. Comparison of baseline and best case availability of content for four popularity categories for LastFM (left) and MIND (right) with *Next 10* actions, LibFM model, and $\beta = 2$.

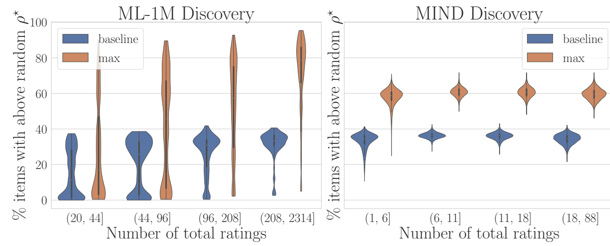

Figure 9. Comparison of baseline and best case fraction of items with better than random $\rho^\star$, grouped across four levels of user history length. Reachability evaluated on ML-1M (left) and MIND (right) for *Next 10* action space, $\beta = 2$, and LibFM model.

popularity bias can occur independent of user behavior. This does not hold for LibFM, where the best case availability is less clearly correlated with popularity ($r_s = 0.57$). The lack of correlation for best case availability holds in the additional settings of music artist and news recommendation with the LibFM model (Figure 8). Our audit does not reveal an unavoidable systemic bias for LibFM recommender, meaning that any biases observed in deployment are due in part to user behaviour. In contrast, we see a systematic bias for the KNN recommender, meaning that regardless of user actions, the popularity bias will propagate.

Experience bias To consider the opportunities for discovery provided to users, we perform user level aggregations of max reachability values as in (3). We investigate experience bias by considering how the discovery metric changes as a function of the number of different items a user has consumed so far, i.e. their experience. Figure 9 illustrates that experience is weakly correlated with baseline discovery for movie recommendation ($r_s = 0.48$), but not so much for news recommendation ($r_s = 0.05$). The best case discovery is much higher, meaning that users have the opportunity to discover many of their target items. However, the weak correlation with experience remains for best case discovery of movies ($r_s = 0.53$).

6. Discussion

In this paper, we generalize reachability as first defined by (Dean et al., 2020) to incorporate stochastic recommendation policies. We show that for linear preference models and soft-max item selection rules, max reachability can be computed via a convex program for a range of user action models. Due to this computational efficiency, reachability analysis can be used to audit recommendation algorithms. Our experiments illustrate the impact of system design choices and historical data on the availability of content and users' opportunities for discovery, highlighting instances in which popularity bias is inevitable regardless of user behavior.

The reachability metric provides an upper bound for discovery and availability within a recommendation system. While it has the benefit of making minimal assumptions about user behavior, the drawback is that it allows for perfectly strategic behaviors that would require users to have full knowledge of the internal structure of the model. The results of a reachability audit may not be reflective of probable user experience, and thus reachability acts as a necessary but not sufficient condition. Nonetheless, reachability audit can lead to actionable insights by identifying inherent limits in system design. They allow system designers to assess potential biases before releasing algorithmic updates into production. Moreover, as reachability depends on the choice of action space, such system-level insights might motivate user interface design: for example, a sidebar encouraging users to re-rate K items from their history.

We point to a few directions of interest for future work. Our result on the lack of trade-off between accuracy and reachability is encouraging. Minimum one-step reachability conditions could be efficiently incorporated into learning algorithms for preference models. It would also be interesting to extend reachability analysis to multiple interactions and longer time horizons.

Lastly, we highlight that the reachability lens presents a contrasting view to the popular line of work on robustness in machine learning. When human behaviors are the subject of classification and prediction, building "robustness" into a system may be at odds with ensuring agency. Because the goal of recommendation is personalization more than generalization, it would be appropriate to consider robust access over robust accuracy. This calls for questioning the current normative stance and critically examining system desiderata in light of usage context.

Acknowledgements

This research is generously supported in part by ONR awards N00014-20-1-2497 and N00014-18-1-2833, NSF CPS award 1931853, and the DARPA Assured Autonomy program (FA8750-18-C-0101). SD is supported by NSF GRF under Grant No. DGE 1752814.

References

Abdollahpouri, H., Mansoury, M., Burke, R., and Mobasher, B. The impact of popularity bias on fairness and calibration in recommendation. *arXiv preprint arXiv:1910.05755*, 2019.

ApS, M. *MOSEK Optimizer API for Python Release 9.0.88*, 2019. URL https://docs.mosek.com/9.0/pythonapi.pdf.

Castells, P., Vargas, S., and Wang, J. Novelty and diversity metrics for recommender systems: choice, discovery and relevance. 2011.

Celma, O. Music recommendation. In *Music recommendation and discovery*, pp. 43–85. Springer, 2010.

Cesa-Bianchi, N., Gentile, C., Lugosi, G., and Neu, G. Boltzmann exploration done right. *arXiv preprint arXiv:1705.10257*, 2017.

Chaney, A. J., Stewart, B. M., and Engelhardt, B. E. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 224–232, 2018.

Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., and He, X. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240*, 2020.

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 456–464, 2019.

Christoffel, F., Paudel, B., Newell, C., and Bernstein, A. Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 163–170, 2015.

Dacrema, M. F., Boglio, S., Cremonesi, P., and Jannach, D. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)*, 39(2):1–49, 2021.

Dandekar, P., Goel, A., and Lee, D. Biased assimilation, homophily, and the dynamics of polarization. In *Proceedings of the National Academy of Sciences*, pp. 5791–5796, 2013.

Dean, S., Rich, S., and Recht, B. Recommendations and user agency: the reachability of collaboratively-filtered information. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 436–445, 2020.

Desrosiers, C. and Karypis, G. A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, pp. 107–144, 2011.

Ekstrand, M. D., Tian, M., Azpiazu, I. M., Ekstrand, J. D., Anuyah, O., McNeill, D., and Pera, M. S. 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*, pp. 172–186. PMLR, 2018a.

Ekstrand, M. D., Tian, M., Kazi, M. R. I., Mehrpouyan, H., and Kluver, D. Exploring author gender in book rating and recommendation. In *Proceedings of the 12th ACM conference on recommender systems*, pp. 242–250, 2018b.

Faddoul, M., Chaslot, G., and Farid, H. A longitudinal analysis of youtube's promotion of conspiracy videos. *arXiv preprint arXiv:2003.03318*, 2020.

Flaxman, S., Goel, S., and Rao, J. M. Filter bubbles, echo chambers, and online news consumption. *Public opinion quarterly*, 80(S1):298–320, 2016.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Harper, F. M., Xu, F., Kaur, H., Condiff, K., Chang, S., and Terveen, L. Putting users in control of their recommendations. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 3–10, 2015.

Herlocker, J., Konstan, J., Terveen, L., and Riedl, J. Evaluating collaborative filtering recommender systems. *ACM transactions on information systems*, 22(1):5–53, 2004.

Ie, E., Jain, V., Wang, J., Narvekar, S., Agarwal, R., Wu, R., Cheng, H.-T., Chandra, T., and Boutilier, C. Slateq: A tractable decomposition for reinforcement learning with recommendation sets. 2019.

Jannach, D., Lerche, L., Kamehkhosh, I., and Jugovac, M. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25(5):427–491, 2015.

Karimi, A.-H., Barthe, G., Schölkopf, B., and Valera, I. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. *arXiv preprint arXiv:2010.04050*, 2020.

Kawale, J., Bui, H. H., Kveton, B., Tran-Thanh, L., and Chawla, S. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in neural information processing systems*, pp. 1297–1305, 2015.

Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, 2008.

Koren, Y. and Bell, R. Advances in collaborative filtering. *Recommender systems handbook*, pp. 77–118, 2015.

Krauth, K., Dean, S., Zhao, A., Guo, W., Curmei, M., Recht, B., and Jordan, M. I. Do offline metrics predict online performance in recommender systems? *arXiv preprint arXiv:2011.07931*, 2020.

Lukoff, K., Lyngs, U., Zade, H., Liao, J. V., Choi, J., Fan, K., Munson, S. A., and Hiniker, A. How the design of youtube influences user sense of agency. *arXiv preprint arXiv:2101.11778*, 2021.

Mary, J., Gaudel, R., and Preux, P. Bandits and recommender systems. In *International Workshop on Machine Learning, Optimization and Big Data*, pp. 325–336. Springer, 2015.

Nguyen, T. T., Hui, P.-M., Harper, F. M., Terveen, L., and Konstan, J. A. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, pp. 677–686, 2014.

Ning, X. and Karypis, G. Slim: Sparse linear methods for top-n recommender systems. In *2011 IEEE 11th International Conference on Data Mining*, pp. 497–506. IEEE, 2011.

Rendle, S. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012. ISSN 2157-6904.

Rendle, S., Zhang, L., and Koren, Y. On the difficulty of evaluating baselines: A study on recommender systems. *arXiv preprint arXiv:1905.01395*, 2019.

Ribeiro, M. H., Ottoni, R., West, R., Almeida, V. A., and Meira Jr, W. Auditing radicalization pathways on youtube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 131–141, 2020.

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*, pp. 1670–1679. PMLR, 2016.

Shakespeare, D., Porcaro, L., Gómez, E., and Castillo, C. Exploring artist gender bias in music recommendation. *arXiv preprint arXiv:2009.01715*, 2020.

Singh, A. and Joachims, T. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2219–2228, 2018.

Steck, H. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems*, pp. 154–162, 2018.

Steck, H. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, pp. 3251–3257, 2019.

Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 10–19, 2019.

Wei, Z., Xu, J., Lan, Y., Guo, J., and Cheng, X. Reinforcement learning to rank with markov decision process. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 945–948, 2017.

Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., et al. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3597–3606, 2020.

Yang, L., Cui, Y., Xuan, Y., Wang, C., Belongie, S., and Estrin, D. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 279–287, 2018.

Yao, S., Halpern, Y., Thain, N., Wang, X., Lee, K., Prost, F., Chi, E. H., Chen, J., and Beutel, A. Measuring recommender system effects with simulated users. *arXiv preprint arXiv:2101.04526*, 2021.

Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management*, pp. 337–348. Springer, 2008.