Supplier competition on subscription-based platforms in the presence of recommender systems *

Jacopo Castellini
Centre for Competition Policy
University of East Anglia
NR4 7TJ, Norwich, UK
j.castellini@uea.ac.uk

Amelia Fletcher†
Norwich Business School, and
Centre for Competition Policy
University of East Anglia
NR4 7TJ, Norwich, UK
amelia.fletcher@uea.ac.uk

Peter L Ormosi‡
Compass Lexecon, and
Norwich Business School, and
Centre for Competition Policy
University of East Anglia
NR4 7TJ, Norwich, UK
p.ormosi@uea.ac.uk

Rahul Savani
Alan Turing Institute, and
Department of Computer Science
The University of Liverpool
L69 3BX, Liverpool, UK
rahul.savani@liverpool.ac.uk

Abstract

Subscription-based platforms offer consumers access to a large selection of content at a fixed subscription fee. Recommender systems (RS) can help consumers by reducing the size of this choice set by predicting consumers' preferences. However, because the prediction is based on limited information on the consumers and sometimes even on the content, the recommendations are susceptible to biases, a phenomenon widely evidenced in the computer science literature. Intuitively, if these biases systematically favour certain suppliers over others, this could impact competition between suppliers. To study this intuition, we introduce a simple framework of a platform that sells to consumers with quasi-linear utility functions via a recommender system. We find that RS biases lead to more concentrated markets and increased entry barriers even when the platform is not self-preferencing their own products, and users are rational. Limited-attention users can reduce the market concentrating impact of RS biases and harm top-selling products, but the platform can counteract this effect by a choice architecture that gives more prominence to popular items. Self-preferencing does not further increase concentration but it ensures that

*This work has benefited from a UKRI Trustworthy Autonomous Systems (TAS) Pump Priming grant EP/V00784X/1. The UKRI TAS Hub assembles a team from the Universities of Southampton, Nottingham and King's College London. The Hub sits at the centre of the £33M Trustworthy Autonomous Systems Programme, funded by the UKRI Strategic Priorities Fund. The views expressed in this paper are the sole responsibility of the authors and cannot be attributed to the CMA, or Compass Lexecon, or any other parties the authors are affiliated with.

†Non-Executive Director of the UK Competition and Markets Authority (CMA). Professor of Competition Policy at University of East Anglia. Amelia Fletcher's contribution to this paper was funded in part by EPSRC grant EP/T022493/1.

‡Corresponding author. Vice President at Compass Lexecon, Professor of Competition Economics at University of East Anglia

the winners are the products preferred by the platform. Although encouraging more exploration can reduce these market consolidating effects, we show that they also reduce recommendation relevance in the short-run.

Keywords: recommender systems, market concentration, recommendation system biases, self-preferencing, subscription-based platforms

1 Introduction

Online platforms are the virtual marketplace that bring together two sides of online markets, the customers and the suppliers. Some of these platforms use a subscription model, whereby customers can sign up to the platform, and, either for free, or in return for a recurring fee, they have access to all content on the platform with no further charges. Customers on these platforms often face a plethora of choices; music streaming platforms offer access to 100 million songs and podcasts, and YouTube hosts over 800 million videos. Evaluating these choices would amount to enormous search costs. To assist the customer in their choice, platforms deploy recommender systems (RS), which help reduce/eliminate these search costs by recommending items that match the consumer's preferences.

Recommender systems are information filtering mechanisms that are designed to provide recommendations to consumers, especially in contexts where the consumer faces large choice sets. But platforms face an inevitable information asymmetry problem when trying to make a recommendation. The RS does not know the consumers' true tastes and therefore has to predict their preferences using data such as previous user interactions. To do this, the RS relies on data that is available to the platform. This could include data on the characteristics of products sold on a platform, or data on how consumers engage (e.g. ignore, click-through, purchase) with products on the platform. The RS is trained on the available data, and the trained model is used to predict the user's preferences. The RS then recommends a product or a set of products that it predicted to be closest to the consumer's preferences. The consumer may or may not engage with these recommendations. A key feature of RS is that the user's engagement or lack of engagement is logged as further data for the RS, which is then re-trained, in the expectation that with the extra information it is going to make an improved prediction next time (the literature refers to this process as feedback loops).

Because data on the customer is never complete, the prediction of the RS is susceptible to biases.[1] These biases are introduced into the system due to the RS' inability to accurately represent the diversity of users, preferences, and content, and can be enhanced or mitigated by the design of RS models, and the data users feed back into the RS. This can manifest in a variety of ways, from the system's inability to properly account for a user's demographic data to its tendency to prioritize certain types of content over others.

Biased recommendations have existed even in the world before algorithmic recommender systems, for example through the personalised suggestions of a shopkeeper. But in the brick-and-mortar

[1] In line with the computer science literature, by bias we mean recommendations that systematically deviate from the true preferences of users, for example, a RS that is disproportionately more likely to recommend popular items.

world, recommendations are delivered in a fragmented, decentralised way, where biases are less likely to lead to systematic distortions to competition between the suppliers of these products and services. One the other hand, online platforms mediate vast swaths of the digital economy, delivering recommendations to billions of users. Biases in these platforms can have more serious consequences if they systematically favour certain suppliers to the detriment of others.

Much of the literature on RS biases focuses on the impact of these biases on the demand side of platforms (for example their impact on the relevance of recommendations), and much less has been said about their impact on the other side of the market, suppliers. In this paper, we show how these systemic biases in the RS inevitably affect competition between suppliers on the platform even where the platform is fundamentally customer-centric (i.e. not self-preferencing). This is because there is a close link between recommendations and actual engagement. Consumers have a strong propensity to click on recommendations (De los Santos & Koulayev 2017, O'Brien & Keane 2006, Joachims et al. 2005) and to make consumption choices on this basis (Ursu 2018, Ghose et al. 2014, Carare 2012, Aguiar & Waldfogel 2021, Lee & Musolff 2021). Thus, if a RS produces systemically biased recommendations, this would drive demand toward those products that benefit from the bias, and away from those that do not. This will in turn distort competition between the suppliers of those products.

This effect is amplified by feedback loops. If consumers are choosing products from a biased recommendation set, this choice is fed back into the RS as data, which risks making the biases more extreme over time. Moreover, suppliers' long-term incentives are also relevant in that they have less incentive to invest or innovate into products if those products are less likely to be recommended.

We simulate a subscription-based platform, where users have access to all items on the platform (i.e. suppliers only compete on the quality of their items). We design a set of recommender systems, and examine how these impact market concentration, entry barriers, and item homogeneity in a simulated environment. Our versatile simulation setup allows us to compare the impact of a range of different RS with different levels of bias, whilst controlling for consumer-side characteristics (such as rationality or limited attention), and platform types (consumer-centric or self-preferencing), something that would have been prohibitively costly to do with real life data or field experiments.

Given the increased political and regulatory attention and the potential impact of a newly forming regulatory system for digital markets, the need for evidence on how fundamental components of digital markets, such as recommender systems, impact supplier competition has become even more pronounced. To provide such evidence, this paper makes several contributions.

First, it shows that RS have a tendency to concentrate the supplier side of the market on subscription-based platforms. The market consolidating impact is directly proportional to the magnitude of RS bias (more biased RS leads to more concentration). This finding is important, because limited competition between suppliers leads to lower consumer welfare (for example through reduced choice). Moreover, as well as harming competition, barriers to entry and expansion risk reducing valuable market dynamism, which has to date been a key positive aspect of the internet (OECD 2021). This could lessen consumer welfare in the longer run.

Second, we provide support to the argument that consumers with limited attention can hurt the most popular products and could reduce the market consolidating effect of RS. This finding, which may seem counter-intuitive at first, holds as long as the popular items are not given disproportionate prominence on the platform.

Third, we show that self-preferencing platforms do not make markets more concentrated than what they would be without self-preferencing, but they determine who the winners are. This carries an important policy message, as it suggests that simply eliminating self-preferencing, which is increasingly being banned around the world, may not make the supply side of the market less concentrated.

Fourth, we show that limiting the likelihood of market-concentrating effects comes at the cost of reducing recommendation quality. This sends an important message to policymakers, as simply requiring the elimination of market-concentrating effects may end up harming customers if the quality of recommendations falls as a result. This reflects a long-term, short-term impact trade-off as well, because recommendation relevance is a fundamentally short-term category but maximising the level of competition between upstream suppliers has much longer-term effects.

The paper starts with an exposition of our simulation framework, which is followed by an introduction of the recommenders used in the paper and the simulation parameters. In the results section we first present results for rational consumers, and then introduce behavioural consumers. We then show how some of the commonly used methods to reduce the supply side impact work in our setup of a subscription-based platform, and present the trade-off between harmful supply-side effects and recommendation quality.

1.1 Relevant literature

Because we study a wide range of research questions, we introduce further relevant literature under later respective sections. In this section, we only explain how our work relates to the more general recommender system literature (focusing on the relevant computer science, management and marketing, and competition policy literature).

There is a rich **computer science** literature on the biases inherent in recommender systems, and Chen et al. (2020) offers a comprehensive review of these works. Most of these papers focus on the demand-side, and look at how the biases impact recommendation quality, and impact on the consumers. On the supply side, the computer science literature discusses the impact of RS on fairness, such as racial or gender fairness. A few papers look at fairness closer to our own research interest, i.e. fair competition. For example, Mehrotra et al. (2018) examined the trade-off between the relevance of the recommendation and fairness in Spotify's RS. Their definition of fairness requires that the content shown to users be spread well across the wide long-tailed popularity spectrum, rather than focusing on a small subset of popular artists. Methodologically they employ a contextual bandit formulation, where they use three different objective functions, one optimising for relevance, one for fairness, and the third one employs a multi-objective function. Ben-Porat & Tennenholtz (2018) investigate RS from a game theoretical perspective, focusing on the fairness of outcomes regarding competition between suppliers.

More relevant to our paper are some of the algorithmic game theoretical approaches to modelling competition between suppliers. Jagadeesan et al. (2022) models the supply side of recommenders with the goal to understand how recommenders shape supply-side competition. They use a high-dimensional producer decision space with heterogeneous users. In their setup, in equilibrium, genres form, where producers specialize to compete for subsets of users, and realise positive profits. In modelling RS with strategic content providers, Ben-Porat & Tennenholtz (2018) propose the Shapley mediator to fulfill fairness (in terms of competition between suppliers) and stability requirements for recommender systems.

Different to these works, we do not formally model the demand and supply sides. This is mainly due to our different research question, which is focused more on studying how a range of recommender designs impact supplier competition under various demand-, and supply-side conditions. For this purpose we preferred a simulation setup with simple but uniformly applicable assumptions about user behaviour.

The study of recommender systems enjoys an equally prominent role in the **management and marketing literature**. Some of these papers focus on the performance of platforms, showing that by enhancing the ability of end-users to choose across a wide range of offerings, RS can facilitate more efficient outcomes, and increase end-user welfare (Zhang 2018, Waldfogel 2017, Zentner et al. 2013, Brynjolfsson et al. 2011). Donnelly et al. (2023) provide evidence that personalised recommendations make consumers search more and generate more purchases relative to uniform bestseller-based rankings. In a similar vein, Lee & Hosanagar (2021) conducted a field experiment on a retail platform to look at the impact of RS on product awareness and conversion rates, and Pathak et al. (2010) and Li et al. (2022) evidence the impact of RS on sales.

Zhou & Zou (2023) look at the impact of the recommender on supplier competition. Their focus is on how sellers strategically choose prices in order to feature in the recommendation set. They also show that excluding prices from the recommendation optimisation can benefit the platform by preventing sellers' race to the bottom competition for recommendations. Similarly, Donnelly et al. (2023) found that online retailers may have incentives to adopt consumer-centric personalisation algorithms as a way to retain consumers and maximise long-term growth. Still studying supplier competition, Li et al. (2020) looks at the advertising and competition effects of recommenders, and analyse the conditions under which suppliers are better/worse off if the platform deploys a recommender system. Li et al. (2018) also presented conditions (the relative size of substitution and demand effects) under which the RS benefits or harms suppliers on a platform.

We complement this body of works by addressing some of the same questions (such as the role of inattentive users, or supplier competition) but we do this in a single framework, that allows comparison across multiple RS designs, and across different demand and supply side assumptions.

Recently more attention has been drawn to the impact of RS on supplier competition from an **antitrust and policy** perspective. Fletcher, Ormosi & Savani (2023) offers a review of the literature on recommender systems, and discusses how these can be linked to core policy problems.

Much of this literature focuses on self-preferencing platforms, which includes empirical works like Cure et al. (2021), Lee & Musolff (2021), or Kotapati et al. (2020), and theoretical analyses, such as Aridor & Gonçalves (2022), Padilla et al. (2022), or De Corniere & Taylor (2014).

Closer to our own paper is Calvano et al. (2021), who, in their default setup, assume that the platform is user-centric (i.e. not self-preferencing). They examine the impact of RS on supplier competition by simulating a collaborative filtering RS to study the price and welfare impact of recommender systems. They find that the RS has a tendency to make markets more concentrated and increase prices. Finally, Fletcher, Ormosi, Savani & Castellini (2023) offers a theoretical approach through a simple theoretical model to demonstrate that RS biases can dampen competition between the suppliers selling through digital platform, arising from the fact that biased recommendations are less closely linked to true preferences.

Our paper contributes to this branch of literature by showing how different RS designs increase market concentration on subscription-based platforms. However, we differ in that we assume a subscription-based platform where suppliers compete on the attributes of their products, but not on price. This allows us to model different RS designs in order to better understand the impact of the specific biases in the recommenders, using the understanding that certain RS designs are more likely to generate biased recommendations. Further, we also allow for different assumptions about the demand side of the market (rational vs behavioural consumers), the platform's objective function (user-centric or self-preferencing), and provide evidence on the recommendation quality - market concentration trade-off.

2 Our simulation framework

In the discussion below we use the shorthand terms *users* to denote the consumers on the platform, and *items* to denote products on the platform. Our thinking on developing this framework and the interaction model was inspired by Chaney et al. (2018) and Schmit & Riquelme (2018).

Assume a platform that sells items to users through a RS. Below we consider a situation where there is a single platform, providing the only means of access for suppliers to end-users. This assumption is for simplicity. In general, the same effects are likely to arise with multiple platforms, so far as they have a degree of "bottleneck" market power over access to end-users. However, the impact of each individual platform on overall competition between suppliers would tend to be reduced with multiple platforms.

Let us denote the total number of users and items as U and I_t respectively, where $t \in 1, 2, \ldots, T$ denotes the time period. Denote the set of users and items as $\mathcal{U}$ and $\mathcal{I}_t$ respectively. The set and number of items are indexed by the time period t because in some configurations we allow items to be introduced over time during a simulation run, which allows us to explore barriers to entry; the set of users is always fixed over all time periods of a simulation run.

Each item $i \in \mathcal{I}_t$ is described by a binary vector y_i with a elements. The kth entry of y_i is 1 if item i has attribute k and is 0 if not. Every item has at least one attribute, and may have up to a attributes. Our focus in this paper is on RS design, rather than modelling supplier behaviour. For this reason item attributes are exogenously chosen, and they cannot be strategically chosen to respond to market outcomes.

All items are priced at zero, therefore items only compete on their attributes. This is equivalent to a subscription-based platform, something that is commonly used for example in content (music or video streaming) or news provision. In this model the platform sets a subscription price to users, and subscribers can access all items on the platform without further charge.

We distinguish between two cases. Where items are horizontally differentiated, the sum of 0/1 values of all attributes is $\sum_{n=1}^{a} y_{in} = C$ for some constant $C < a$ and for all items i. For vertically differentiated items, the quality indicator $\sum_{n=1}^{a} y_{in}$ can differ across the items (an item with more non-zero attributes is ranked vertically higher). In practice, on subscription-based platforms it is unlikely that items are only horizontally differentiated (think for example of a movie streaming platform with different budget films). Under vertical differentiation items with more non-zero attribute values represent higher value items. We assume that all items are loosely competing, i.e. any user could choose any item under the right circumstances, and that users know their own preferences for the attributes.[2]

Each user $u \in \mathcal{U}$ is represented by a (non-binary) vector x_u that describes their true preferences over the item attributes. From y_i and x_u we calculate the true user-item score (or utility from consuming an item) for user u and item i as the dot product:

$$s_{u,i} = x_u^T y_i \ , \tag{1}$$

where we assume that x_u and y_i are column vectors. Throughout the paper we use the terms score and utility synonymously.

To ensure vertical differentiation we set the bounds of preferences asymmetrically around zero, so that users can have stronger positive than negative preferences towards each attribute. In our headline results we use attribute preference score bounds as $[-20, 150]$, i.e. a user's most negative utility regarding an item attribute is -20, and the most positive utility to an attribute is 150.[3]

The advantage of this flexible framework for user utilities is that it can accommodate multiple different recommender systems and therefore it allows us to model and compare the impact of different biases in recommenders. On the flip side, it necessitates important assumptions on our user preferences. The implication of binary item attribute vectors is that they impose linearity on users' preference curves (i.e. constant marginal rate of substitution between two item attributes). This does not imply that users' preferences for items are strictly monotone in the number of item attributes, because preferences for the attributes can be negative. Consider an example with $a = 2$. The possible item attribute vectors are $[1, 0], [0, 1], [1, 1]$. A user with an attribute preference vector

[2]Dzyabura & Hauser (2019) study the case where users learn their attribute preferences through engaging with the RS.

[3]We ensure rationality, by restricting interactions only to cases where the overall utility $s_{u,i}$ for an item i is non-negative.

$[10, -2]$ will prefer item $[1, 0]$ to item $[0, 1]$ but also to item $[1, 1]$.

If search costs were not prohibitive, users could discover the item attribute vectors of all items on the platform. In this case each user u could simply solve the utility maximisation task to select their most favoured item:

$$\arg\max_i \; s_{u,i} = x_u^T y_i \; . \tag{2}$$

We assume that, due to extensive search costs, users do not observe all items on the platform, and instead rely on recommendations, which the platform delivers through its recommender system (RS). The task of the RS is to find the items with the attributes that users would like most.

If the RS had perfect information on the users' preferences and item attributes, it could choose the items that give the highest utility to each user u. But the RS typically does not observe the true user preference x_u and item attributes y_i. Instead, it tries to estimate the utility $s_{u,i}$, either directly, or via estimating y_i and/or x_u, using data on user-item interactions over time. We use the following notation for user-item interactions over time.

Definition 1 (Interaction matrix)**.** *Let* $\mathbf{R}(t)$ *denote the* $|\mathcal{U}| \times |\mathcal{I}_t|$ *non-negative matrix, where the entry for user* u *and item* i*, which we denote by* $r(t)_{u,i}$ *says how many times* u *has interacted item* i *in periods* $1, \ldots, t$*. Typically,* t *will be clear from the context, and we just write* $r_{u,i}$ *for brevity.*

The RS provides a prediction score $\hat{s}_{u,i}(t)$ at time t of how much user u will enjoy item i (the predicted utility the user would get out of consuming item i). We assume that the RS is user-centric, in that it aims to maximise user satisfaction, rather than some other objective (as would be the case for example for a self-preferencing platform). Denote by $\mathcal{L}_u(t)$ the set of recommendations made to user u at timestep t. For each user u at time t the RS picks k items with the largest predicted scores. That is, the items are sorted in decreasing order according to their predicted scores, and the first k items in this sorted list are recommended. Formally, for each user u and timestep t, the RS orders (i.e., sorts) their items by their predicted scores (breaking ties at random) to give a sequence of items:

$$\{\omega(i)\}_{i=1}^{I_t} \quad \text{such that} \quad \hat{s}_{u,\omega(1)}(t) \geq \hat{s}_{u,\omega(2)}(t) \geq \ldots \geq \hat{s}_{u,\omega(I_t)}(t). \tag{3}$$

Then the RS recommends the items $\mathcal{L}_u(t) = \{\omega(1), \omega(2), \ldots, \omega(k)\}$ to that user.

When presented with this list, the user learns the attribute vectors of the recommended items, and solves the utility maximisation problem in Eq.(2), i.e. the user calculates the true $s_{u,i}(t)$ for each item $i \in \mathcal{L}_u(t)$ recommended at time t and picks an item i^* such that

$$s_{u,i^*}(t) = \max_{i \in \mathcal{L}_u(t)} s_{u,i}(t) \; .$$

This assumes that users are rational.[4] Later we will relax on this rationality assumption.

[4] Other papers that assume rational consumers are Chaney et al. (2018), Abdollahpouri et al. (2019), Kotkov et al. (2016).

This design of users being aware of a recommendation set, and choosing a specific item from this set allows us to distinguish between awareness and consideration as we can control for the items that the user sees (which was a limitation in Helmers et al. (2019)).

3 Simulation details

Our simulations build on Lucherini et al. (2021), which we adjusted to fit our specific problem.[5] The code to replicate our experiments can be found at: [*removed for double-blind review*]

3.1 RS design

For the problem of estimating $\hat{s}_{u,i}$, we use different RS designs that exhibit the features of the RS that are commonly used in industry.Our RS do not attempt to mimic the complexity of RS deployed in real markets. However we believe that these models are adequate to demonstrate the biases that are inherent in real-life recommenders, and to test the conditions under which upstream competition between suppliers is hindered as a result of these biases.

Below we briefly explain how each of our RS designs estimates the user-item utilities. The first three of these are very simple but are useful as benchmarks. Whilst we present results for all five RS, our focus will be on content-based and collaborative filtering (and to some extent popularity RS) as the most commonly used RS in practice. Two of the RS are non-personalized (the Random and Popularity RS), and the other three (the Ideal, Collaborative Filtering, and Content-based RS) give personalized, user-specific utilities for a given item.

3.1.1 Ideal RS

The ideal RS is a hypothetical benchmark case, where we assume that the RS is fully aware of $s_{u,i} = x_u^T y_i$. The ideal RS is able to find a best match for each user, and recommend this best match at all timesteps. Similar users are more likely to be recommended similar items than dissimilar ones.

3.1.2 Random RS

Another benchmark is the random RS, which considers the full item set, and randomly chooses k items at each timestep t. The random RS ignores user preferences and item characteristics and

[5] The main changes to the original `t-recs` package were: added an ideal recommender, allowed for negative attribute utilities for users, constrained users to not interact with items that have a net utility, changed the estimation methods of the RSs in `t-recs` (for example we fixed random recommenders to avoid duplicates, replaced non-linear least squares in the content-based RS with least squares, and made the latent vector space in collaborative filtering more flexible - these are explained in detail below), added the crucial outcome metrics (such as market concentration, entry, or recommendation heterogeneity), added vertical and horizontal product differentiation, added a way to measure self-preferencing, added the possibility of an RS with passive or active online choice architecture, added hybrid recommenders which combine each RS method with a random recommender, added two specific hybrid RS (ensemble and mixed hybrid), added measures of recommendation quality, added a way to force the RS to always include new items in the recommendation sets, fixed the way the code handles drifting user preferences.

as such is likely to give poor recommendations. Under a random RS, users are likely to be shown different items at each timestep.

3.1.3 Popularity RS

Whilst the ideal and random RS are hypothetical benchmarks, popularity recommenders are frequently used by platforms. Retailers displaying best sellers at the top of search lists, content streaming platforms listing the most viewed or listened to content, or news platforms listing the top read articles are just a few examples. It is rare that a platform would only rely on a popularity RS, yet popularity recommendations are likely to be prominently displayed on a platform, and consumers are likely to engage with them.

To approximate the predicted user-item score, a single set of scores for items is produced and used for all users. For a given item i, the score is simply the total numbers of interactions with the given item across all users until this point in time. Formally, in terms of $\mathbf{R}(t)$:

$$\hat{s}_i(t) = \sum_{u \in users} r_{u,i} \ . \tag{4}$$

In other words, the score is the sum of the entries of the respective item column of $\mathbf{R}(t)$. Intuitively, the popularity RS will recommend the item most users chose in the first few timesteps to every user for the rest of the experiment.

3.1.4 Collaborative filtering (CF)

Collaborative filtering RS are very common on all types of platforms, but are probably most prominently known for their application in movie streaming recommendations (Bennett et al. 2007). The idea behind CF models is that people with similar tastes similarly value items on a platform. If there are two users, and both users liked item A, and one of the users also liked item B, then it makes sense to recommend item B to the other user as well. Most CF models are based on either a nearest neighbour method to calculate similarities between users or items, or matrix factorisation. Although CF models provide a scalable tool for RS applications, biases in the predictions of these models, such as popularity bias, or the cold start problem, are well known, and widely documented in the computer science literature (Yao & Huang 2017, Sun et al. 2019, Bobadilla et al. 2012, Guo et al. 2014).

We use a matrix factorisation method for our collaborative filtering RS, based on the LensKit Python framework for recommender systems (Ekstrand 2020). The basic idea of matrix factorization, also known as matrix decomposition, is to write a matrix as the product of two, lower-dimensional matrices. In our context, the matrix that we decompose encodes the interaction count between a particular user u and a particular item i as the product of two vectors, one which describes the user, and one which describes the item (described formally below).

As such, this is called *implicit* matrix factorization within the RS literature (including e.g. in LensKit and (Hu et al. 2008), because in this setup we are not decomposing explicitly requested ratings or preferences of the users, but rather implicit preferences derived from the interactions only.

Within LensKit, implicit matrix factorization is based on a line of work that originated with Hu et al. (2008), who introduced the basic setup, a corresponding model objective function, and an alternating least squares approach to train a model to optimize such objective function. Hu et al. (2008) decomposed a binary matrix that encoded whether an item had been interacted with or not, using the number of interactions to derive a weight for the reconstruction error of matrix entries. We deviate from this approach and use a simpler default setting in LensKit, which better matches what we do below for content-based filtering. This allows us to make more relevant and fairer comparisons across collaborative filtering and content-based RS.

In particular, we directly decompose a matrix of interaction counts, with all entries weighted equally in the objective function. Here we give an overview of the setup and model training; further details can be found in the LensKit documentation (`https://lenskit.org/`), along with the papers of Hu et al. (2008) and Takács et al. (2011).

Formally, our implicit matrix factorization decomposes $\mathbf{R}$, the matrix of interaction counts, so that the entry $r_{u,i}$ (the number of times user u has interacted with item i) is estimated by the dot product of a user vector $\bar{x}_u$ and an item vector $\bar{y}_i$; we denote this estimate by $\bar{r}_{u,i}$, which can then be written as:

$$\hat{s}_{u,i} = \bar{r}_{u,i} = \bar{x}_u^T \bar{y}_i \ .$$

Note that the same item representation is used with all the different users (i.e., for all i, the fitted $\bar{y}_i$ depends only on i and not u). Overall, the goal of the optimization is to find the item and user vectors so that the estimates $\bar{r}_{u,i}$ are good in aggregate, across all users. The aggregation across users and items is done according to the resulting optimization function (5). In this function, the quality of the reconstruction in terms of errors, as we have just been describing, is captured by the term $\sum_{u,i}(r_{u,i} - \bar{x}_u^T \bar{y}_i)^2$, which simply sums up all the squared reconstruction errors $(r_{u,i} - \bar{r}_{u,i})^2$ across all users u and items i. The reason to square the errors, as is standard (and called the Mean Squared Error when one also divides by the number of summands), is to ensure that positive and negative errors do not cancel each other.

$$\min_{\bar{x},\bar{y}} \sum_{u,i}(r_{u,i} - \bar{x}_u^T \bar{y}_i)^2 + \lambda(\sum_u \|\bar{x}_u\|^2 + \sum_i \|\bar{y}_i\|^2) \ . \tag{5}$$

Note that the dimensionality of all $\bar{x}_u$ and $\bar{y}_i$ (which are equal) need not be the same as the true number of attributes a (i.e., the length of the true x_u, y_i, which are unknown to the RS, do not need to be the same as of $\bar{x}_u$ and $\bar{y}_i$). The chosen dimensionality of $\bar{x}_u$ and $\bar{y}_i$ is known as the "number of features", or the dimensionality of the resulting model's *latent space*. This is one of most important hyperparameters of this model. While not necessary, we intentionally chose it *to be equal to the number of true attributes*, a. The reason for this choice is that it ensures that the model is, in principle, rich enough to capture the true preferences. This choice for the collaborative filtering model also makes it more fairly comparable with the content-based model described below, in which, the features are taken to be the true attributes themselves (which are assumed to be known by the platform).

The final term in (5), namely $\lambda(\sum_u \|\bar{x}_u\|^2 + \sum_i \|\bar{y}_i\|^2)$, is a regularisation term to avoid overfitting. It ensures that the Euclidean norms of the latent vectors $\bar{x}_u$ and $\bar{y}_i$ will stay relatively low (allowing them to grow arbitrarily large would allow a good fit on training data, but often bad generalization to unseen data). For the regularisation weight parameter, λ, we use the default value in LensKit.

Having defined the training objective, we finally describe how training, i.e., finding good latent vectors that are close to optimal for (5), is done. To find $\bar{x}_u$ and $\bar{y}_i$, we use a standard approach called alternating least squares (ALS), in which we alternate between re-computing user factors and item factors, and each step lowers the value of the cost function. Hu et al. (2008) introduced this approach for implicit matrix factoriztion, using LU (lower–upper) decomposition; we use an improvement introduced in Takács et al. (2011) who also use ALS but using Conjugate Gradient Descent as opposed to LU decomposition.

Once the model is trained, it can give an estimated score for any user and item that was present during training. For example, these scores can be used to simply rank the items for a given user, as a prediction of how that user would rank these items.

As part of our experiments we also introduce new items (to explore barriers to entry, and supply-side competition). These items are not in the training of the implicit matrix factorization, so we need a method to produce an initial item representation $\bar{y}_i$ the first time that i is available for recommendation; after that, in subsequent periods, i is present during the new training of the model. The initial $\bar{y}_i$ for a new item is *imputed* as the average item vector of all existing items on which the model was trained, i.e., it treats the new item as if it where an "average" item in feature space. This way we aim to give some chance that a new item is actually recommended when it is first available.

3.1.5 Content-based filtering (CB)

Content-based filtering is another canonical recommender method used by platforms. CB recommenders rely on item features to recommend other items similar to what the user had liked before. If a user likes item A, and item B has similar features to item A, it makes sense to also recommend item B to the same user. Because CB recommenders consider item features, they are much better at eliminating certain biases, for example the cold start problem, which CF recommenders are less able to deal with. However, systematic biases are still widely documented (Nguyen et al. 2014, O'Callaghan et al. 2015, Nagulendra & Vassileva 2014, Pagano et al. 2016).

Under content based filtering, the RS is assumed to know the true item attributes y_i. The RS learns user-attribute preferences $\hat{x}_u$ from the user-item interactions. To do this, the RS observes the interaction matrix $\mathbf{R}(t)$, and then, for each user independently, regresses the interaction counts for that user on the binary attribute vectors of the corresponding items with Ordinary Least Squares. To contrast CF and CB: while CF learns a representation of each item and each user with a single model based on the matrix $\mathbf{R}(t)$, CB learns a different model for each user u, where each such model has the regression targets as the interactions described by a single row of $\mathbf{R}(t)$ for u, with the true item attributes taken as known. Section A in the Online Appendix provides a simplified example to help understand our CB model.

3.2 Simulation parameters

Our simulations are based on 100 users, and 200 initial (incumbent) items (all items are independent, as if there were 200 different suppliers on the platform). Each item has a attributes, and each user has a preference for each of these attributes. We create the 200 incumbent items with random attributes. Each true user vector x_u is sampled from a multivariate uniform distribution $x_u \sim \mathcal{U}(-M_1, M_2)$, where $M_1 = -20$ and $M_2 = 150$ are the minimum and maximum value for a user attribute respectively. Similarly, each true item vector y_i is sampled from a multivariate binomial distribution $y_i \sim \mathcal{B}(1, 0.5)$. We run our simulations over 100 timesteps, where users are offered recommendations at each timestep, the user chooses to interact with one of the recommended items, and the RS is retrained based on the interaction of users.

It is additionally assumed that at each timestep each user u interacts with item i at most once, but more than one user can interact with the same item. We assume that users' choice is based purely on the optimisation problem in Eq.(2), and the ranking of the items does not affect the user in their choice of the item to consume (later we relax on this assumption and will allow for behavioural users).

In our framework, we allow for the possibility that the RS pre-learns consumer preferences, by offering random recommendations over a number of training periods T', and logging interactions with these recommendations before the experiment starts (this may be seen as equivalent to a platform doing market research). If there were no such training period, then the RS would no no information on the users at timestep $t = 1$ and therefore the very first iteration in the simulation would be a random recommendation, irrespective of our choice of RS. In our experiments, we consider two values of the parameter T' (called *training* in the code-base and Table A1), 10 and 1000, which should be contrasted with $T = 100$, which is the number of time periods in the main simulation after training.

Over the simulations, we allow the entrance of new items. The attribute vectors of these new items are also exogenously chosen, i.e. the creators of these items do not strategically choose their attributes based on previous recommendations and interactions. Incumbent items correspond to $\mathcal{I}_1$, i.e., the items that are present in the first period, and, importantly, were all present during training (which happens before period 1). Then from period 1 until T a fixed number of new items is added in every period. This number, ν, which is called *num_ creators* in the code-base, is always 10 in our experiments (see Table A1). We denote the set of new entrant items at timestep t, i.e., items introduced after the first period up to and including period t, as $\mathcal{I}_t^e = \mathcal{I}_t$- $\mathcal{I}_1$. There will be $\nu \times t = 10 \times t$ such items in period t.

We control for parameters such as the number of new items entering the market at every timestep, the number of pre-startup training periods, and the number of attributes each item has. Details on the values of these parameters are given in Section 9. For each parameter configuration we do 30 runs. The users and items vectors are re-sampled after each run.

4 The impact of RS on upstream competition

We focus on three different metrics for approximating market outcome[6]. For these metrics, we use the following notation: For $i \in \mathcal{I}_t$, let $\mathbb{1}_{i=c_u(t)}$ be 1 if i was the item that user u interacted with in period t and 0 otherwise.

The platform defines the objective function of the RS, which may be to maximise consumer satisfaction or the relevance of the recommendation. As a starting point, we assume that the incentives of the platform and its users are broadly aligned, in the sense that the RS are intended to be as user-centric as possible (i.e. maximising the relevance of the recommendations for the users).

4.1 Barriers to entry and expansion

Fletcher, Ormosi & Savani (2023) explains the intuitive link between RS biases, such as popularity bias, or homogeneity bias, and market entry. One of the concerns arising from RS biases is that they may create barriers to entry and expansion, which in turn could make/keep markets more concentrated. If popular or well-established items feature disproportionately in recommendations, it will be difficult for smaller or new entrants to be recommended. This is amplified by feedback loops, as the RS is more likely to collect data on interactions with popular and incumbent items as opposed to new entrants (the 'cold start problem'), causing a relative lack of data on new or smaller items (new entrants). Without data it is more difficult for new items to establish themselves in the market.

To measure market entry, we define the market share of new entrant items. We consider this market share over the last 10 periods, i.e., for period t, we consider the interval $[t-9, t]$. Denote the market share of new entrant items within this interval as $MS^e(t)$, where:

$$MS^e(t) = \frac{1}{10 \cdot U} \sum_{\substack{e \in \mathcal{I}_t^e,\ u \in \mathcal{U} \\ \tau = t, \ldots t-9}} \mathbb{1}_{e=c(u,\tau)} \ . \tag{6}$$

Recall that U is the (constant) total number of users, so the denominator $10 \cdot U$ is the total number of interactions over the 10 periods. A lower value of MS^e implies that new items are less likely to be recommended by the RS and in turn be selected by users.

Figure 1 plots MS^e for the 5 recommenders, and offers some illustration of the tendency of RS to make entry more difficult for new items. Although our model depicts a subscription-based platform, and it is very unlikely that such a platform would only host content that is not vertically differentiated, for expositional purposes the figure distinguishes between two cases (horizontal, and horizontal + vertical product differentiation).

The ideal recommender provides a useful benchmark to interpret Figure 1. Under this setup the RS knows the users' preferences and therefore there are no RS biases. If an item is more popular than the others, it is because it is valued higher by users. Figure 1 shows that MS^e reaches around 0.75 under the ideal RS. In comparison, the content based RS does not get to this level of entry even under the horizontal differentiation setup. In the more realistic vertical differentiation case,

[6]See Fletcher, Ormosi & Savani (2023) for a motivation of these metrics.

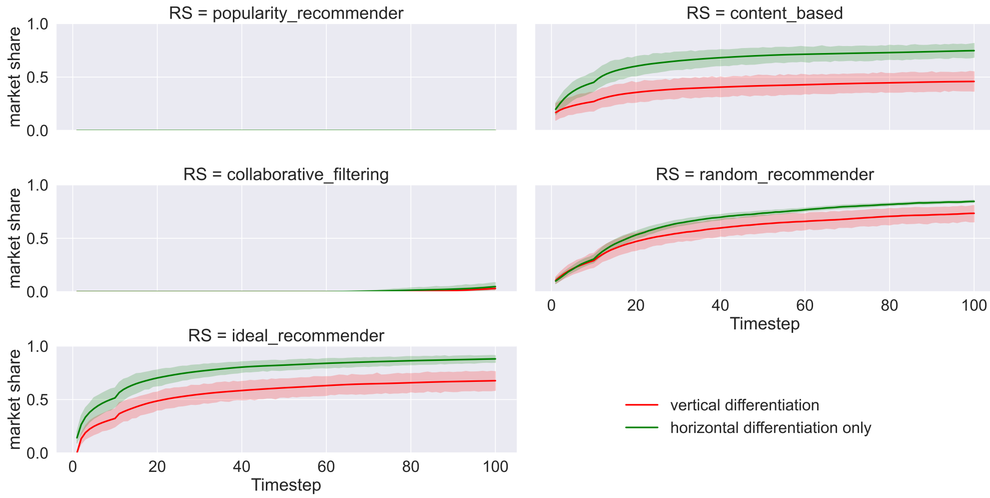

Figure 1: Market share of new entrant items.

MS^e only increases to around 0.5 entry levels. Under collaborative filtering (and the popularity recommender) MS^e remains zero, implying that it is very difficult (almost impossible) for new entrants to gain market shares if items are vertically differentiated. This cold start problem is to do with how CF works, i.e. that it only uses data on users' prior interactions with items rather than item attributes. If new items enter the market, the CF recommender does not know how much the users would like these new items. The cold start problem is also a form of incumbency bias, whereby incumbent products are recommended simply because the RS has more data on them, not because they are the best. Under these conditions new products are not recommended, even if they fit better with end-users' underlying preferences.

4.2 Increased market concentration

If entry is more difficult for new items, especially if these effects are further amplified by feedback loops, market concentration is likely to increase. We measure market concentration by the market share of the 1% of "largest", i.e., most interacted with, items. Similarly to the market share of new entrants, at each time step t we consider the market share over the preceding 10 periods, i.e., for period t, we consider the interval $[t-9, t]$. Denote the market share of item i within this interval as $MS^i(t)$, where:

$$MS^i(t) = \frac{1}{10 \cdot U} \sum_{\substack{u \in \mathcal{U} \\ \tau = t, \ldots t-9}} \mathbb{1}_{i = c_u(\tau)}. \tag{7}$$

Then, using the market share for item i that we have just defined we define the following *concentration index* as the combined market share of the 1% of largest items (i.e., those with the highest market shares):

$$CI(t) = \max_{\substack{X \subset \mathcal{I}_t \\ |X| = 0.01 \cdot I_t}} \sum_{i \in X} MS^i(t). \tag{8}$$

A high value of $CI(t)$ means that a small percentage of the total number of items is being consistently selected by the users, either because these are the more satisfying according to their preference or because of inherent recommendation biases from the recommender system itself.

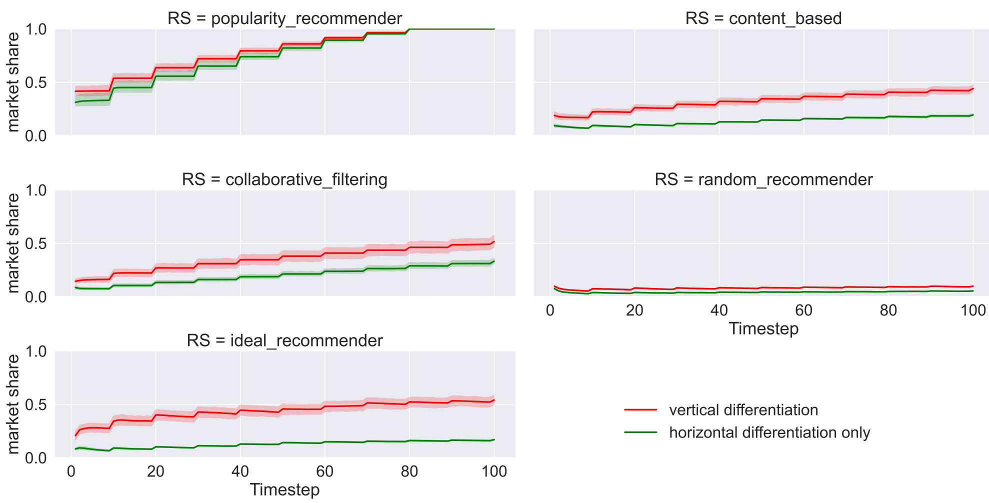

Figure 2: Market share of 1% most popular items.

Similarly to above, the ideal RS provides, as a benchmark, a recommender that is aware of the users' preferences, i.e. where there is no RS bias. Under the ideal RS, the top 1% of items get all the market under vertical and around 2/3 of the market under horizontal differentiation. We know from Figure 1 that new entrants can also gain market share and become part of the top 1%, because as new items enter the market, new high-quality items also enter. The level of concentration stays just under half of the market shares at the final iteration.

It is worth noting that market concentration increases even under horizontal differentiation. The reason is that we draw user preferences randomly, and therefore, although each item has the same number of attributes, some attributes are preferred more than others under any given draw of user preferences. Items with these attributes will become more popular. The concentration under horizontal differentiation increases because new items enter the market, some of which will have attributes that many of our particular randomly drawn users like.

Against this counterfactual, it stands out that under both collaborative filtering and content based RS, market concentration is similar to what would prevail if the RS knew the users preferences (the ideal RS). The reason for this is that high-quality new entrants can gain high interaction-share under an ideal RS, and this keeps markets more concentrated. The difference is that under the ideal RS, the high concentration is achieved by both incumbent and new entrant items, whereas under popularity, and collaborative filtering RS, the high concentration is associated with incumbent items. So RS biases do not seem to make markets more concentrated, but they strongly impact who the winners are.

4.3 How much of these effects are due to RS biases?

The concentration increasing impact of recommenders are driven by the inherent characteristics of the RS, but it is difficult to pin down causality between the specific types of RS bias and the impact on supply-side competition. However, a consistent pattern emerges from the experiments above. Firstly, the negative impact on upstream competition is most enhanced in popularity recommenders, which are most likely to suffer from popularity bias.

When comparing collaborative filtering with content based filtering, the latter leads to a lower reduction in the intensity of supplier competition. Collaborative filtering is also more likely to suffer from popularity bias and incumbency bias (for example because it is more susceptible to the cold start problem), which would be consistent with our Hypothesis 1, that increased RS biases increase market concentration and barriers to entry, and Hypothesis 2, that increased biases lead to more homogeneous recommendations. These suggest that although the choice of RS design may seem like a technical question, it contributes directly to the type and extent of any market concentrating impact of the RS.

5 Users with limited attention

In the above discussion our focus was on users that can evaluate the recommendation set and choose the item that maximises her own utility. In the real life context of recommenders this assumption is unlikely to hold, therefore we relax the rationality assumption to allow for users that simplify complex decisions.

Even when shown a limited recommendation set, users may not be able to make a fully informed choice. For example, there is ample cognitive science evidence on users exhibiting behavioural biases such as position bias (where users are more likely to engage with items listed at the top), or salience bias (where users are more likely to engage with more prominently displayed items). These decision biases can feed into the recommender system through feedback loops and can impact competition between suppliers on the platform (Zhou et al. 2023). Some of the previous works on RS assume behavioural users with attributes such as decoy, position bias, and risk aversion in Teppan & Zanker (2015), the preference of using more positive ratings in Wang & Chen (2021), or position bias in Guo et al. (2014) and Long & Liu (2023).

Here we look at one example of cognitive biases, limited attention. In our framework, we design limited attention as users who are more likely to chose the top ranked one of the items presented to them, because of position bias, irrespective of the fit or previous popularity of the recommended items. If the top item happens to be different from the most popular item, this could limit the market concentrating impact of popularity bias. For example, Helmers et al. (2019) show that having users with limited attention can harm top-selling products if all products are equally highlighted, simply because in this case the popular products are less clearly prominent and are therefore less likely to be considered. If, however, the most popular items are also most likely to be listed at the top of a list of recommendations and these items receive disproportionately more attention and user interaction (position bias), this would generate feedback data that further supports an item's popularity, which will in turn increase the drive towards market concentration.

To introduce users with limited attention, we modify the utility maximisation problem the user faces in Eq.2, by multiplying the true user-item score of the recommended items by a power of the recommendation ranking, i.e. position, of the given item, $rank_{u,t}(i)$, which is an integer $1, \ldots, k$ which says at which position within the k recommended items the item i is shown as (with rank 1 being the most prominent item, and k being the least prominent). Algebraically, Eq.(2) becomes:

$$c_u(t) = \underset{i \in \mathcal{L}_u(t)}{\arg\max} \quad s_{u,i}(t) \cdot (rank_{u,t}(i))^{\theta} \ . \tag{9}$$

The case we have dealt with so far, with rational users, corresponds to $\theta = 0$, so that $rank_{u,t}(i)^{\theta}$ for any rank, and the users just use the true score $s_{u,i}(t)$ alone to make their decision. Users with limited attention that favour the top ranked items correspond to negative θ values. We use the value $\theta = -1$, which turns (9) into:

$$c_u(t) = \underset{i \in \mathcal{L}_u(t)}{\arg\max} \frac{s_{u,i}(t)}{rank_{u,t}(i)} \ . \tag{10}$$

In short, the user still chooses the item with the highest (modulated) user-item score, where the scores are modulated by dividing the original true score by the rank (an integer between 1 and k), so the highest ranked item (i.e. the item in the top position) has its score unaffected; the item in second position has its original score halved, and so on, with the lowest position item's original score divided by k. We will refer to these users that make their decisions based on (10) as *limited-attention users* in the rest of the paper.

Inspired by the findings in Helmers et al. (2019) we distinguish between two platforms based on the choice architecture they employ. One presents the recommendations for each user in descending order of the predicted user-item score as shown in Eq.(3) (we call this a *passive online choice architecture*, OCA). The other platform sorts their items by their predicted scores (same as in Eq.(3)) in order to first pick the set of k items but then presents them in the order of their popularity (i.e. the total number of previous interactions across time periods and users), so the most popular item is in rank 1, the second-most popular in rank 2 and so on (*active OCA*).

Figure 3 shows that limited-attention users (with a passive OCA platform) lower supplier concentration under our two main RS (CB and CF). When the users just take the highest ranked recommendations, the RS effectively exercises more exploration, as users interact with items that they otherwise would not. Although the finding that users with limited attention dampen the market-consolidating impact of RS may seem counter-intuitive, our explanation is similar to that in De Clippel et al. (2014), who find that consumers with limited attention are more likely to miss the best offers, but limited attention introduces a new dimension of cross-market competition as behavioural users engage with items that rational users would not have engaged with.

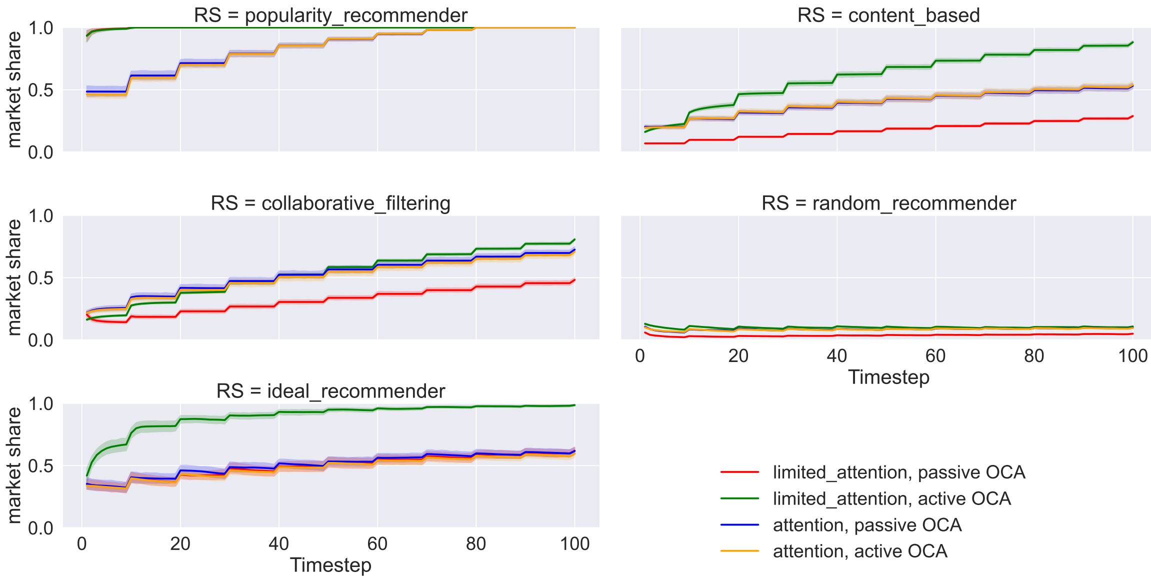

Figure 3: The impact of limited-attention users - Market share of 1% most popular items.

If however the platform uses an active OCA and displays the most popular items on the top, users with limited attention tend to choose more popular items, and all recommender systems (except the random one, which randomly selects the k recommendation set) start to behave more like the popularity RS.

Figure 4 shows that entry becomes more difficult with limited-attention users under content-based filtering (in comparison to rational users). Under collaborative filtering, entry is almost impossible both under rational and limited-attention users. One reason is that new items are less likely to be shown at the top of recommendation sets, and are therefore less likely to be picked by limited-attention users.

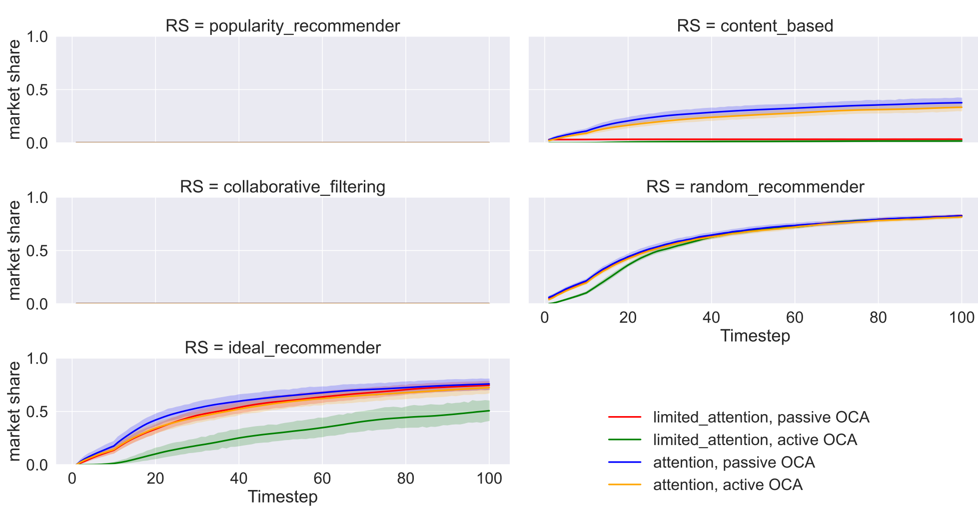

Figure 4: The impact of limited-attention users - Market share of new entrant items.

Our results highlight an important point. Behavioural biases can mitigate or conflate the market consolidating impact of recommenders, depending on the choice architecture employed by the online platform. This central role of the OCA is highlighted in the Google Shopping case, where the European Commission found that consumers click far more often on items that are more visible (the 10 highest ranking items received approximately 95% of all click-throughs).[7]

6 Self-preferencing platforms

The concerns raised in this paper so far are not associated with platforms acting strategically to limit or distort competition. This has been more than a hypothesising assumption. After all, if the RS delivers more relevant recommendations, end-users may be expected to buy more items, which will in turn generate more profit. A RS that seeks to maximise platform revenue need not harm other objectives, such as end-user satisfaction (Azaria et al. 2013). This is more likely to be the case if there is competition between platforms. Likewise, if an RS can generate more competition between upstream suppliers, then prices will tend to be lower on the platform, again attracting more end-users.

However, this coincidence of interests is not necessarily always true. There are reasons to believe that a platform's interests may not be entirely in line with those of end-users. First, the platform may be in a position to profit from restricted supplier competition, for example by charging a fee that extracts a share of the resulting rents (Zhang et al. 2021). Second, the platform may favour those suppliers from which it can extract a higher margin (Bourreau & Gaudin 2021, Hunold et al. 2020). Third, the platform may be vertically integrated into supply. Finally, some larger suppliers may have bargaining power with respect to the platform and may be able to impose contracts that influence RS recommendations (Antal et al. 2021). These platforms may have an incentive to tweak their RS to favour the product that maximises the platform's revenue, creating a 'self-preferencing bias'.

Self-preferencing by large online platforms has been at the centre of policy attention. It is now banned for gatekeepers (the European Commission designated the largest online platforms as gatekeepers), and is dealt with by Regulation (EU) 2022/1925 (Digital Markets Act), which imposes that "The gatekeeper shall not treat more favourably, in ranking and related indexing and crawling, services and products offered by the gatekeeper itself than similar services or products of a third party. The gatekeeper shall apply transparent, fair and non-discriminatory conditions to such ranking."[8]

We introduce a self-preferencing platform that favours certain items (either because it is their own vertically integrated product, or because they receive higher commission from these products, or because these products have a superior bargaining position vis-a-vis the platform). For this, we assume that the platform favours items that are likely to be among the better ones. An example of this would be a music streaming platform where major record labels, which typically represent

[7]European Commission (EC). (2017). CASE AT.39740 Google Search (Shopping).
[8]Article 6(5) of the Digital Markets Act.

the most popular artists, have superior bargaining position to acquire disproportionate exposure for their own artists.

Our goal is to pick a randomly chosen subset of b reasonably good items that will be "self-preferenced". In our experiments b was set as 1, but the method we used would apply for larger b too. First we form a Boltzmann distribution with the probability of item i being proportional to its quality as measured by the number of non-zero attributes of items, i.e., $q(i) := \sum_{n=1}^{a} y_{in}$. Formally the probability of item i is:

$$\frac{1}{Z} \exp\left(q(i)\right) \ ,$$

where Z is the normalizing constant $\sum_i \exp\left(q(i)\right)$. For each simulation run with self-preferencing, at the start we sample with replacement b items according to the distribution above, computed for the particular random draw of items. The resulting set of $b = 1$ items are then forced to be shown to every user in the first τ periods (with the remaining items in their recommendation set chosen as in the usual case of no self-preferencing).[9] An equivalent of this would be a streaming platform, that features songs on popular playlists that it receives higher commission for (or songs that are represented by labels that have stronger bargaining power).

Figure 5 shows that even when users are rational, and the platform gives preferential treatment to only 1 item, and only for 10 time periods (blue line), under the popularity, the content-based, and the collaborative filtering RS this one item holds on to around a fifth of all interactions even after the preferential treatment has stopped (with a moderate drop under CB). If users have limited attention, and the platform designs the choice architecture to give prominence to the preferred item (red line), then the preferred item will have all interactions under the popularity the content-based, and collaborative filtering RS. By definition the market share of preferred items is zero if there is no self-preferencing (green line).

Self-preferencing is likely to also affect the market shares of other items, not just the preferred one, because it might trigger a replacement effect (the preferred item replaces the non-preferred ones in having the largest market share), and/or a market-consolidating effect (self-preferencing makes the whole market more concentrated).

Figure 6 shows that market share of the top 1% of items is around the same under two scenarios: where there is no-self-preferencing (green line) and if the platform is self-preferencing but users are rational (blue line). If users have limited attention, and the platform is giving prominence to its preferred items (red line) the market becomes more concentrated (fully concentrated under all but the random recommender).

Figure 7 shows that under popularity, and collaborative filtering RS, new entrants cannot gain any market shares if the platform is self preferencing and it applies an active online choice architecture and users have limited attention. Once again, with rational users entry barriers seem no different from a no self preferencing scenario.

In interpreting these findings, we must note how the mechanism behind self-preferencing works in our simulation setup. Because self-preferencing ensures that certain items receive disproportionate

[9] The parameter b is is known as *num_forced_items* in Table A1.

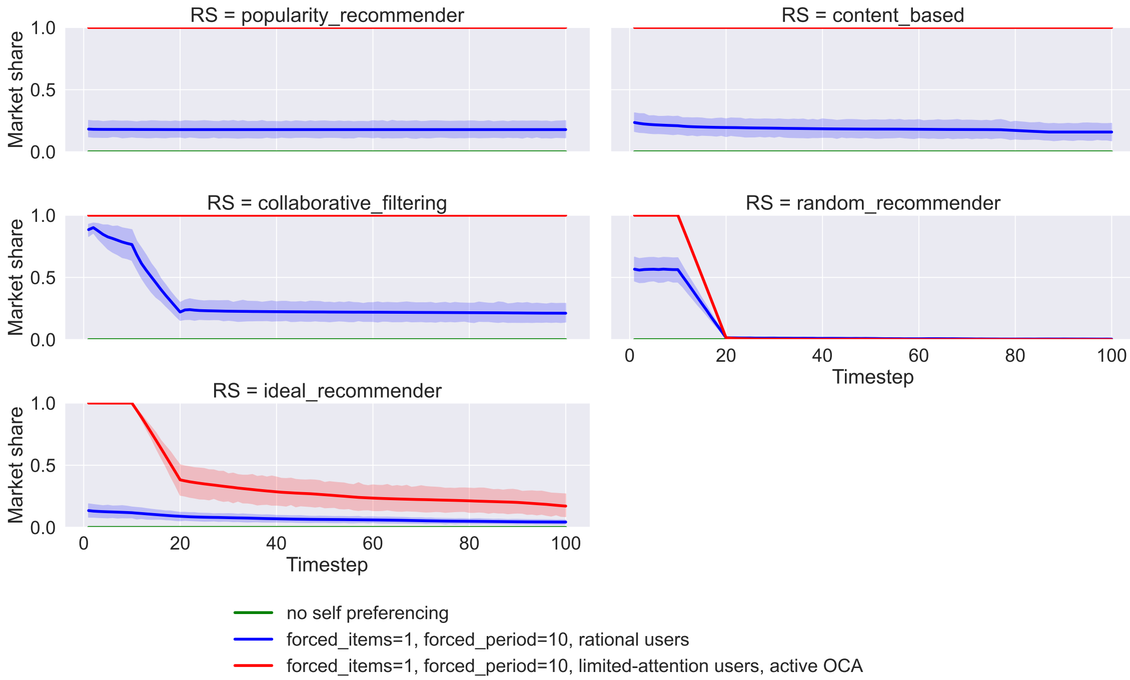

Figure 5: Market share of the preferred item(s) under a self preferencing platform.

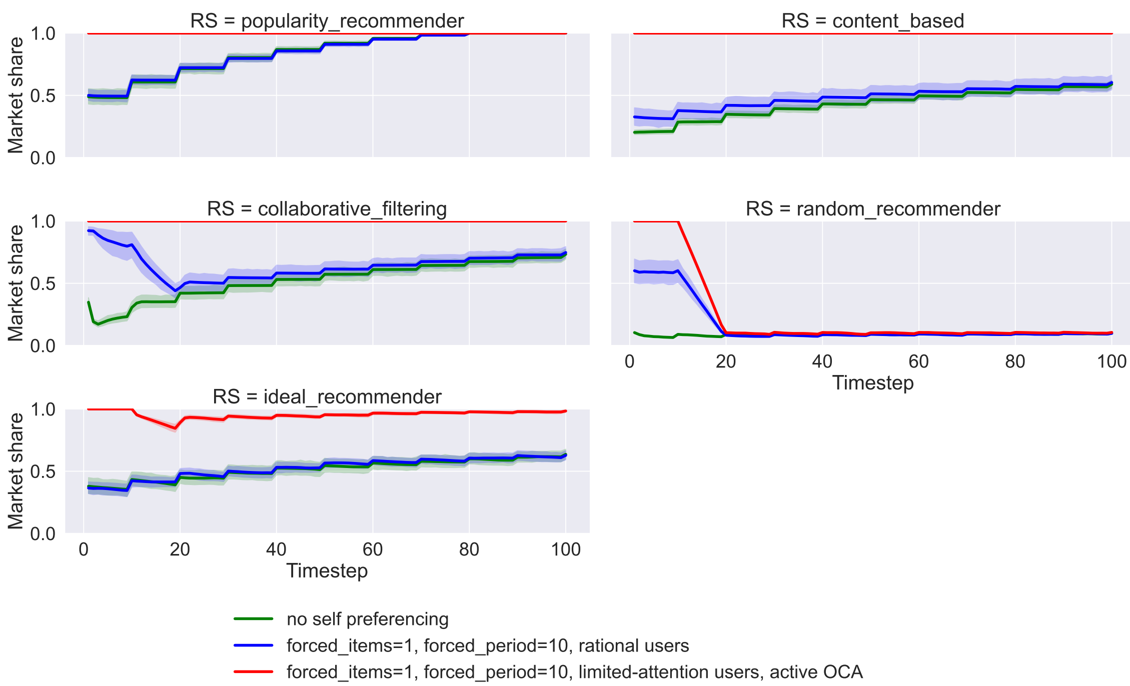

Figure 6: Market share of the most popular 1% of items under a self preferencing platform.

attention, it increases the likelihood that users interact with these items. Feedback loops then make sure that this increased interaction likelihood becomes hard-wired in the RS, even after the

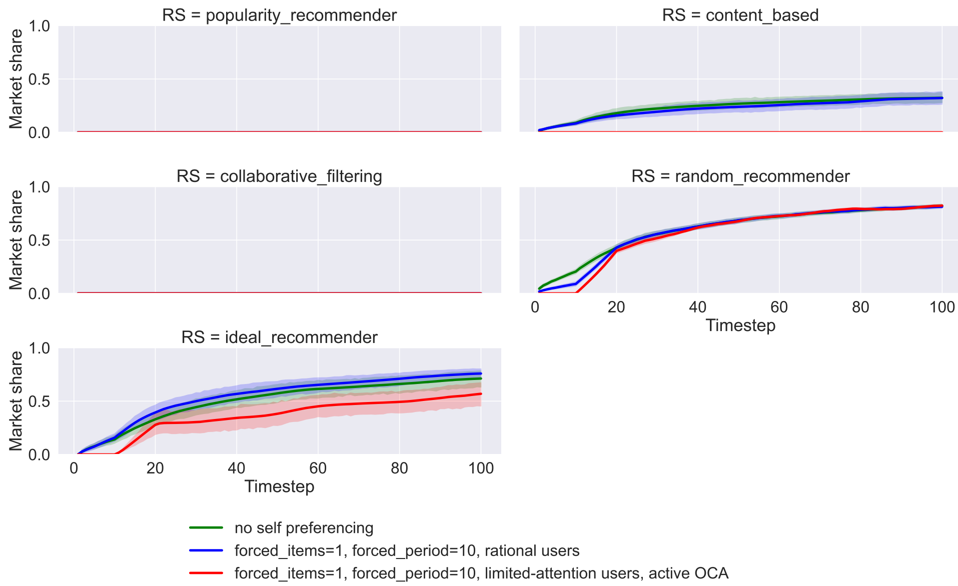

Figure 7: Market share of new entrants under a self-preferencing platform.

self-preferencing stops.[10]

These results carry two important policy implications. First, self-preferencing gives more prominence and consequently increased market shares to preferred items, even if the platform is not using an active OCA (i.e. if it ranks items in a non-discriminatory order). If the ordering favours the preferred items, these will gain even more market shares. Markets on the other hand only get more concentrated than under a non-self-preferencing platform if users have limited attention and the platform uses an active OCA. Entry for new items becomes more difficult (and virtually impossible if users have limited attention and the platform gives more prominence to its preferred product in the choice architecture). This is in line with results in Calvano et al. (2021), who find that the manipulation of recommendations by the platform may be more of an exclusionary abuse (favouring the platform's preferred items) than exploitative one.

Second, our results demonstrate the difficulty of monitoring and enforcing that platforms do not self-preference. If some items are given disproportionate exposure even for a short interval (1 period in our simulation), they can give significant advantage to the preferred items over their competitors due to feedback loops. Proving that a platform 'self-preferenced' certain items for such a short period can become practically impossible.

Moreover, the simulations above can be informative for actual cases, such as the Amazon Buy Box case.[11] The algorithm used by Amazon for the Buy Box is likely to be different from those in our simulations (firstly, because the buy box algorithm is less of a recommender system, more of a rule based ranking algorithm, and secondly, unlike in the buy box case, we do not include

[10]For example in the content-based RS the weights of the preferred items increase, and the RS remembers these weights.

[11]`https://ec.europa.eu/competition/antitrust/cases1/202310/AT_40703_8990760_1533_5.pdf`

price in our model). Nevertheless, our framework can shed light on the effectiveness of remedies in similar settings (platform with the ability to influence the OCA). One of the remedies was in the Buy Box case was to treat all sellers equally when ranking the offers for the purposes of the selection of the Buy Box winner. The green line in our figures can be thought of as a scenario that reflects the desired objective of this remedy (neutrality in ranking). Our findings provide support to the intuition that this remedy would help make markets less concentrated than what they would be under a platform that uses an active OCA, like Amazon's Buy Box (see Figure 6). But unless the self-preferencing is erased from the RS (and from its feedback loops), this remedy itself is inadequate to fully remedy the impact on who the winners are.

7 Reducing market-concentrating effects by design

If the market-consolidating impact is due to biases in the outcome generated by the RS, then mitigating or eliminating these biases could help reduce the undesired market impact. We use two intuitively simple approaches to do this. In the first one, we combine each RS with a random RS. In the second one we change the RS to select the recommendation set on a probabilistic basis. Both methods are intended to increase the level of exploration in the RS.

7.1 Inclusion of additional objectives

One approach to increase exploration is to include additional objectives within the RS, to be jointly optimised alongside 'relevance'. For example, some RS designs specifically include as objectives: 'diversity' (Helberger et al. 2018, Hamedani & Kaedi 2019, Wasilewski & Hurley 2016); 'fairness' (Farnadi et al. 2018, Mehrotra et al. 2018, Abdollahpouri et al. 2017); or 'serendipity' (Kotkov et al. 2016, 2020, Akiyama et al. 2010, Ziarani & Ravanmehr 2021). Another way to achieve a similar outcome is to adjust the accuracy-oriented algorithm, for example by changing CF to recommend items chosen by consumers who are not the most similar, but are somewhat different (Said et al. 2013, Tuzhilin & Adamopoulos 2013).

In a similar vein we change our recommenders to force items other than the ones with the highest predicted user-item score to be included in the recommendation set. We do this through a hybrid RS, which combines each of the recommenders with a random RS. For example, a music streaming platform could include songs that similar users listened to, but could also include a few randomly chosen songs with the expectation that listeners might discover something they did not know they liked. The concept is similar to the idea in Zhang et al. (2020), who show that that users' high reliance on the recommender system provides highly sub-optimal performance outcomes, which can be changed by adding random items to the recommendation set. It is also similar to provisions of the EU Digital Markets Act, which require recommenders of very large online platforms to combine personalised recommendations with non-personalised ones.[12]

[12]Article 38 of The Digital Services Act: "In addition to the requirements set out in Article 27, providers of very large online platforms and of very large online search engines that use recommender systems shall provide at least one

To implement this, we change the recommended set of items the following way, where we are describing the method specifically for the parameters value $r = 0.5$ (called random_items_per_within the code-base; see Table A1), which corresponds to showing $k.r = \frac{k}{2}$ randomly drawn items. Let $\mathcal{L}'_u(t)$ denote the first $k/2$ items from the sequence ω in Equation (3), and let $\mathcal{L}^R_u(t)$ denote a $k/2$ size set of items drawn randomly from $\mathcal{I}_t$. Then the set of items recommended to user u at timestep t is:

$$\mathcal{L}_u(t) = \mathcal{L}'_u(t) \cup \mathcal{L}^R_u(t). \tag{11}$$

That is, the RS recommends $k/2$ items with the highest predicted user-item scores, plus the same number of items randomly chosen from the pool of items not in the $k/2$ set of items. [13]

Figure 8: The impact of combining with random recommendations - Market share of 1% most popular items (limited-attention users).

We focus on limited-attention users in the main text because entry is more difficult in this scenario, therefore we get a better understanding how more exploration can ameliorate this issue.

Figure 8 shows that concentration is lower (even under a popularity RS) where random recommendations foster exploration (Figure C1 in the Online Appendix shows that the difference is there but smaller under rational users). The difference is most pronounced where users have limited attention (i.e. where they are more likely to engage in exploration).

Figure 9 shows that increasing exploration improves the probability for new entrants to gain market shares where the cold start problem otherwise creates entry barriers (under collaborative filtering and content-based RS, where with limited-attention users there would be no entry).

Adding random items to the recommendation set has the largest impact of reducing market concentration and increasing entry for the collaborative filtering and popularity RS. At the same

option for each of their recommender systems which is not based on profiling"

[13]A natural alternative would be to sample the random items only from those items in $\mathcal{I}_t$ that are not in $\mathcal{L}'_u(t)$ but the difference would be negligible.

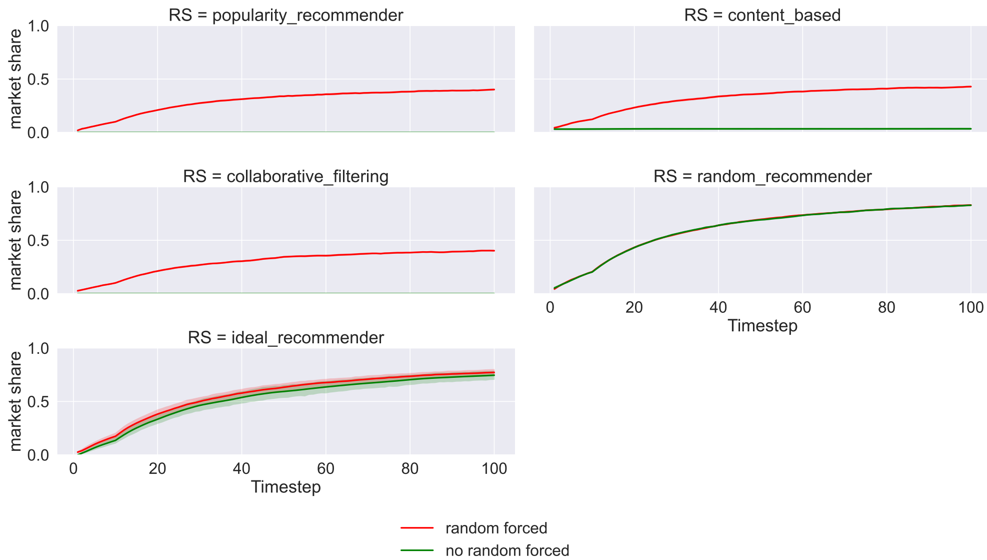

Figure 9: The impact of combining with random recommendations - Market share of new entrant items (limited-attention users).

time, it does not increase the market share of new entrants, which suggests that the impact of these hybrid RS more likely comes from reducing the concentration of incumbent items rather than increasing the market share of new entrants.

7.2 Probabilistic recommendations

Another way to enhance exploration is by changing how the RS selects the items to include in the recommendation set. Instead of simply choosing the k items with the highest predicted user-item scores, we allowed the RS to probabilistically (proportionate to their predicted score) sample items for the recommendations. To do this, for each user u at time t the RS orders the prediction scores as $\hat{s}_{u_1}(t) \leq \hat{s}_{u_2}(t) \leq ... \leq \hat{s}_{u_I}(t)$ and multiplies this vector of ranked scores with an exponentially decreasing probability distribution. The RS then draws k items according to this distribution.

Figure 10 shows that with limited-attention users there is a reduction in concentration as a result of probabilistic recommendations in all but the random RS. This is because under probabilistic recommendations, there is more randomness in what appears as the first listed item, and if users have limited attention, they are more likely to end up interacting with items that are listed on top, irrespective of whether they are the most popular ones (this is similar to one of the findings in Helmers et al. (2019)). Figure C3 in the Online Appendix show similar results under rational users. Adding more exploration through probabilistic recommendations also lowers entry barriers (Figure 11).

Both mixing the RS with random recommendations, and forcing probabilistic recommendations enhance exploration to help reduce the tendency of feedback loops to amplify supplier-side market concentration. Additionally, in our experiments we also used a variation of the above, where we

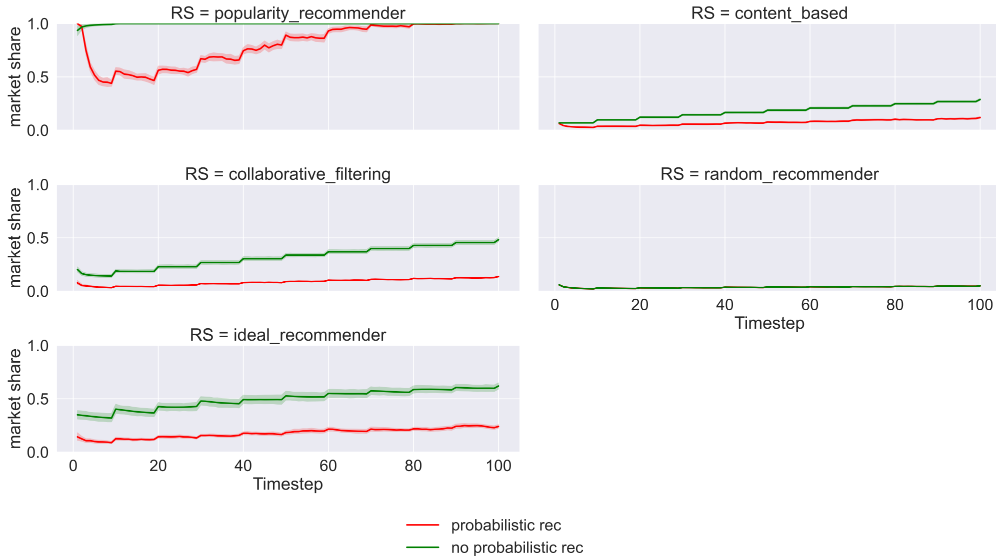

Figure 10: The impact of probabilistic recommendations - Market share of 1% most popular items (behavioural users).

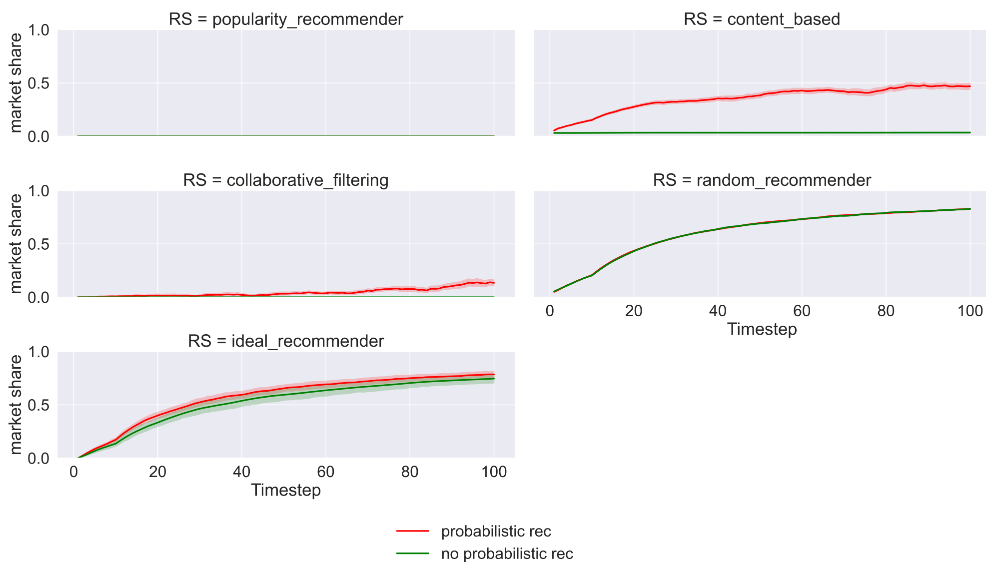

Figure 11: The impact of probabilistic recommendations - Market share of new entrants (limited-attention users).

force the RS to do more exploration by recommending random items but only from the set of new entrant items. This produced similar results to the RS where it is forced to include random items in the recommendation set. We did not report these results in the paper but these are replicable by running our code.

7.3 The market-concentrating effect and recommendation quality trade-off

Although based on the above discussion it would seem straightforward that reducing RS biases could improve the fairness of competition between suppliers on the platform, previous literature warns that increasing fairness may come at the cost of reducing recommendation relevance (Mehrotra et al. 2018). Our two simple solutions to increase exploration both reduced the market consolidating effect of RS. In this section we look at what happens when we activate these two changes to the original RS design.

To measure recommendation quality we introduce a portmanteau of three metrics. (1) The *correlation* coefficient between the predicted and the true user-item scores. (2) *Recommendation ranking* is the proportion of the k recommended items that are among the top k items on the user's true preference ranking. (3) *Interaction ranking* is the inverse of the ranking of the interacted item on the user's own ranking of item preferences (e.g. if the RS recommends and the user interacts with the item that is highest ranked on the user's ranking, then the interaction ranking is 1/1). Normalised recommendation quality is the unweighted arithmetic mean of these three metrics.

Figures C5 and C6 report how the quality of recommendations changes when we force the RS to do more exploration to reduce the market consolidating impact of RS under rational users. First of all, both forcing probabilistic recommendations, and forcing the RS to include random items improved social outcomes (see above), but it comes at the cost of reduced recommendation quality for the users. Finally, we can see that the quality of recommendation drops even if the RS is fully aware of users' preferences (ideal RS) when we force more exploration.

Looking at the quality of recommendations also draws light to some other observations. A pure collaborative filtering RS performs worse than a pure content-based one on both recommendation quality and competition impact. One of the possible reasons is that the content-based filtering RS is less likely to suffer from inherent biases (such as cold start problem) than collaborative filtering.

Finally, we use another measure of the quality of recommendations, by looking at recommendation similarity between the most similar pairs of users. This metric can be used as a measure of how good the RS is in predicting that two users are similar and therefore need similar recommendations. Apart from being a measure of the quality of the recommendations, this type of homogeneity is important as it means more intensive competition for users with similar preferences. Figures **??** and **??** show that under probabilistic recommendations, the RS is less able to recommend similar items to the most similar pairs of users, which hints that these recommendations are of lower quality.

It is important to emphasise that this trade-off is likely to only be a trade-off in the short run. Having more exploration improves the predictive power of RS, and whilst it may lead to a short-term fall in recommendation quality, in the long-run, this exploration should make the RS deliver better recommendations.

8 Swaying preferences

In real life recommenders it is likely that users' tastes are influenced by the things they consume. For example, a listener on a music streaming platform may discover a new song through a recommender, which she develops a taste for. If the platform is able to influence consumers' taste, it is possible that preferences are adjusted towards more popular items, reinforcing popularity bias even more.

Previous works that use a simulation setup differ in their assumption about the consumers. Some of these assume that consumers' tastes do not change. Other papers allow for a more dynamic consideration of consumers. In Song et al. (2019) customers' tastes can satiate, which can trigger the customer to move on to the consumption of new types of content. Similarly, Jiang et al. (2019) considers user preferences that may shift over time, and in Warlop et al. (2018) boredom drives the change in consumer tastes. Using a laboratory experiment, Adomavicius et al. (2013) find that recommenders can anchor consumer choices and therefore can sway their preferences.

To look at the impact of swaying user preferences, we introduce a preference adjustment parameter, denoted by ρ, where $\rho = 0$ means no adjustment, and $\rho = 1$ implies perfect adjustment to the interacted item attributes. For each user, at each timestep, their preference vectors (x_u) are rotated towards the attribute vectors of the items they interacted with (y_i), in the following way:

$$x_u(t+1) = \left(\frac{\sin(1-\rho)\cdot\omega}{\sin(\omega)\cdot\left(\frac{x_u(t)}{\|x_u(t)\|}\right)^{\intercal}} + \frac{\sin(\rho\omega)\cdot\omega}{\sin(\omega)\cdot\left(\frac{y_i}{\|y_i\|}\right)^{\intercal}} \right)^{\intercal} \cdot \|x_u\|$$

where $\omega = \arccos\left(\frac{x_u}{\|x_u\|}\cdot\frac{y_i}{\|y_i\|}\right)$.

Figure 12 plots the result for two values of the preference adjustment parameter $\rho = [0, 0.75]$, and shows the impact of sway on the level of market concentration. In general, swaying users' tastes towards the attributes of the interacted items has little impact on the overall level of market concentration (although it makes markets more concentrated under the content based and ideal RS). But what is more important, is that it makes it virtually impossible for new items to gain market shares under all but the random recommender, as shown on Figure 13 below.

9 Implications of our findings and conclusions

One of the main objectives of this paper has been to improve our understanding of the impact of RS designs on competition between suppliers on an online platform. For this effect, we looked at different RS designs on a subscription-based platform and simulated the user interactions with the recommendations that each design offered. We were particularly interested in three canonical RS designs that are most commonly used by platforms: popularity, collaborative filtering, and content based RS. We showed that popularity and collaborative filtering RS led to more barriers to entry, and more concentrated markets. Because these RS are frequently used by platforms, it is likely that suppliers compete on online markets that are more concentrated than what it would be under an ideal RS on most currently active online platforms.

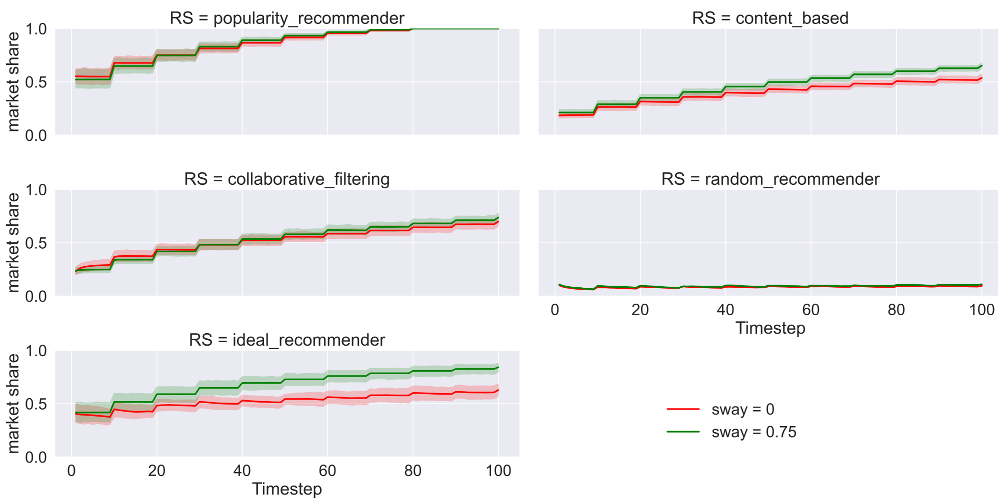

Figure 12: The impact of swaying user tastes - Market share of 1% most popular items (rational users).

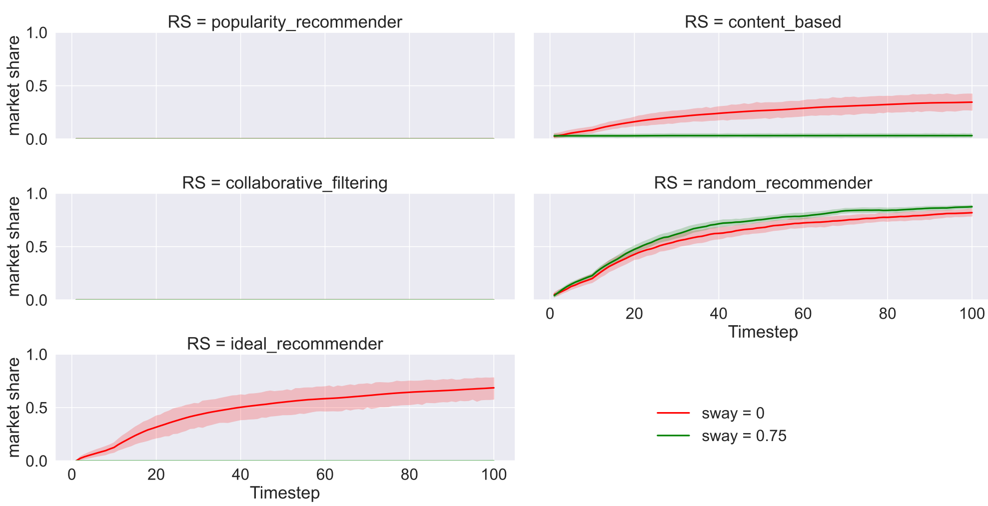

Figure 13: The impact of swaying user tastes - Market share of new entrant items (rational users).

When we relax on the assumption of rational users, and introduce users with limited attention, we find that the platform's choice architecture plays an important role. If the platform uses a passive choice architecture (listing items in the order of predicted relevance), then limited attention implies that users ignore the most popular products and choose what is listed at the top. If however the platform gives more prominence to the most popular items, it enhances the market shares of these items. Self-preferencing platforms can make markets more concentrated with the preferred items winning the competition between suppliers.

These findings have some immediate implications. Firstly, if competition between suppliers is limited, and the most popular items are recommended disproportionately more than other items, it can have a short-term impact of reducing effective choice for customers on the platform. These popular items may be able to command a price premium. This is less likely to harm consumers if rivals can contest the position of 'most popular' product through pricing low or investing in high quality, and thereafter gain the benefits that such popularity brings. By contrast, if it is almost impossible for a rival to gain the position of 'most popular' product (for example due to the cold start problem in collaborative filtering or popularity RS), the impact of 'popularity bias' may dampen the competitive incentives for rivals too, with prices increasing for all suppliers.

Moreover, if customers are herded towards the most popular items, and if less popular content, or new content is underrepresented in the recommendation sets, it can hinder the longer term incentives of suppliers to provide certain types of content. This can lead to the homogenisation of content, and consequently to lesser incentives to innovate.

Secondly, the harm to supplier competition is likely to be particularly serious where the platform itself faces limited competition, for example due to network effects and scale economies, or where it has 'single homing' end-users, i.e. where the platform is the essential route to end-users, with strong 'bottleneck' market power over suppliers. RS biases on such platforms are more likely to affect competition across the upstream supply market as a whole.

Our results also suggest that hybrid recommenders and forcing recommenders to do more exploration can reduce these negative effects. But more exploration may come at the cost of diminished recommendation quality. This is important not just for the platform, but for any policy response. The trade-off between recommendation quality and market concentration can translate to a short-term, long-term trade-off. Forcing more exploration in the RS may reduce short term recommendation quality (and therefore short term consumer surplus), but in the long-run these are likely to enhance the ability of the RS to predict customers' preferences. The paper also shows that users that exercise less attention in choosing from the recommendation set are less likely to choose the popular items, and therefore, inadvertently, lower the market consolidating effect of popularity bias.

References

Abdollahpouri, H., Burke, R. & Mobasher, B. (2017), Controlling popularity bias in learning-to-rank recommendation, *in* 'Proceedings of the eleventh ACM conference on recommender systems', pp. 42–46.

Abdollahpouri, H., Mansoury, M., Burke, R. & Mobasher, B. (2019), 'The unfairness of popularity bias in recommendation', *arXiv preprint arXiv:1907.13286* .

Adomavicius, G., Bockstedt, J. C., Curley, S. P. & Zhang, J. (2013), 'Do recommender systems manipulate consumer preferences? a study of anchoring effects', *Information Systems Research* **24**(4), 956–975.

Aguiar, L. & Waldfogel, J. (2021), 'Platforms, power, and promotion: Evidence from Spotify playlists', *The Journal of Industrial Economics* **69**(3), 653–691.

Akiyama, T., Obara, K. & Tanizaki, M. (2010), Proposal and evaluation of serendipitous recommendation method using general unexpectedness., *in* 'PRSAT@ RecSys', pp. 3–10.

Antal, D., Fletcher, A. & Ormosi, P. (2021), 'Music streaming: Is it a level playing field?', *Competition Policy International: Antitrust Chronicle* .

Aridor, G. & Gonçalves, D. (2022), 'Recommenders' originals: The welfare effects of the dual role of platforms as producers and recommender systems', *International Journal of Industrial Organization* **83**, 102845.

Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., Weintraub, O. & Netanely, I. (2013), Movie recommender system for profit maximization, *in* 'Proceedings of the 7th ACM conference on Recommender systems', pp. 121–128.

Ben-Porat, O. & Tennenholtz, M. (2018), 'A game-theoretic approach to recommendation systems with strategic content providers', *Advances in Neural Information Processing Systems* **31**.

Bennett, J., Lanning, S. et al. (2007), The netflix prize, *in* 'Proceedings of KDD cup and workshop', Vol. 2007, p. 35.

Bobadilla, J., Ortega, F., Hernando, A. & Bernal, J. (2012), 'A collaborative filtering approach to mitigate the new user cold start problem', *Knowledge-based systems* **26**, 225–238.

Bourreau, M. & Gaudin, G. (2021), 'Streaming platform and strategic recommendation bias', *Journal of Economics and Management Strategy* **31**, 25–47.

Brynjolfsson, E., Hu, Y. & Simester, D. (2011), 'Goodbye pareto principle, hello long tail: The effect of search costs on the concentration of product sales', *Management Science* **57**(8), 1373–1386.

Calvano, E., Calzolari, G., Denicolò, V. & Pastorello, S. (2021), 'Artificial intelligence, product recommendations and market concentration', *Ongoing work. Presented at Toulouse School of Economics Digital Economics Conference 14 january 2022* .

Carare, O. (2012), 'The impact of bestseller rank on demand: Evidence from the app market', *International Economic Review* **53**(3), 717–742.

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

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

Cure, M., Hunold, M., Kesler, R., Laitenberger, U. & Larrieu, T. (2021), Vertical integration of platforms and product prominence, Technical report, Working paper.

De Clippel, G., Eliaz, K. & Rozen, K. (2014), 'Competing for consumer inattention', *Journal of Political Economy* **122**(6), 1203–1234.

De Corniere, A. & Taylor, G. (2014), 'Integration and search engine bias', *The RAND Journal of Economics* **45**(3), 576–597.

De los Santos, B. & Koulayev, S. (2017), 'Optimizing click-through in online rankings with endogenous search refinement', *Marketing Science* **36**(4), 542–564.

Donnelly, R., Kanodia, A. & Morozov, I. (2023), 'Welfare effects of personalized rankings', *Marketing Science* .

Dzyabura, D. & Hauser, J. R. (2019), 'Recommending products when consumers learn their preference weights', *Marketing Science* **38**(3), 417–441.

Ekstrand, M. D. (2020), Lenskit for python: Next-generation software for recommender systems experiments, *in* M. d'Aquin, S. Dietze, C. Hauff, E. Curry & P. Cudré-Mauroux, eds, 'Proc. of CIKM '20: The 29th ACM International Conference on Information and Knowledge Management', ACM, pp. 2999–3006.

Farnadi, G., Kouki, P., Thompson, S. K., Srinivasan, S. & Getoor, L. (2018), 'A fairness-aware hybrid recommender system', *arXiv preprint arXiv:1809.09030* .

Fletcher, A., Ormosi, P. & Savani, R. (2023), 'Recommender systems and supplier competition on platforms', *Journal of Competition Law and Economics* .

Fletcher, A., Ormosi, P., Savani, R. & Castellini, J. (2023), 'Biased recommender systems and supplier competition', `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4319311` .

Ghose, A., Ipeirotis, P. G. & Li, B. (2014), 'Examining the impact of ranking on consumer behavior and search engine revenue', *Management Science* **60**(7), 1632–1654.

Guo, G., Zhang, J. & Thalmann, D. (2014), 'Merging trust in collaborative filtering to alleviate data sparsity and cold start', *Knowledge-Based Systems* **57**, 57–68.

Hamedani, E. M. & Kaedi, M. (2019), 'Recommending the long tail items through personalized diversification', *Knowledge-Based Systems* **164**, 348–357.

Helberger, N., Karppinen, K. & D'acunto, L. (2018), 'Exposure diversity as a design principle for recommender systems', *Information, Communication & Society* **21**(2), 191–207.

Helmers, C., Krishnan, P. & Patnam, M. (2019), 'Attention and saliency on the internet: Evidence from an online recommendation system', *Journal of Economic Behavior & Organization* **161**, 216–242.

Hu, Y., Koren, Y. & Volinsky, C. (2008), Collaborative filtering for implicit feedback datasets, *in* '2008 Eighth IEEE international conference on data mining', Ieee, pp. 263–272.

Hunold, M., Kesler, R. & Laitenberger, U. (2020), 'Rankings of online travel agents, channel pricing, and consumer protection', *Marketing Science* **39**(1), 92–116.

Jagadeesan, M., Garg, N. & Steinhardt, J. (2022), 'Supply-side equilibria in recommender systems'.

Jiang, R., Chiappa, S., Lattimore, T., György, A. & Kohli, P. (2019), Degenerate feedback loops in recommender systems, *in* 'Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society', pp. 383–390.

Joachims, T., Granka, L., Pan, B., Hembrooke, H. & Gay, G. (2005), Accurately interpreting clickthrough data as implicit feedback, *in* 'Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', Association for Computing Machinery, p. 154–161.

Kotapati, B., Mutungi, S., Newham, M., Schroeder, J., Shao, S. & Wang, M. (2020), 'The antitrust case against apple', *Available at SSRN 3606073* .

Kotkov, D., Veijalainen, J. & Wang, S. (2020), 'How does serendipity affect diversity in recommender systems? a serendipity-oriented greedy algorithm', *Computing* **102**(2), 393–411.

Kotkov, D., Wang, S. & Veijalainen, J. (2016), 'A survey of serendipity in recommender systems', *Knowledge-Based Systems* **111**, 180–192.

Lee, D. & Hosanagar, K. (2021), 'How do product attributes and reviews moderate the impact of recommender systems through purchase stages?', *Management Science* **67**(1), 524–546.

Lee, K. H. & Musolff, L. (2021), Entry into two-sided markets shaped by platform-guided search, Technical report, mimeo, Princeton University.

Li, L., Chen, J. & Raghunathan, S. (2018), 'Recommender system rethink: Implications for an electronic marketplace with competing manufacturers', *Information Systems Research* **29**(4), 1003–1023.

Li, L., Chen, J. & Raghunathan, S. (2020), 'Informative role of recommender systems in electronic marketplaces: A boon or a bane for competing sellers.', *MIS Quarterly* **44**(4).

Li, X., Grahl, J. & Hinz, O. (2022), 'How do recommender systems lead to consumer purchases? a causal mediation analysis of a field experiment', *Information Systems Research* **33**(2), 620–637.

Long, F. & Liu, Y. (2023), 'Platform manipulation in online retail marketplace with sponsored advertising', *Marketing Science* .

Lucherini, E., Sun, M., Winecoff, A. & Narayanan, A. (2021), 'T-recs: A simulation tool to study the societal impact of recommender systems', *arXiv preprint arXiv:2107.08959* .

Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M. & Diaz, F. (2018), Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems, *in* 'Proceedings of the 27th acm international conference on information and knowledge management', pp. 2243–2251.

Nagulendra, S. & Vassileva, J. (2014), Understanding and controlling the filter bubble through interactive visualization: a user study, *in* 'Proceedings of the 25th ACM conference on Hypertext and social media', pp. 107–115.

Nguyen, T. T., Hui, P.-M., Harper, F. M., Terveen, L. & Konstan, J. A. (2014), Exploring the filter bubble: the effect of using recommender systems on content diversity, *in* 'Proceedings of the 23rd international conference on World wide web', pp. 677–686.

O'Brien, M. & Keane, M. (2006), Modeling result-list searching in the world wide web: The role of relevance topologies and trust bias, *in* 'Proceedings of the 28th Annual Conference of the Cognitive Science Society', p. 1881–1886.

OECD (2021), 'Harnessing the productivity benefits of online platforms', *https://www.oecd.org/global-forum-productivity/events/Harnessing-the-productivity-benefits-of-online-platforms.pdf* .

O'Callaghan, D., Greene, D., Conway, M., Carthy, J. & Cunningham, P. (2015), 'Down the (white) rabbit hole: The extreme right and online recommender systems', *Social Science Computer Review* **33**(4), 459–478.

Padilla, J., Perkins, J. & Piccolo, S. (2022), 'Self-preferencing in markets with vertically integrated gatekeeper platforms', *The Journal of Industrial Economics* **70**(2), 371–395.

Pagano, R., Cremonesi, P., Larson, M., Hidasi, B., Tikk, D., Karatzoglou, A. & Quadrana, M. (2016), The contextual turn: From context-aware to context-driven recommender systems, *in* 'Proceedings of the 10th ACM conference on recommender systems', pp. 249–252.

Pathak, B., Garfinkel, R., Gopal, R. D., Venkatesan, R. & Yin, F. (2010), 'Empirical analysis of the impact of recommender systems on sales', *Journal of Management Information Systems* **27**(2), 159–188.

Said, A., Fields, B., Jain, B. J. & Albayrak, S. (2013), User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm, *in* 'Proceedings of the 2013 conference on Computer supported cooperative work', pp. 1399–1408.

Schmit, S. & Riquelme, C. (2018), Human interaction with recommendation systems, *in* 'International Conference on Artificial Intelligence and Statistics', PMLR, pp. 862–870.

Song, Y., Sahoo, N. & Ofek, E. (2019), 'When and how to diversify — a multicategory utility model for personalized content recommendation', *Management Science* **65**(8), 3737–3757.

Sun, W., Khenissi, S., Nasraoui, O. & Shafto, P. (2019), Debiasing the human-recommender system feedback loop in collaborative filtering, *in* 'Companion Proceedings of The 2019 World Wide Web Conference', pp. 645–651.

Takács, G., Pilászy, I. & Tikk, D. (2011), Applications of the conjugate gradient method for implicit feedback collaborative filtering, *in* B. Mobasher, R. D. Burke, D. Jannach & G. Adomavicius, eds, 'Proc. of the ACM Conference on Recommender Systems, RecSys 2011', pp. 297–300.

Teppan, E. C. & Zanker, M. (2015), 'Decision biases in recommender systems', *Journal of Internet Commerce* **14**(2), 255–275.

Tuzhilin, A. & Adamopoulos, P. (2013), 'Probabilistic neighborhood selection in collaborative filtering systems', *NYU: Center for Business Analytics Working Papers (Topic)* .

Ursu, R. M. (2018), 'The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions', *Marketing Science* **37**(4), 530–552.

Waldfogel, J. (2017), 'How digitization has created a golden age of music, movies, books, and television', *Journal of economic perspectives* **31**(3), 195–214.

Wang, N. & Chen, L. (2021), User bias in beyond-accuracy measurement of recommendation algorithms, *in* 'Proceedings of the 15th ACM Conference on Recommender Systems', pp. 133–142.

Warlop, R., Lazaric, A. & Mary, J. (2018), 'Fighting boredom in recommender systems with linear reinforcement learning', *Advances in Neural Information Processing Systems* **31**.

Wasilewski, J. & Hurley, N. (2016), Incorporating diversity in a learning to rank recommender system, *in* 'FLAIRS Conference'.

Yao, S. & Huang, B. (2017), 'Beyond parity: Fairness objectives for collaborative filtering', *Advances in neural information processing systems* **30**.

Zentner, A., Smith, M. & Kaya, C. (2013), 'How video rental patterns change as consumers move online', *Management Science* **59**(11), 2622–2634.

Zhang, J., Adomavicius, G., Gupta, A. & Ketter, W. (2020), 'Consumption and performance: Understanding longitudinal dynamics of recommender systems via an agent-based simulation framework', *Information Systems Research* **31**(1), 76–101.

Zhang, L. (2018), 'Intellectual property strategy and the long tail: Evidence from the recorded music industry', *Management Science* **64**(1), 24–42.

Zhang, X., Ferreira, P., Godinho de Matos, M. & Belo, R. (2021), 'Welfare properties of profit maximizing recommender systems: Theory and results from a randomized experiment.', *MIS Quarterly* **45**(1).

Zhou, B. & Zou, T. (2023), ‘Competing for recommendations: The strategic impact of personalized product recommendations in online marketplaces’, *Marketing Science* **42**(2), 360–376.

Zhou, M., Zhang, J. & Adomavicius, G. (2023), ‘Longitudinal impact of preference biases on recommender systems’ performance’, *Information Systems Research* .

Ziarani, R. J. & Ravanmehr, R. (2021), ‘Serendipity in recommender systems: A systematic literature review’, *Journal of Computer Science and Technology* **36**(2), 375–396.

Appendix

Experimental parameters

The figures presented in the main text show the average and the standard deviation of the outcome measures derived from running simulations under all combination of parameter settings described below. Table A1 shows the parameters used in our experiments, where multiple values in square brackets mean that different configurations have been experimented with.

Parameter	Math notation	Value
training	T'	[10, 1000]
timesteps	T	100
num_users	$U = \|\mathcal{U}\|$	100
num_items	$I_1 = \|\mathcal{I}_1\|$	200
num_creators	ν	10
num_items_per_iter	k	10
num_attributes	a	30
num_latent_attributes (for CF)	length of $\bar{x}$ and $\bar{y}$	30
min_preference_per_attribute	M_1	-20
max_preference_per_attribute	M_2	100
limited_attention	θ	[-1.0,0]
num_forced_items	b	[0,1]
forced_period	τ	[1,10]
drift	ρ	[0, 0.75]
random_items_per_iter	r	[0, 0.5]
probabilistic_recommendations		[False, True]
horizontally_differentiated_only		[True,False]
sort_rec_per_popularity		[True,False]
train_between_steps (feedback loops)		True
individual_rationality		True
repeated_items		True

Table A1: Experimental parameters of `t-recs` used our experiments.

- *num_users* and *num_items* are respectively the number of users and items present in the simulation;
- *num_creators:* the number of creators in the simulation. Each creator can add a new item to the simulated recommender system at every simulation step with probability p_creation, which is always probability 1 in our experiments. When num_creators is 0, no new item is added throughout the simulation;
- *num_items_per_iter:* the argument controlling for the number of items recommended to each user at each timestep, chosen according to the recommender system model (denoted as k in the paper);

- *timesteps:* the actual number of simulation steps, where the recommender system gives recommendations to the users based on its learned model. During these steps, the recommender system is learning from its own recommendations;
- *train_between_steps:* whether we turn feedback loops on (True) or off (False);
- *num_attributes:* the number of attributes used to model the items' properties;
- *num_latent_attributes:* the number of latent attributes used in collaborative filtering;
- *min_preference_per_attribute:* the most negative preference a user have towards an attribute;
- *max_preference_per_attribute:* the most positive preference a user have towards an attribute;
- *runs:* the number of individual simulations ran for each experiment;
- *individual_rationality:* whether users engage with items with negative utility;
- *repeated_items:* whether the same item can be recommended and interacted with in more than one iteration;
- *training:* the number of startup steps used to learn the recommender system before allowing it to give tailored recommendations to the users. During this startup period, random items are recommended to the users. The RS uses these interactions to predict user preferences;
- *random_items_per_iter:* the proportion of random items added to the recommendation set (see Section 7.1);
- *probabilistic_recommendations:* indicates if recommended items are selected deterministically based on their predicted utility for the users (eventual ties are broken at random), or probabilistically based on the same predicted utility (see Section 7.2).
- *horizontally_differentiated_only:* sets whether items are only horizontally or also vertically differentiated;
- *limited_attention:* whether users have limited attention, see Section 5;
- *num_forced_items:* the number of items a self-preferencing platform forces into the recommendation set, see Section 6;
- *forced_period:* the number of periods a self-preferenced item is forced into the recommendation set, see Section 6;
- *sort_rec_per_popularity:* sets whether the platform uses an active (True) or passive (False) online choice architecture, see Section 5;
- *drift:* an argument that allows users to adjust their preferences to drift towards the attributes of the items previously recommended to, and interracted by them. 0 means no adjustment to original user preferences, and 1 is full adjustment of preferences, see Section 8).

Online appendix

A A stylised simulation example of a content-based RS

To help understand our simulation framework, assume a simple setup, where we have 4 users and 4 items, with 4 item attributes each. The user - attribute matrix is given by $\mathbf{U}$. For example, user U_1 values attribute A_4 most, and A_1 the least.

$$\mathbf{U} = \begin{array}{c} \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \\ 1 & 3 & 2 & 0 \\ 2 & 0 & 3 & 1 \end{bmatrix} \end{array}$$

The attribute - item matrix $\mathbf{I}$, which tells us which item has which attributes, is given by:

$$\mathbf{I} = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \end{array} \begin{array}{c} \begin{array}{cccc} I_1 & I_2 & I_3 & I_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Assume a content-based RS, with no pre-training. At t_1 the RS randomly chooses which items to recommend to the user. We choose $k = 2$, i.e. two items are recommended. The user then interacts with the item that maximises her utility (the item that have attributes which the user values more). For example, assume that items I_1 and I_2 are recommended to user U_1. I_1 has attributes A_1 and A_2, and I_2 has attribute A_3. User U_1 values attributes A_1 and A_2 together by 1 unit, attribute A_3 by 2 units, so U_1 will choose item I_2.

Table A1 shows the items recommended, and chosen by the user for the first three timesteps. The other users interact following the same procedure. The RS then regresses (using non-negative least squares - NNLS) the number of interactions on the binary attribute vectors to make a prediction for a user-item score, and makes recommendation at t_2 accordingly. Users choose at t_2, and the loop starts again. Note that by engaging with I_2, this item will have a larger coefficient in the NNLS regression in the next loop and is more likely to be recommended.

	User	Items recommended	Item chosen
t_1	RS makes random recommendation		
	U_1	I_1, I_2	I_2
	U_2	I_2, I_3	I_3
	U_3	I_2, I_4	I_4
	U_4	I_1, I_4	I_1
t_2	RS retrains		
	U_1	I_2, I_3	I_2
	U_2	I_3, I_1	I_1
	U_3	I_4, I_1	I_4
	U_4	I_1, I_3	I_3
t_3	RS retrains		
	U_1	I_2, I_3	I_2
	U_2	I_3, I_1	I_1
	U_3	I_4, I_1	I_4
	U_4	I_1, I_3	I_3

Table A1: A stylised content-based recommender example.

B Joint effect of changing RS parameters

In the discussion in the main text we introduced two ways of improving the social outcome of our recommenders (forcing probabilistic recommendations, and mixing recommendations with random recommendations in a hybrid model). We looked at the impact of each on of these ways individually. This section looks at the joint effect of these changes. To visualise the results, we focus at the outcome metrics at the the final time step only ($t = 100$) and we plot the level of market shares, homogeneity, and quality at this time step. The respective outcome metrics are plotted for each parameter combination (displayed on the horizontal axis), for each outcome metric separately, displaying outcomes under both rational and behavioural users.

Concentration: Figure B1 shows that concentration tends to be lower under behavioural users, and where our exploration parameters are turned on. This is in line with intuition. Position bias makes these users more likely to choose the recommendation listed on the top, which is more likely to be a random item when the RS is doing more exploration. On the other hand, even under rational users, concentration is lower if the RS is doing more exploration, and higher is users' preferences are swayed towards the attributes of items they interact with.

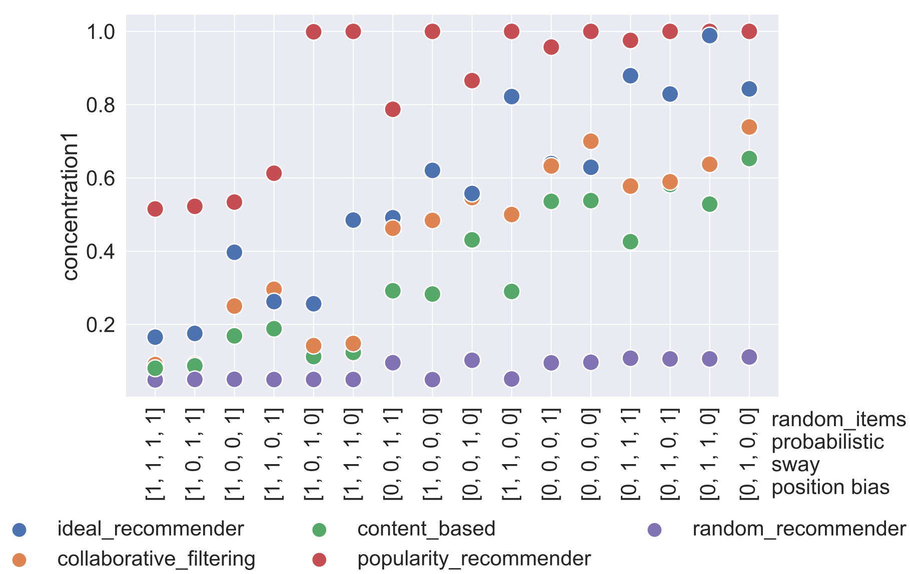

Figure B1: Joint effect on concentration.

Entry: Similarly to the impact on concentration, Figure B2 shows that new entrants are more likely to gain market shares under behavioral users and if the RS is doing more exploration, but here exploration seems more important than whether the users are more or less attentive. Doing more exploration also reduces the variance in new entrant market shares over the different RS designs (i.e. more exploration can help new entrants even if the core RS design is more biased, for example if it is a popularity RS). Having a mixed hybrid RS helps entry, irrespective of other parameter settings.

Homogeneity: Figure B3 shows that the ideal recommender recommends dissimilar items to users with dissimilar preferences. For the other recommenders, forcing the RS to recommend random

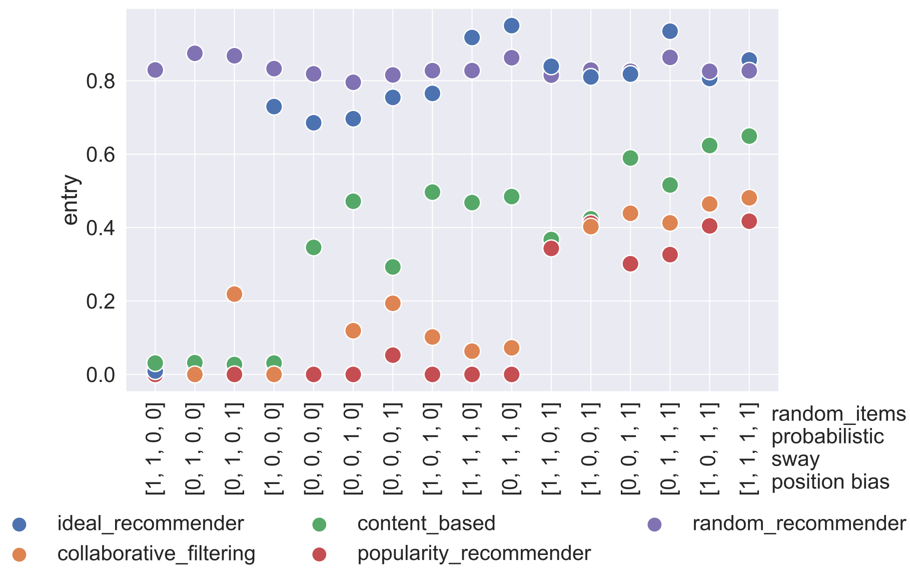

Figure B2: Joint effect on entry.

items helps keeping these recommendations dissimilar (see left hand side of Figure B3), which is especially the case if the users are not rational.

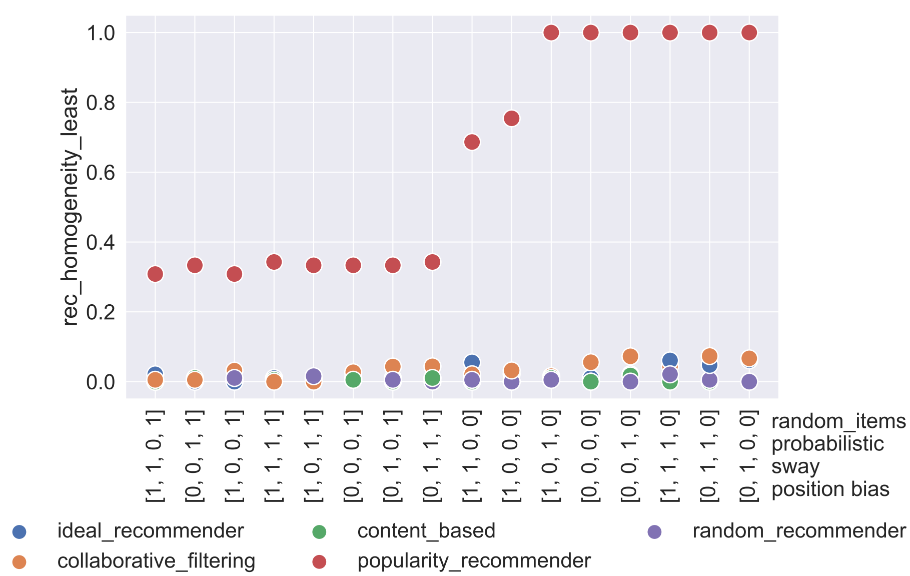

Figure B3: Joint effect on homogeneity (between least similar pair of users.)

Quality: Figures B4 and B5 show how recommendation quality (measured by the ratio of the user's preferred item's score and the score of the item the user interacted with) changes with different parameter settings. Most strikingly, when users' preferences are swayed by the recommendation, quality is higher, which is somewhat of a tautology, as the recommendation is closer to the users' preferences if these preferences are swayed towards the recommendation. With behavioural users, the recommended item that users interacted with tends to be more different from their most preferred item. This is due to the fact that these users are less attentive to choosing from the recommended set. Exploration is associated with lower quality in the short run, as it is more likely to make recommendations that are further from the user's preference (although it is also more likely to make recommendations that are closer to her preference). Nevertheless, this figure demonstrates the short-term trade-off that exists between more exploration and recommendation quality.

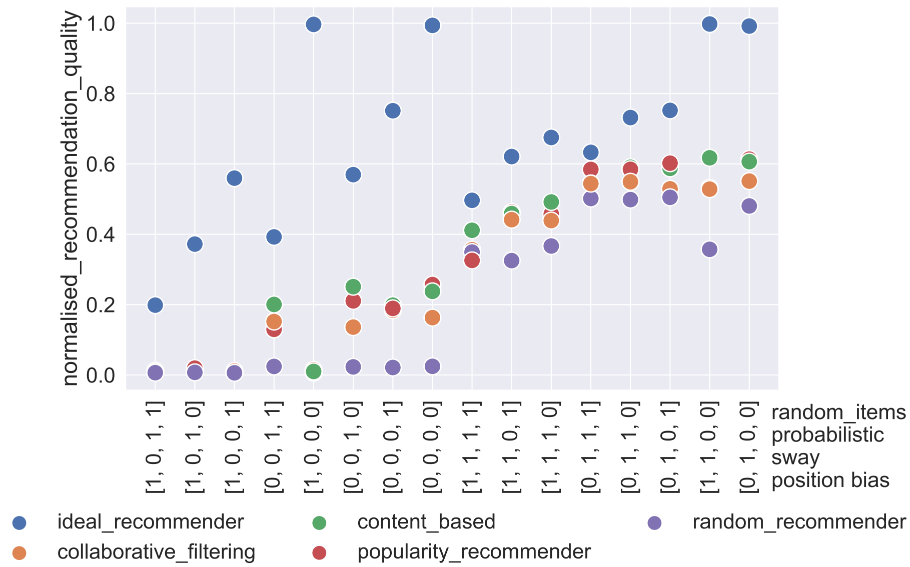

Figure B4: Joint effect on normalised recommendation quality.

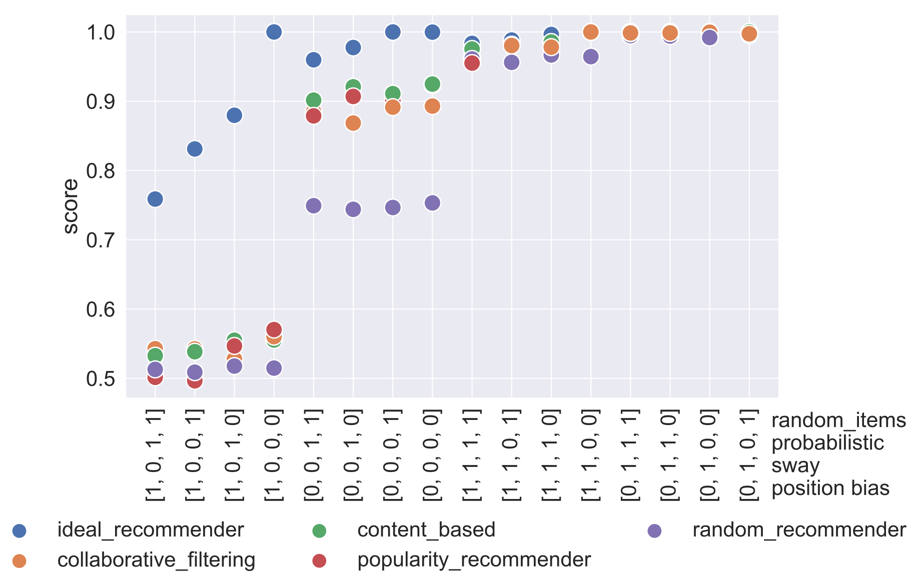

Figure B5: Joint effect on quality score.

C Additional figures and tables

C.1 Forcing exploration through including random items

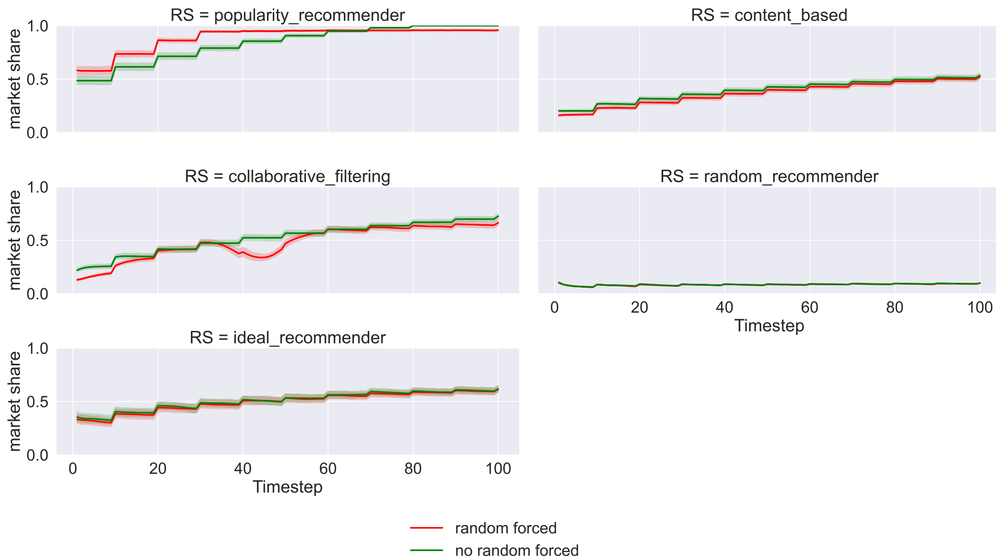

Figure C1: The impact of combining with random recommendations - Market share of 1% most popular items (rational users).

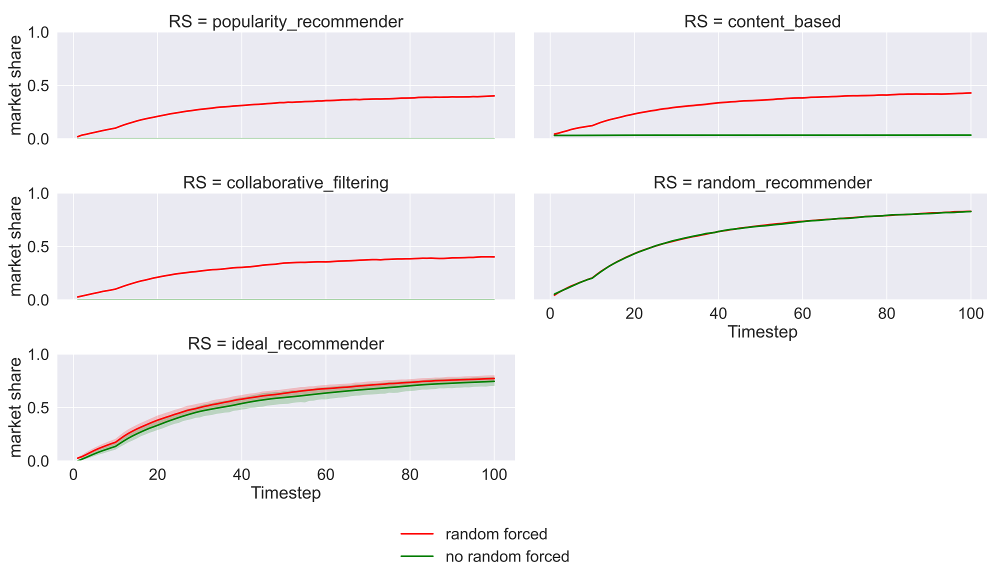

Figure C2: The impact of combining with random recommendations - Market share of new entrant items (limited attention users).

C.2 Forcing exploration through probabilistic recommendations

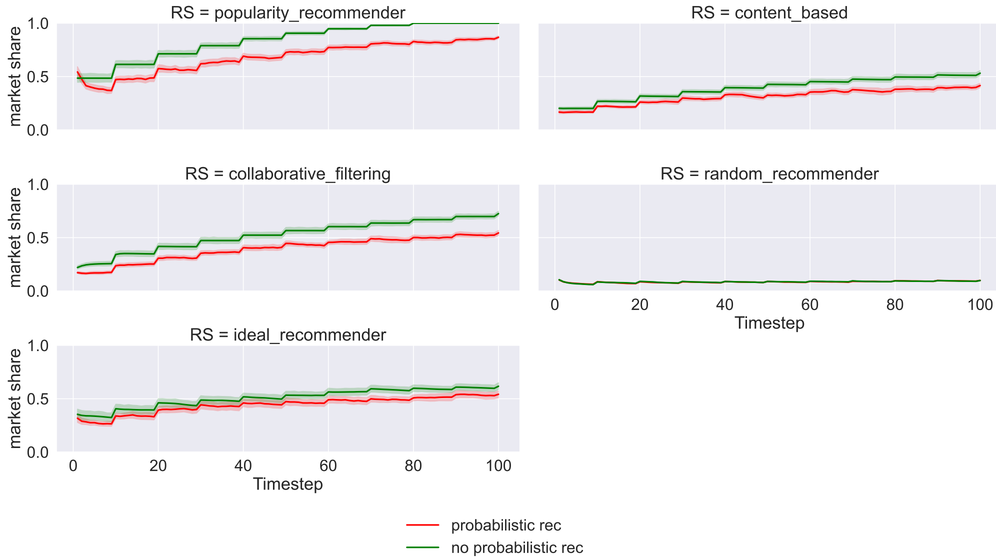

Figure C3: The impact of probabilistic recommendations - Market share of 1% most popular items (rational users).

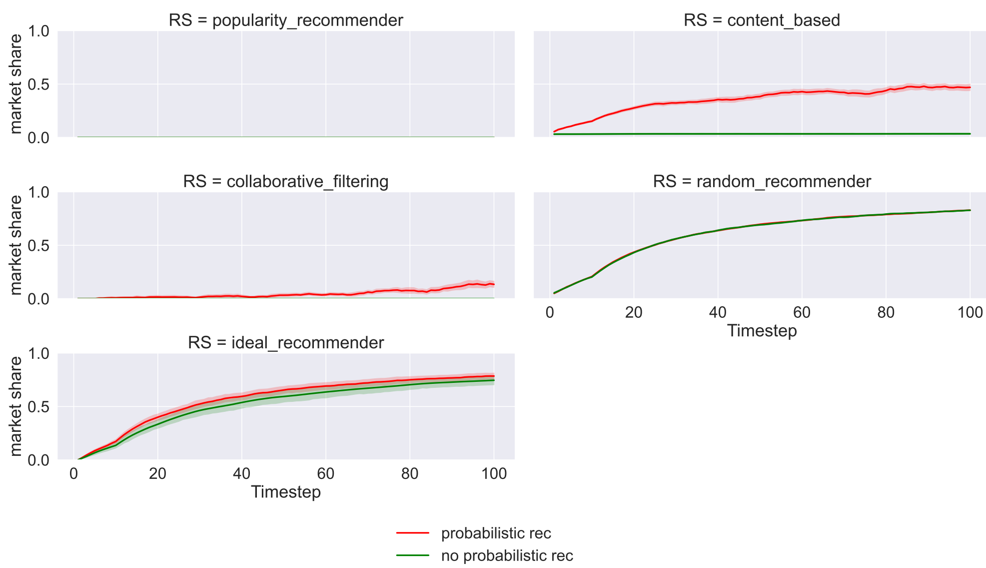

Figure C4: The impact of probabilistic recommendations - Market share of new entrants (limited attention users).

C.3 The quality of the recommendation vs more exploration

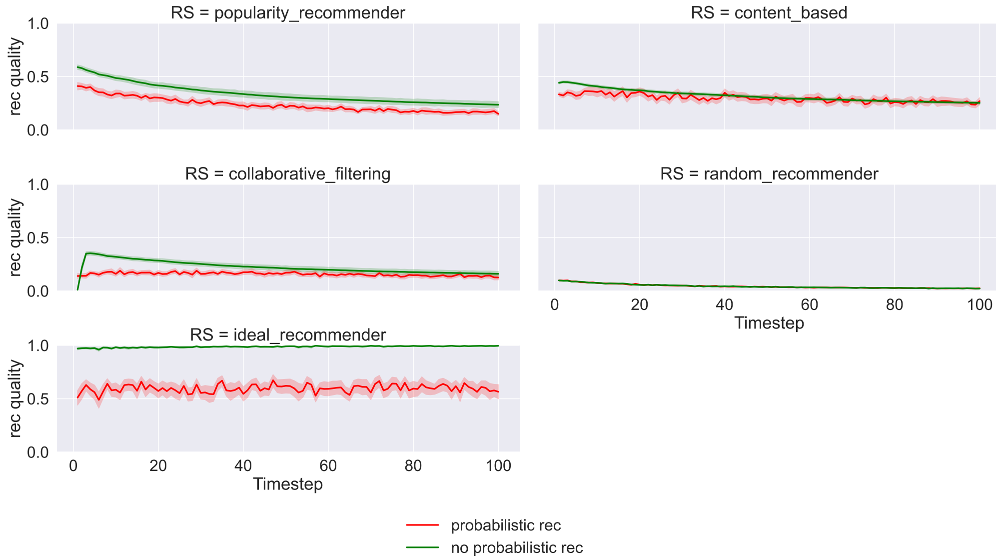

Figure C5: Quality of the recommendation - by probabilistic recommendation (rational users).

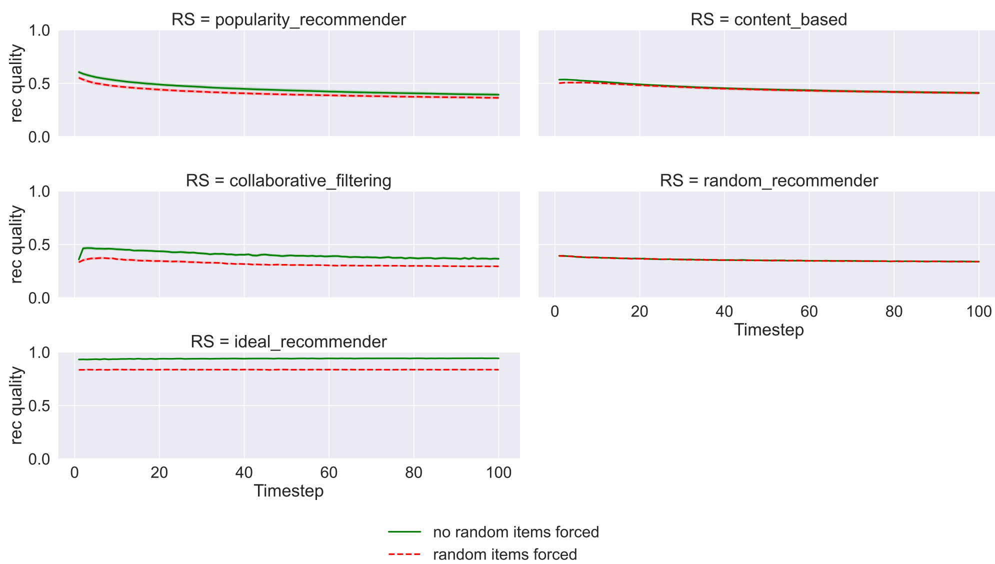

Figure C6: Quality of the recommendation - by forced random recommendations (rational users).

C.4 Impact of swaying tastes

Figure C7: The impact of swaying user tastes - Market share of 1% most popular items (limited attention users).

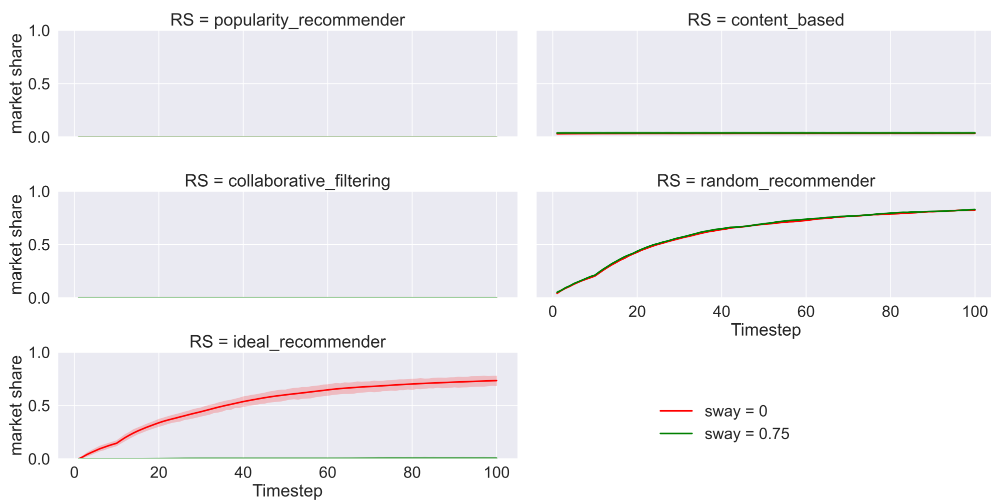

Figure C8: The impact of swaying user tastes - Market share of new entrant items (limited attention users).