PRODUCT RECOMMENDATIONS AND MARKET CONCENTRATION[†]

EMILIO CALVANO, GIACOMO CALZOLARI, AND VINCENZO DENICOLÒ AND SERGIO PASTORELLO[*]

SEPTEMBER 2021

PRELIMINARY AND INCOMPLETE
PLEASE DO NOT CIRCULATE WITHOUT AUTHORS' PERMISSION.

We explore the impact of algorithms that use personal data to predict tastes and make recommendations, on consumption choices and in particular on the level of market concentration. We model consumer preferences in a flexible way that allows for varying degrees of vertical and horizontal product differentiation. Like previous analyses, we find that recommender systems produce a significant degree of product market concentration. Contrary to the conventional wisdom however, we show that such increase in market concentration is not due to the "feedback loop" created by the endogeneity of the data. We discuss the implications of our findings for the intensity of competition and competition policy.

Keywords: Artificial Intelligence, Recommender Systems, Market Concentration, Product Differentiation.

J.E.L. codes: L41, L13, D43, D83.

1. INTRODUCTION

To navigate through the sea of products, consumers today use more sophisticated tools than they did in the past. Just as sailors have replaced the compass and the sextant by the GPS, consumers are replacing individual search and the word of mouth by algorithms, called *recommender systems* (hereafter RSs), which use personal data to predict tastes and filter content. Because these algorithms can help consumers discover new products and find good matches to their tastes, they are believed to boost consumer engagement and sales. That is the reason why RSs are extensively used by online businesses such as Netflix, Spotify, Amazon, Facebook and many others. Upon logging on these platforms, consumers are indeed greeted with a tailored selection of items from the catalogue that they might enjoy or need.[1]

Evidently, RSs affect consumption choices and thus have the potential to change the structure of markets and the competition among the firms. From an industrial organization perspective, these developments therefore raise important and yet unresolved questions. Will markets become more or less competitive as consumers increasingly resort to RSs? Will incumbents become more powerful, or will it be easier for new entrants to challenge their dominance?

Little is known on these issues, and the existing views are often contradictory. For us economists, it is tempting to conceptualize RSs as means for reducing the cost of individual search – a conceptualization that suggests that RSs may increase the elasticity of demand and thus may have pro-competitive effects. The concerns voiced by computer scientists are rather opposite, though. They have argued that RSs may generate a "rich get richer" dynamics whereby a handful of popular items are increasingly favored over time.[2] This allegedly results in political polarization in social media, insufficiently personalized news diets, or loss of diversity in culture. In economics, the risk is that markets may become more highly concentrated.

To shed light on this debate, in this paper we borrow elements from both the economics and the computer science literature. From the computer science literature, we borrow the insight that RSs change the process of information acquisition qualitatively. That is, RSs collect information from a large number of consumers and combine this information by exploiting the estimated similarities and dissimilarities among consumers. This process

[1] RSs are applied in many different environments, such as recommending news to users in social networks, financial products to investors (Robo-advising), job offers to job seekers, and possible referees to editors of academic journals and new articles to read to scholars. For an assessment of the business value of RSs see Jannach and Jugovac (2019).

[2] See, among many others, Fleder and Hosanagar (2009) and, more recently, Ciampaglia et al. (2018) and Goldfarb and Tucker (2019).

may not be equivalent to each consumer individually searching more extensively. Trying to predict the economic consequences of RSs by mechanically applying models of individual search may therefore be misleading. In particular, if consumers searched more extensively but still individually, their choices would presumably better reflect the idiosyncrasies of their personal tastes. We should therefore observe more diversity, not less.

From the economics literature, we import a consistent model of consumer preferences and product characteristics that can generate both vertical and horizontal product differentiation. We also borrow meaningful benchmarks and criteria to assess the economic consequences of RSs.

We have three main sets of results. First, we confirm computer scientists' claim that RSs produce an increase in market concentration. In other words, taking product characteristics and prices as given, the market is more highly concentrated when consumers follow the recommendations of RSs than if they rely on individual search. This substantiates the concern that viewing RSs merely as devices to reduce the search cost may be misleading.

Second, we show that the increase in market concentration is only partly due to the standard explanation proposed in the computer science literature. According to this explanation, the increase in concentration is due to the endogeneity of the data used to feed the RSs – the so called "feedback loop" effect. RSs, so the story goes, initially identify a few "good" products and recommend them to many consumers because they overestimate the similarity of their tastes. As time passes, RSs keep recommending the products they have recommended in the past because they acquire better information precisely on them. This reinforcement process amplifies small differences in products' perceived quality. Thus, RSs create their own "champions" almost from scratch. By way of contrast, we show that market concentration increases significantly even if the data used to feed the RSs are generated in a purely random manner rather than being endogenously produced by the RS itself. The additional contribution of the feedback-loop effect to the increase in concentration is relatively small.

The source of the increase in market concentration is important because it sheds light on the possible effects of such an increase. Generally speaking, high levels of market concentration may signal the weakness of competition when few large firms posses high market power. However, they may also reflect the selection effect of competition: in highly competitive markets, that is to say, small asymmetries in the quality-adjusted price may translate into large asymmetries in market shares. The feedback-loop narrative suggests that RSs may create market power where there would be none, or they may artificially amplify small initial competitive advantages, possibly leading to an entrenchment of monopoly. The fact that the feedback-loop effect in itself has only a modest impact on market con-

centration suggests a more benign interpretation of the competitive effects of RSs.

To analyze these effects more directly, we have therefore allowed firms to change the price of their products. We show, and this is our third result, that RSs generally lead to lower prices and higher consumer surplus. The effect is quite robust and is obtained for different combinations of vertical and horizontal differentiation. In particular, when vertical differentiation prevails, the adoption of RSs increases the market share of the dominant products but at the same time reduces their prices.

After establishing these results, we inquire into the underlying mechanism. In particular, our analysis shows that the feedback loop is not the main cause of the increase in market concentration, but this raises the problem of what are, then, the true causes. We rule out by design two other possible candidates. The first one is that platforms may deliberately manipulate recommendations so as to favor certain products (e.g., those with the highest margins, or their own products) and damage others. In our analysis, we use "neutral" RSs that are not instructed to support or penalize specific products. The second possible candidate is that RSs basically estimate a mis-specificed model of consumer preferences. But the result holds even if the RSs estimate a correctly specified model of consumer preferences.

Still, RSs make mistakes in estimating consumers' true preferences, for various reasons. First and foremost, RSs can observe only a subset of all possible consumer/product pairs, and typically a very small one. Second, even for those pairs, RSs may observe only a noisy signal of the true utility that the consumer obtains by consuming that product. The reason why these mistakes systematically produce levels of market concentration higher than in the individual search model is briefly discussed the last part of the paper.

2. LITERATURE REVIEW

The economic literature on RSs is limited. As noted, the most popular approach is to conceptualize RSs as tools that reduce search costs (Goldfarb and Tucker, 2019). To the extent that this interpretation is appropriate, RSs may allow consumers to discover rare or niche products, leading to a long-tail effect (Anderson 2008, Brynjolfsson et al. 2011). At the same time, low search cost can lead to a superstar effect (Rosen 1981), i.e., the discovery of superior products that attract more consumers. Bar-Isaac, Caruana, and Cunat (2012) show that the long-tail effect and the superstar effect may coexist when products are both vertically and horizontally differentiated, as they are in our model. However, RSs do not simply reduce search costs; as discussed in the introduction, they change the process of information acquisition qualitatively.

Another strand of the economics literature studies platforms' incentives to manipulate recommendations and steer consumer decisions toward certain products. In this literature, which includes Hagiu and Jullien (2011), Calvano and Jullien (2018) and Teh and Wright (2020), among others, RSs are modeled in a highly stylized way. Our analysis focuses on more complex and realistic RSs but assumes that all products are treated fairly.

A more indirectly related literature studies the effect of improved information about the distribution of product-match utility as perceived by consumers and the firms' price reaction (see e.g. Zhou, 2020 and Zhong, 2018). The possible connection with RSs comes from the fact that the informational improvement could be interpreted as the consequence of the adoption of RSs. However, the connection is loose because in those papers the change in the structure information is taken as exogenous and does not reflect the specificities of RSs.

The relevant computer science literature is much larger: see, for instance, Abdollahpouri et al. (2021) for a recent survey. This literature has identified a "popularity bias" of common RSs, particularly the ones based on collaborative filtering (as those considered in this paper). This popularity bias is the tendency of algorithms to disproportionately recommend popular items and disfavor others in spite of the fact that a significant number of users would actually prefer the latter as they better fit their idiosyncratic preferences.[3]

The computer science literature has categorized the possible sources of the popularity bias as "bias in the data" and "bias in the algorithms" (Abdollahpouri and Mansoury, 2020). The bias in the data is the fact that many users may intrinsically prefer a few popular items. As economists, we would not regard this as a bias but rather as a feature of consumers' true preferences, possibly reflecting vertical product differentiation.

The bias in the algorithms is more interesting and is generally attributed to the interaction between algorithms and data. It has been noted that this interaction may produce a feedback loop, as the algorithms are periodically retrained on the data that they have contributed to generate through their past recommendations (see e.g. Felder and Hosanagar, 2009; Chaney et al., 2018; Lee and Hosanagar, 2019; and Abdollahpouri and Mansoury, 2020).[4] The conventional wisdom is that this feedback loop induces homogenization of

[3]Most analyses identify this bias by means of numerical simulations, as we do. However, Ciampaglia et al. (2018) demonstrates this bias theoretically in a stylized model where, depending on the users' cost of exploration, algorithms that recommend popular items may hinder the discovery of high-quality items.

[4]To identify the bias in the algorithms, the computer science literature typically compares two cases, a benchmark case where the RS is trained only once, and the case where the RS is periodically retrained over new, endogenously generated data. As described in greater detail below, our procedure is different: the algorithm is retrained in each period in both cases, but in the exogenous data case the new data is generated randomly, whereas in the endogenous data case it is created by the RS's past recommendations.

consumer decisions and increases the concentration in the distribution of product-market shares (Zhang et al. 2020). As noted, one contribution of this paper is to challenge this conventional wisdom.

3. MODEL-BASED RECOMMENDER SYSTEMS

It would be interesting to study the RSs actually used by Amazon, Netflix, and the like, but the design of these RSs is confidential. For this reason we focus on algorithms that are prominent in the computer science and marketing literatures, that are the building blocks for the more sophisticated versions in production.

In particular, we concentrate on latent-factor collaborative filtering methods, also known as model-based systems.[5] There are several reasons for this choice. First, model-based models are, indeed, popular in computer science and are "considered to be state-of-the art in recommender systems" in specialized textbooks (Aggarwal 2016, p 91). Second, these tools are consistent with structural econometric modeling, as discussed in greater detail below. Third, model-based collaborative filtering models are relatively simple and transparent, which facilitates the analysis of the ultimate drivers of observed outcomes.

3.1. *Basics*

Consider a finite set of users $i = 1, 2, ..., I$ who have preferences over a finite set of items $j = 1, 2, ...J$. The preferences are represented by real numbers $\{r_{ij}\}$, one for each user-item pair. User i prefers item j over item k if $r_{ij} \geq r_{ik}$. In the jargon of computer science, r_{ij} is i's "rating" of item j. In economics, it is natural to interpret r_{ij} as the net utility that consumer i obtains from product j.

In its simplest formulation, a RS's problem is to predict user ratings $\{r_{ij}\}$ based on a subset of imperfectly observed ratings. The problem is typically formulated as a "matrix completion" problem as follows. Let $\tilde{\mathbf{R}}$ denote a sparse $I \times J$ matrix whose entries are the (possibly imperfectly) observed ratings. The goal of the RS is to estimate the missing entries, completing the matrix $\mathbf{R}$.

3.2. *Latent factors*

Broadly speaking, the simple idea behind the design of collaborative filtering RSs is to exploit row and column correlations of the observed rating matrix $\tilde{\mathbf{R}}$. The implicit assumption is that users whose observed ratings are correlated are likely to also rate the

[5]For a thorough treatment of RSs in computer science, see **?**.

unobserved items in a similar way. Likewise, items whose ratings are correlated across observed users are likely to be rated similarly by the unobserved ones.

This simple idea may be articulated and implemented in different ways. In this paper, we focus on a specific class of RSs that build on that idea, namely, latent-factor RSs. These models posit a specific structure of the ratings, i.e., a relatively small set of latent variables, whose goal is to parsimoniously explain the observed ratings and predict the unobserved ones.

Specifically, the rating of item j by user i is viewed as the inner product of a vector of user-specific factors t_{ih} and a vector of item-specific factors v_{jh}:

$$(1) \qquad r_{ij} = \sum_{h=1}^{K} t_{ih} v_{jh},$$

where K is the number of latent factors. In matrix notation:

$$(2) \qquad \mathbf{R} = \mathbf{T}\mathbf{V}',$$

where $\mathbf{T}$ and $\mathbf{V}$ are $(I \times K)$ and $(J \times K)$ matrices, respectively.

The variable t_{ih} may be thought of as user i's proclivity for factor h, and the variable v_{jh} as product j's affinity to factor h. In applications, the K factors typically have a semantic interpretation. For the algorithms, however, they need not have any specific meaning.

The algorithm chooses the number of factors $\hat{K}$ and the parameters $\hat{\mathbf{T}}$ and $\hat{\mathbf{V}}$ so as to minimize some measure of the distance between the predicted ratings,

$$(3) \qquad \hat{r}_{ij} := \sum_{h=1}^{\hat{K}} \hat{t}_{ih} \hat{v}_{jh}$$

and the observed ones, $\tilde{r}_{ij}$.

The latent factor model that is the basis of our RSs is consistent with the Random Utility Model that underpins much of the structural empirical work on demand estimation. The main difference is that in econometric estimations, it is typically assumed that the choice attributes (the v_{jh}s) are observable while preferences (the t_{jh}s) are not. RSs, in contrast, assume that neither product attributes nor consumers tastes are observable. Thus, RSs *jointly* estimate both attributes and tastes.

3.3. *Estimation procedure*

Let S denote the set of all pairs (i, j) for which r_{ij} is (at least imperfectly) observed. For given K, the problem of the RS is to minimize a measure of prediction error on observables. In its simplest incarnation, the estimated parameters $\hat{\mathbf{T}}$ and $\hat{\mathbf{V}}$ solve:

$$\min_{\hat{\mathbf{T}},\hat{\mathbf{V}}} \sum_{ij\in S} \left(\tilde{r}_{i,j} - \sum_{h=1}^{K} \hat{t}_{ih}\hat{v}_{jh} \right)^2 . \tag{4}$$

Once the latent factors are estimated, one can reconstruct the entire matrix of estimated ratings, $\hat{\mathbf{R}}$, by simply using the estimator $\hat{\mathbf{R}} = \hat{\mathbf{T}}\hat{\mathbf{V}}'$. With this matrix at hand, the algorithis then able to rank all the items for each users and make recommendations.

In real-life applications, the set S of observed ratings tends to be small; that is, the rating matrix $\tilde{\mathbf{R}}$ is very sparse. In this case, the estimation procedure (4) may cause overfitting. A common approach for addressing this problem is to use regularization techniques.[6] The idea is to penalize large values of the coefficients in the matrices $\hat{\mathbf{T}}$ and $\hat{\mathbf{V}}$. This decreases the variance of $\hat{\mathbf{R}}$, at the cost of introducing some bias. The actual minimization problem then becomes

$$\min_{\hat{\mathbf{T}},\hat{\mathbf{V}}} \sum_{ij\in S} (\tilde{r}_{i,j} - \sum_{h=1}^{K} \hat{t}_{ih}\hat{v}_{jh})^2 + \lambda_t \sum_{i=1}^{I}\sum_{h=1}^{K} \hat{t}_{ih}^2 + \lambda_v \sum_{j=1}^{J}\sum_{h=1}^{K} \hat{v}_{ih}^2, \tag{5}$$

where the non-negative parameters λ_t and λ_v are the regularization weights, one of the hyper-parameters of the algorithm.

The minimization of the loss function proceeds by calculating the gradient vector of partial derivatives (w.r.t the $J \times K + I \times K$ decision variables) and using numerical methods to find values of $\hat{\mathbf{T}}, \hat{\mathbf{V}}$ such that the gradient is nil. In vector notation, the gradient is

$$\mathbf{G} = \begin{pmatrix} \mathbf{E}\hat{\mathbf{V}} - \lambda_t\hat{\mathbf{T}} \\ \mathbf{E}'\hat{\mathbf{T}} - \lambda_v\hat{\mathbf{V}} \end{pmatrix} \tag{6}$$

where $\mathbf{E}$ is the $I \times J$ matrix of residuals $e_{ij} = \tilde{r}_{i,j} - \sum_{h=1}^{\hat{K}} \hat{t}_{ih}\hat{v}_{jh}$. The numerical method used to find $\hat{\mathbf{T}}, \hat{\mathbf{V}}$ such that $\mathbf{G} = 0$ is, typically, some version of the Gradient Descent algorithm. The idea is to start with some initial values for $\hat{\mathbf{T}}_0, \hat{\mathbf{V}}_0$ and the associated $\mathbf{G}_0$

[6] This approach is used not only for RSs but also in many other machine learning algorithms.

and iterate according to the following recursive equation

$$\begin{pmatrix}\mathbf{T}_{t+1}\\ \mathbf{V}_{t+1}\end{pmatrix} = \begin{pmatrix}\mathbf{T}_{t}\\ \mathbf{V}_{t}\end{pmatrix} - \eta\mathbf{G}_t \tag{7}$$

where $\eta > 0$ is the step size and $\mathbf{T}_{t+1}, \mathbf{V}_{t+1}$ are the new values of the variables of interest at the end of the iteration. The iterations are repeated to convergence.

From a computational viewpoint, this estimation procedure, which is generally referred to as matrix factorization, comes in many different variants. For our simulations, we use a publicly available, state-of-the-art library (Chin et al. 2016) designed for parallelized computation.

4. EXPERIMENT DESIGN

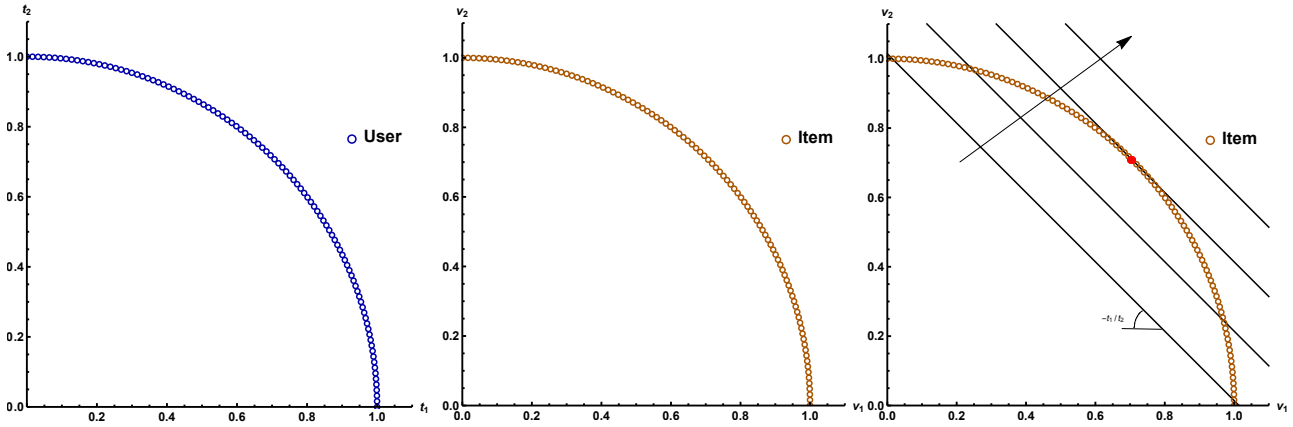
We use the collaborative filtering RSs described in the previous section to predict ratings in an artificial economic environment. The economic environment comprises the set of products, the consumers' preferences over these products, and the way consumption choices are made, new information is produced and fed into the RSs.

We refer to a combination of preferences, product characteristics, hyper-parameters of the algorithm, and mode of generating the data that are fed into the algorithm as an *experiment*. Since the data are generated randomly, as described in greater detail presently, for each experiment we run a number of independent *sessions*. We then average the results across sessions so as to eliminate the noise.

Each session lasts for T periods. In period 1, each user i is randomly matched with one item j. The rating $\tilde{r}_{ij}$, which includes a random shock, is then observed. (The size of the shocks is a parameter that we control.) As for the subsequent periods, we consider two different scenarios. In the first scenario (*exogenous data*), users and items are randomly re-matched in each period $\tau = 2, 3, ..., T$.[7] In the second scenario (*endogenous data*), in each period $\tau = 2, 3, ..., T$ the user follows the personal recommendation of the RS and consumes the recommended item. In both cases, the new ratings associated with the new products consumed are observed and the estimated rating matrix $\hat{\mathbf{R}}_\tau$ is updated accordingly. As time passes, the matrix of observed rating $\tilde{\mathbf{R}}_\tau$ becomes less and less sparse in both scenarios, but at a different speed.

We then compare the choices made with the assistance of the RS with those made by consumers who search individually.

[7] If the same user-item pair is observed in more than one period, the matrix $\tilde{\mathbf{R}}_\tau$ records only the last observation of r_{ij}, overwriting previous entries.



(a) User location (b) Item location (c) User $t_1 = t_2$ indifference map and optimal item

Figure 1: Distribution of user preferences

4.1. *Preferences and products*

Consumer's preferences and product characteristics are taken to be consistent with the latent factor model used by the RS. They are modelled in a way flexible enough to accommodate the classic notions of horizontal and vertical product differentiation used in the industrial organization literature.

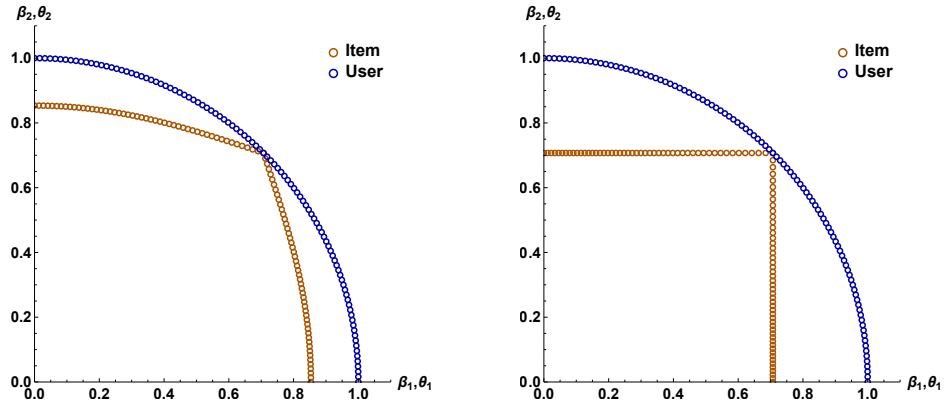
In our baseline setting, we consider $I = 99$ consumers, $J = 99$ products,[8] and $K = 2$ factors. Thus, preferences of consumer i over product j at time τ are given by

$$u_{ij\tau} = t_{i1}v_{j1} + t_{i2}v_{j2} + \varepsilon_{ij\tau}, \tag{8}$$

where $\varepsilon_{ij\tau}$ are idiosyncratic time-varying taste shocks that follow a Gumbel distribution with zero mean, $t_i := (t_{i1}, t_{i2})$ captures consumer i's preferences over the $K = 2$ factors (or choice characteristics, to use the discrete choice literature jargon) and $v_j := (v_{j1}, v_{j2})$ captures product j's affinity with those factors. In the baseline case, we set the standard deviation of the shocks equal to one tenth of the standard deviation of each user's expected utilities across the 99 products.

In all cases, consumers $(t_{i1}, t_{i2}) \in \mathbb{R}^2$ are assumed to be equally spaced on the portion

[8]The fact that the number of consumers equals the number of products may sound implausible, as there typically are more consumers than products. The case $I > J$ is considered among the extensions. Keep in mind, however, that we assume that all consumers report their ratings in all periods. In practice, only a fraction of them does. Thus, the model is equivalent to one where there are I consumer types, and there are m consumers of each type, but each consumer has only a probability $\frac{1}{m}$ of reporting his or her rating to the RS in each period.



(a) Mixed differentiation (b) Pure vertical differentiation

Figure 2: Item differentiation

of the unit circle lying in the positive orthant (figure 1, center).[9] As for products, their location determines whether products are horizontally or vertically differentiated.

With pure horizontal differentiation, products are placed on the same locus as consumers, and they are equally spaced as well (figure 1, left). By construction, a consumer with coordinates (t_{i1}, t_{i2}) would then strictly prefer the product with the same coordinates, as indifference curves are straight lines with slope $-\frac{t_{i1}}{t_{i2}}$. The consumer's utility from more distant products would be a decreasing function of the Euclidean distance between his or her location and that of the product. To illustrate, Figure 1 (right panel) depicts the map of indifference curves of a consumer with $t_1 = t_2$ and shows that in this case the optimal product is the one with $v_1 = v_2$. In this model of pure horizontal differentiation, the surplus maximizing allocation is where each product is chosen by exactly one consumer. As a result, all products should have the same market share, equal to 1/99 (i.e., slightly above 1%).

Under pure vertical differentiation, on the other hand, products are placed on the square $[0, \frac{1}{\sqrt{2}}]^2$ (Figure 2, right panel). As a result, all consumers agree on their favorite product, which is the one that lies on the 45° line. This product will be referred to as the central, or dominant product.

Intermediate cases between purely horizontal and purely vertical product differentiation are obtained by varying the shape of the product locus smoothly, from a circle to a square. This can be done by means of a parameter, α, that ranges from $\alpha = 0$ (circle, horizontal differentiation) to $\alpha = 1$ (square, product differentiation) (Figure 2 left). To be precise, for

[9] One of the consumers is located on the 45 degree line and half of the remaining consumers is located at equal distance on the left and on the right arc.

each product j on the circle, i.e. $v_j = (v_{j1}, v_{j2})$, define its radial projection on the square as $v'_j = \frac{1}{\sqrt{2}} \left(\frac{v_{j1}}{\max\{v_{j1}, v_{j2}\}}, \frac{v_{j2}}{\max\{v_{j1}, v_{j2}\}} \right)$. By taking the convex combination $\alpha v_j + (1 - \alpha) v'_j$ one can then obtain all intermediate degrees of vertical/horizontal differentiation.

When $\alpha > 0$, a bunch of consumers (i.e., those closest to the 45° line) will prefer the central product, whereas consumers closer to the axes will prefer more peripheral products. Products close to but different from the central one would not be chosen by any consumer. As α increases, the bunch of consumers who prefer the central product gets larger. So also does the set of products, close to but different from the central one, that are not chosen by any consumer.

4.2. *Experimental protocol*

Each session consists of $T = 50$ periods indexed by τ. The matrix of ratings evolves over periods as follows. At $\tau = 0$, the matrix $\tilde{\mathbf{R}}_0$ is initialized as an empty matrix (that is, a matrix with all entries missing). At the beginning of each of the following periods, we construct $\tilde{\mathbf{R}}_\tau$ adding one and only one entry $\tilde{r}_{ij}$ for each of the I rows (i.e. for each consumer), chosen as described below. The entry is equal to the utility that i obtains from item j at period τ, i.e. $u_{ij\tau}$.[10] If $\tilde{r}_{ij}$ is not missing in $\tilde{\mathbf{R}}_{\tau-1}$, its value is overwritten. All other elements of each row of $\tilde{\mathbf{R}}_{\tau-1}$ are unchanged. Thus, $\tilde{\mathbf{R}}_\tau$ is basically equal to $\tilde{\mathbf{R}}_{\tau-1}$ plus one new rating per consumer, which could be either a previous value overwritten or a brand new one.

Intuitively, at the beginning of every period each user is matched with a product and truthfully reports the realized utility from consumption. As time passes, the sparsity of $\tilde{\mathbf{R}}_\tau$ mechanically decreases. At each τ, $\tilde{\mathbf{R}}_\tau$ is then fed into the algorithm to produce $\hat{\mathbf{R}}_\tau$ according to the estimation procedure described above. We are interested in the evolution of the recommendations implied by $\hat{\mathbf{R}}_\tau$.

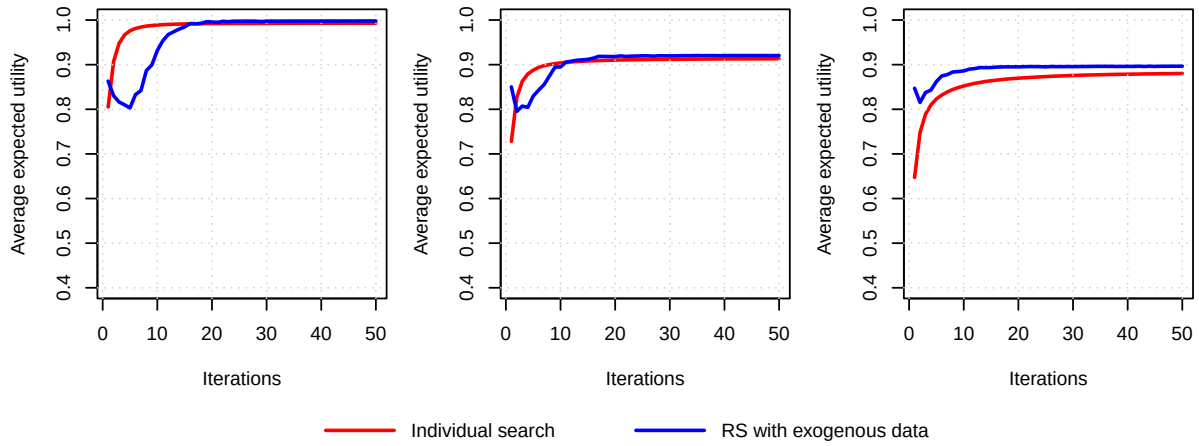
4.3. *New data and matrix update*

As for the new data that each period enter the rating matrix $\tilde{\mathbf{R}}_\tau$, we consider two cases.

With *exogenous data*, for each consumer(row) i we randomly draw a product (column) j among the J products (with equal probability) and then add (or over-write) $\tilde{r}_{ij} = u_{ij\tau}$ to the cell i, j of the rating matrix $\tilde{\mathbf{R}}_\tau$.

[10] We will consider environments in which the rating matrix contains other types of assessments of the user-product match, for example 1 if the user consumed the item and 0 otherwise, or some rating of the matching on a discrete scale.

Figure 3: Average expected (normalized) utility



Horizontal (left), Vertical (right) and Mixed (center) differentiation

With *endogenous data*, we populate $\tilde{\mathbf{R}}_\tau$ with the recommendations made by the RS itself. Specifically, let $j^\star(i,\tau)$ denote the product j with the best predicted rating for consumer i in period τ, $j^\star(i,\tau) = \max_j\{\hat{r}_{ij\tau}\}$. The cell that we populate (or overwrite) at period τ is then $i, j^\star(i,\tau)$ and the new rating is $\tilde{r}_{ij^\star} = u_{ij^\star\tau}$.

4.4. *Individual search benchmark*

In the individual search benchmark, each consumer can rely only on his own search and past experience. Specifically, at the beginning of every period τ, each individual consumer i randomly draws an item j and observes $u_{ij\tau}$. The consumer recollects all past draws and then simply chooses the product with the highest observed utility.[11]

5. QUALITY OF CHOICES

As a preliminary consistency check, we ask whether RSs improve consumer choice. To this end, in this section we compare the RS's recommendations with the choices based on the individual search process described above.

To assess the quality of choices, we use the expected utility, averaged between users and normalized to a maximal value of 1. This measure is equal to 1 if the utility provided by the recommended item $j^\star(i,\tau)$ is the best product for user i. We then compute the average across users and sessions, for each period τ.

[11]Similarly to what we assume for the RS, when the same item is consumed more than once, only the last value of u_{ij} is recorded.

Figure 3 shows the performance of our RS relative to the benchmark, for three different values of parameter α, which measures the degree of vertical/horizontal differentiation: $\alpha = 0$ (pure horizontal), $\alpha = \frac{1}{2}$ and $\alpha = 1$ (pure vertical). In all three scenarios, the RS starts just above the median and eventually reaches a plateau at around the 95 % percentile. The individual search benchmark does better as long as the rating matrix is very sparse (≤ 10) but is then overcome by the RS.[12] After 20 periods or more, the RS improves the quality of choice by 5-6 percentiles.

Notice that the RS performs best under vertical product differentiation. This is intuitive as RSs exploit the similarities across consumers, which are large when $\alpha = 1$ but do not exist when $\alpha = 0$.

6. MARKET CONCENTRATION

In this section, we confirm the finding of the computer science literature, that RSs raise market concentration when they are fed with the information produced by their own recommendations.

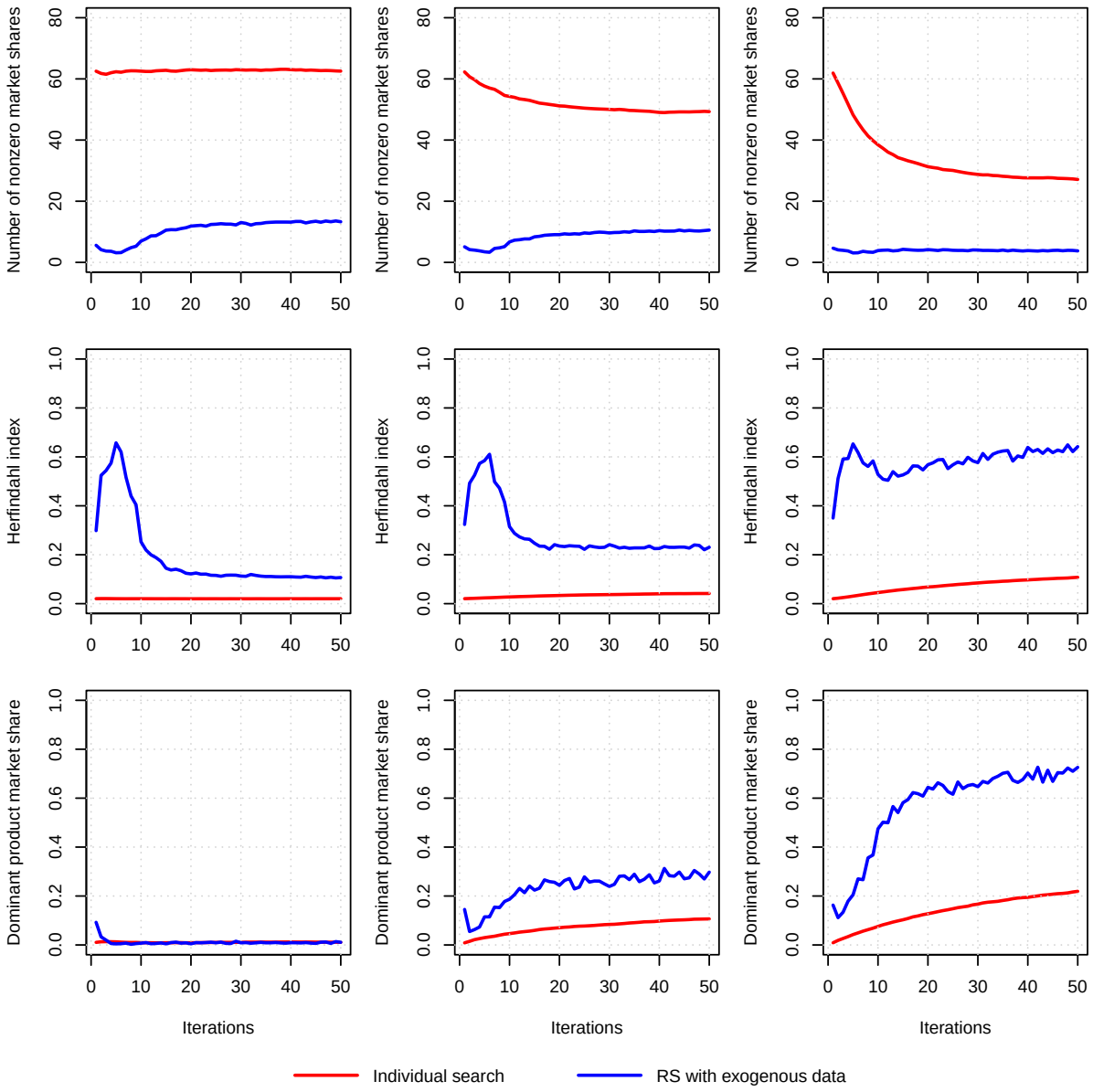
Figure 4 shows various measures of market concentration for different values of α, for the baseline case $K = 2$. The first row shows the number of products consumed by at least one consumer (i.e., the products with a positive market share). The second row plots the Hirsh-Herfindahl Index (HHI) (a standard measure of market concentration in industrial organization).[13] The final row of Figure 4 shows the market share of the central, or dominant product.

The first measure of concentration captures the diversity of consumer choice. For perfectly informed consumers, the number of positive market share would be 99 for $\alpha = 0$ and 1 for $\alpha = 1$. Naturally, the RS lies below the theoretical benchmark for $\alpha = 0$ and above it for $\alpha = 1$, but in any case it lies significantly below the individual search benchmark. In other words, RSs concentrate their recommendations on a much smaller set of products than in the individual search benchmark.

On the other hand, consider the market share of the central product. This product plays no special role when $\alpha = 0$ but is the one that would be chosen, under complete information, by the largest number of consumers when $\alpha > 0$, and by all consumers when $\alpha = 1$. When $\alpha = 0$, neither RSs nor the individual search process, correctly, attach any prominence to the central product. With pure vertical differentiation, however, RSs give the dominant

[12] Nevertheless, the RSs' "cold start" is noteworthy. We conjecture that it is due to the mechanism of regularization described above, but other explanations are possible.

[13] A similar pattern is exhibited by the Gini index, another standard measure of concentration.



Horizontal (left), Mixed (center) and Vertical (right) differentiation

Figure 4: Concentration measures

product a market share of about 75%, whereas under individual search it would be slightly above 20%. RSs, that is to say, allow consumers to identify with much greater precision a superior product when it exists.

Summarizing, RSs raise concentration significantly compared to the individual search benchmark. The same pattern is observed with a greater number of factors (i.e., $K = 5$ or $K = 10$).

7. CAUSES OF HIGHER CONCENTRATION

What is the source of the increase in market concentration? The computer science literature has argued that it is the feedback-loop channel, i.e., the fact that the algorithms are fed with the ratings of the products that they themselves have recommended. In this section, we show that in fact the increase in concentration is observed even when the data fed into the RS are produced randomly, so that the feedback-loop channel is muted.

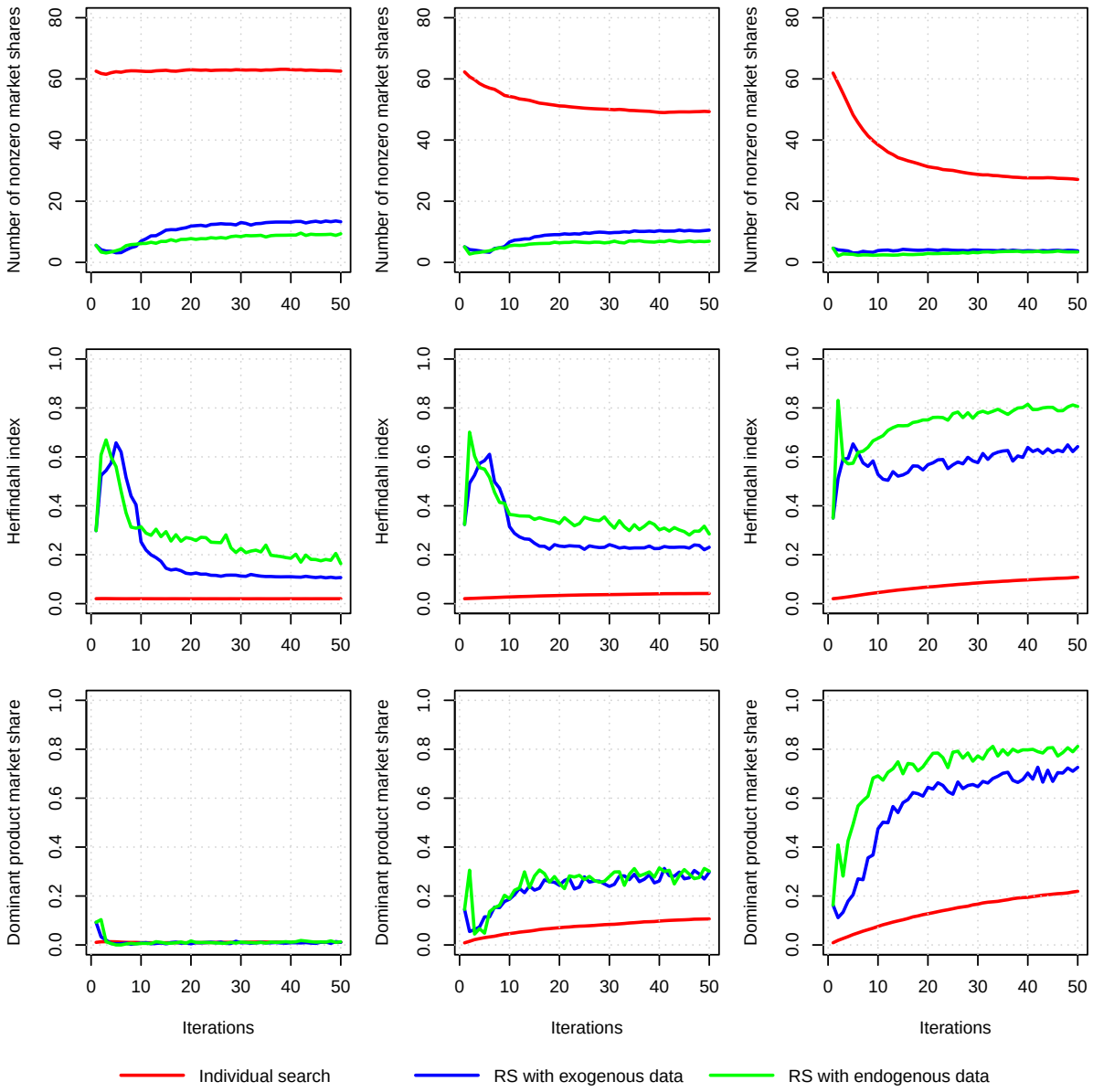
Figure 5 plots the different measures of market concentration for the case of endogenous and exogenous data. It shows that market concentration increases, with respect to the individual search benchmark, even with exogenous data. In fact, the effect of data endogeneity is small compared to the overall effect of RSs on market concentration. For example, the market share of the dominant product with pure vertical differentiation increases from about 75% to just above 80%. By comparison, the share would be around 20% with individual search.

In sum, the endogeneity of data is not the key reason why RSs lead to more market concentration.

8. CONSEQUENCES OF HIGHER CONCENTRATION

As noted in the introduction, an increase in the level of market concentration may be associated with different effects of RSs on the intensity of competition. On the one hand, concentration is high when few large firms possess significant market power. On the other hand, concentration is also high when the market is very competitive and superior products are purchased by many consumers.

The feedback-loop narrative suggests that RSs may create market power where there would be none, or they may artificially amplify small initial competitive advantages. The fact that the increase in market concentration is not due to the feedback-loop effect, however, suggests the possibility that the competitive effects of RSs may be more benign. However, it remains to be seen how RSs actually affect the intensity of competition.



Horizontal (left), Mixed (center) and Vertical (right) differentiation

Figure 5: Concentration measures with and without endogenous data

To this end, in this section we endogenize firms' pricing choices and analyze the impact of RSs on pricing. We assume that each product is supplied by a separate profit maximizing firm. For simplicity, we normalize marginal production costs to zero. We assume that all consumers have the same marginal utility of income (normalized to one),[14] so consumer i' net utility from consuming product j is

$$u_{ij\tau} - p_{j\tau}.$$

Prices are observed by the RS. The algorithm's learning process follows the same steps as in previous sections, but now the assessment of the different products relies on the estimated ratings *net of prices*. The recommended product is thus $j^\star(i,\tau) = \max_j\{\hat{r}_{ij\tau} - p_{j\tau}\}$. The same changes apply to the individual search benchmark.[15]

When setting prices, firms anticipate the demand implied by the RS's recommendation, or the individual search. To obtain smooth demand functions,[16] we assume that firms anticipate that the probability that consumer i will purchase product j in period τ is

$$q_{ijt} = \frac{e^{\left(\frac{\hat{r}_{ij\tau}-p_{j\tau}}{\sigma_{ij\tau}}\right)}}{\sum_{j=1}^{J} e^{\left(\frac{\hat{r}_{ij\tau}-p_{j\tau}}{\sigma_{ij\tau}}\right)}}. \tag{9}$$

The interpretation is that firms know $\hat{\mathbf{R}}_\tau$ and believe that there are many incarnations of user i, each one associated with different realizations of an idiosyncratic shock $\tilde{\varepsilon}_{ij\tau}$, which follows a Gumbel distribution.[17] The aggregate demand function of product j is then obtained by summing the probabilities of selling to all I consumers.

Given the system of demand functions, we finally numerically calculate the static Nash equilibrium prices $\mathbf{p}^\star_\tau$,[18] for all the J products.[19] These hypothetical prices provide an

[14] We assume that RS knows the marginal utility of income. With heterogeneous marginal utilities of income, the RS should use the observable prices and estimate the unknown marginal willingness to pay of each user. This would require the adoption of a different type of RS, known as "Hybrid (or ensemble)-RS," which rely on some observable dimensions of items and users. We leave this extension for future work.

[15] That is, each user randomly draws an item j, observes both his or her the utility $u_{ij\tau}$ and the price $p_{j\tau}$, and makes his or her choices by maximizing the difference $u_{ij\tau} - p_{j\tau}$.

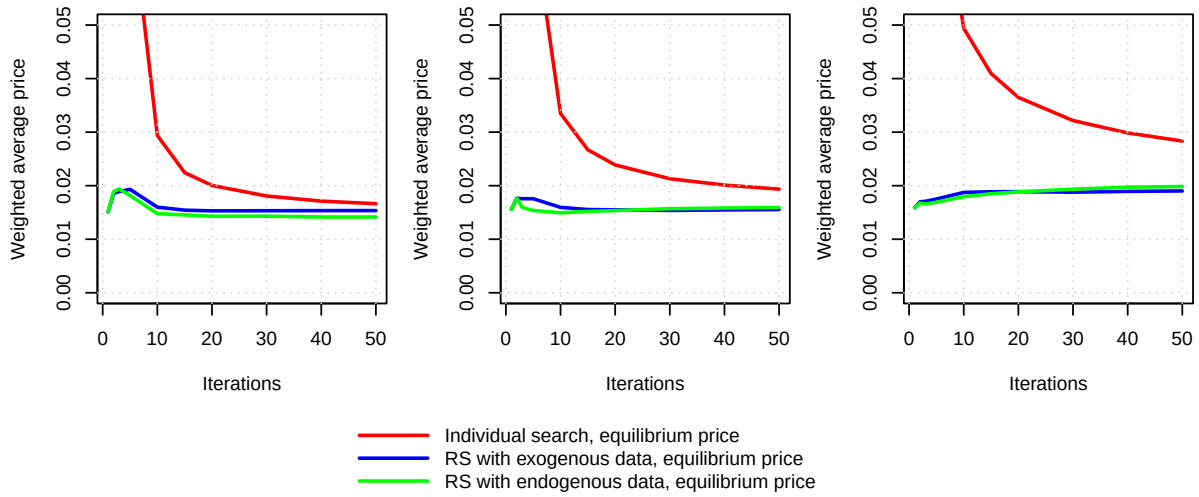
[16] One could directly derive a step-wise decreasing demand function from $\hat{\mathbf{R}}_\tau$. With these demand functions, however, profits would be discontinuous in prices, making the derivation of the Nash equilibrium difficult.

[17] Since these shocks are introduced only to smooth out demand, the size of the shocks is taken to be very small, in the order of $\sigma_{ij\tau} \simeq 0.01$. For the sake of consistency across different cases, the exact value depends on α.

[18] We do not consider the possibility that firms try to manipulate future recommendations, for example inflating consumption with low current prices.

[19] To this end, we use the Iterated-Best-Response algorithm (Gauss-Seidel): see Davis and Garces (2009)

Figure 6: Average weighted price



Horizontal (left), Vertical (right) and Mixed (center) differentiation

inverse index of the intensity of competition.

Figure 6 represents the average price[20] induced by RSs with exogenous and endogenous data and compares it to the prices that would arise in the individual search benchmark. Prices tend to be very low, reflecting the fact that the economic environment that we are studying is quite competitive, especially for low values of α. That said, it appears that with RSs, prices are lower than in the individual search benchmark. This shows that RSs actually intensify competition.

The effect of RSs on prices is stronger, the sparser the rating matrixes, and the higher α. The effect of α suggests that RSs are better at identifying superstar products than at discovering niche products. The effect of the sparsity of the matrix is natural: as the number of iterations increases, consumers become better informed in all cases, and thus the differences between the case with and without RS, and with different RSs, become smaller.

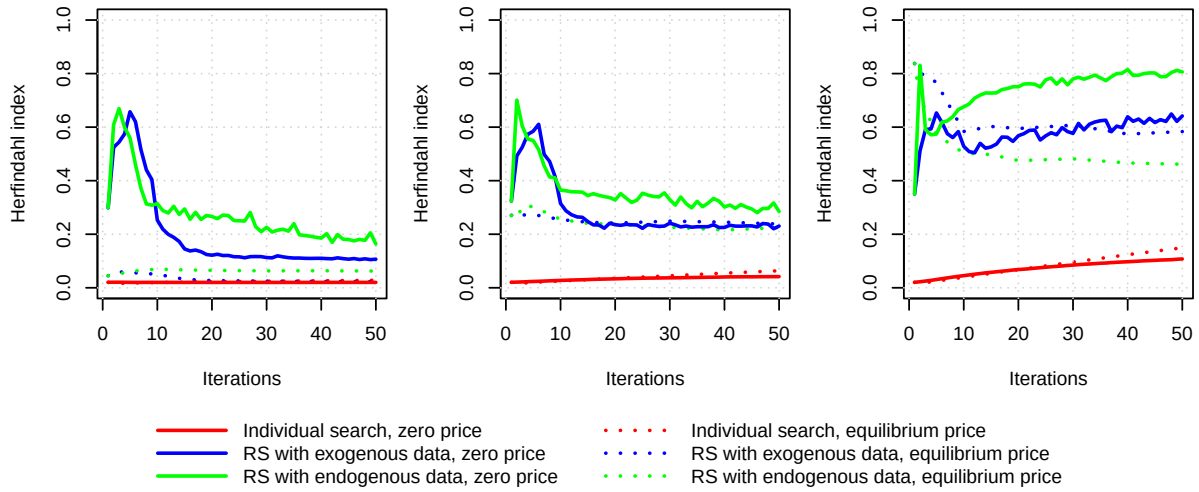
Note that when it comes to prices, the differences between the case of endogenous and exogenous data almost vanish. The impact of RSs on the intensity of competition, that is to say, is not due to the feedback-loop effect.

Figure 7 shows that with endogenous prices, market concentration decreases. This is

pp: 417-18 for details. Sometimes the procedure converges to limit cycles (which could be interpreted as Edgeworth cycles). In these cases, we take the mean of the firm's prices in the cycle. Cycles however are rare. They only occur in the individual search benchmark for a small number of iterations (i.e., with very sparse rating matrices).

[20] We use a weighted mean, with weights equal to the equilibrium market share of each product.

Figure 7: Herfindahl index with endogenous prices



Horizontal (left), Vertical (right) and Mixed (center) differentiation

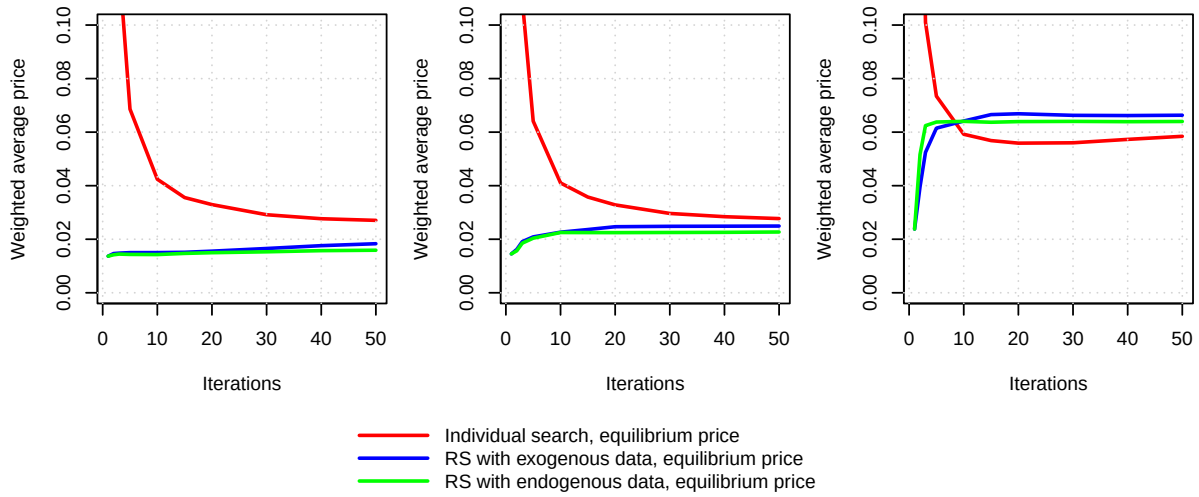
intuitive, as products that are favored by the RS will carry higher prices and this tends to reduce their market share.

Figure 8 shows the price effects of RSs when $K = 5$. The pattern that emerges is similar. One notable difference, however, is that when $\alpha = 1$ prices can become now higher with RSs than in the individual search benchmark (when sparsity reduces). However, this is entirely due to a composition effect. Prices are in fact higher in the individual search benchmark than with RSs, both for the dominant product and for the dominated products. However, the prices of dominated products are much lower than that of the dominant product, and in the individual search benchmark dominated products command a higher market share than with RSs. Another difference is that the price-reducing effect of RSs is more persistent than in the case $K = 2$.

Summarizing, RSs have a pro-competitive, price reducing effect. The increase in market concentration created by RSs is therefore a sign of intensified competition.

The analysis in this section has focused on the price as the relevant competitive variable. However, a similar logic applies to the case of quality improvements (e.g. by investment in R&D). RSs make it easier for consumers to identify better products and therefore amplify the positive demand effects of quality improvements. This increases firms' incentives to invest in R&D.

Figure 8: Average weighted price ($K = 5$)



Horizontal (left), Vertical (right) and Mixed (center) differentiation

9. MECHANISMS

In this section, we speculate on potential drivers of recommendation bias by looking at how the coefficients' estimates $\hat{\mathbf{T}}$ and $\hat{\mathbf{V}}$ compare with the data generating process. Concentration obviously reflects a systematic failure to recover the true parameters due to small and/or non random sampling.

To provide geometric intuition, figure 9 offers a handpicked example depicting one set of estimates for both consumers' and products' latent vectors $\{(\hat{t}_{i1}, \hat{t}_{i2})\}_{i=1}^{I}$ and $\{(\hat{v}_{i1}, \hat{v}_{i2})\}_{j=1}^{J}$ drawn for one particular session in period $\tau = 15$ under pure horizontal differentiation (and nil prices). $\hat{t}_i$ are basically 'virtual' consumers that are directed towards the 'virtual' products $\hat{v}_j$ that maximize their utility. The 99 green dots laying on the circle with unit radius represent the true location of consumers and products. The blue empty circles represent the 99 $\hat{t}_i$ estimates obtained by using the estimation procedure described in section 3.3 with input $\mathbf{R}_{\tau=14}$. The solid black and solid red disks depict the subset of the J estimates (one for each product) that are not dominated (a product is dominated if it exists another product that has higher estimated coordinates).[21] Red disks represent products which are *not* recommended to any consumer despite not being dominated. The larger orange disks around the solid black products have a radius proportional to the (virtual) products' market shares. The dashed lines stemming from the origin represent indifference loci. Consumers on these loci (if any) would be indifferent between different virtual products. Consumers lying between two loci, strictly prefer one particular product

[21] Dominated products are never recommended as they cannot possibly rank first.

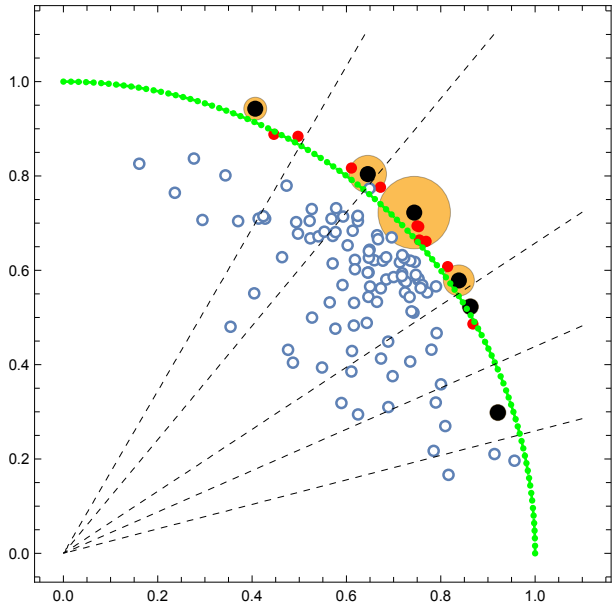


Figure 9: $\hat{t}$, $\hat{v}$ and market share of non-dominated products at $\tau = 15$

over the others. For example, the three virtual consumers in the bottom right 'slice', strictly prefer the product represented by the black dot at the lowest bottom right position (the market share is small and not depicted).

We distinguish at least three economically meaningful sources of failure. First, the algorithms may fail to identify consumers' tastes over factors by placing them 'too close' to a particular product. Geometrically this means that the estimated coefficients in the (t_1, t_2) space are bunched together. In the figure the product with the largest market share is strictly preferred by the large cloud of virtual consumers in the middle slice.

Even if the algorithm were to correctly estimate $\hat{\mathbf{T}} = \mathbf{T}$ thus placing virtual consumers on the unit circle, concentration would still occur if it were to systematically fail to estimate product characteristics $\hat{\mathbf{V}}$. This could occur in one of two ways: first, product placed outside of the circle (in the north east direction) would automatically attract more consumers (geometrically they would command larger 'slices'). These are products that would be incorrectly assigned a 'quality' advantage over rival products. Finally, even if all products were to lay on the circle so that no product would command a larger 'quality', if some products were more 'isolated' then others, this would lead to a larger market share.

10. CONCLUSIONS

In future work, we plan to conduct and extensive robustness analysis. Among the extensions that we plan to consider are:

- the case where the number of consumers and products are different;
- the case of misspecified algorithms (for example, the algorithms may use an exogenously given number of factors that is different from the true one, or the number of relevant factors could be estimated jointly with the $\mathbf{T}$ and $\mathbf{V}$ matrices, with the possibility of estimation errors);
- different models of vertical/horizontal product differentiation;
- different amounts of ratings for different users, which could allow to study the implication of users' data-portability.

REFERENCES

Abdollahpouri H. and M. Mansoury, (2020), Multi-sided Exposure Bias in Recommendation, International Workshop on Industrial Recommendation Systems with ACM KDD 2020.

Abdollahpouri H., M. Mansoury, R. Burke, B. Mobasher, E. C. Malthouse, (2021) User-centered Evaluation of Popularity Bias in Recommender Systems, arXiv:2103.06364.

Anderson C. (2008) The Long Tail: Why The Future Of Business Is Selling Less Of More (Hyperion Books, New York).

Bar-Isaac, H., G. Caruana, and V. Cunat (2012) Search, design, and market structure, American Economic Review 102 (2), 1140-60.

Brynjolfsson E, Hu YJ, Simester D (2011) Goodbye Pareto principle, hello long tail: The effect of search costs on the concentration of product sales. Management Sci. 57(8):1373-1386.

Calvano E. and B. Jullien (2018) Can we trust the algorithms that recommend products online? A theory of biased advice, wp.

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

Chin W.S., B.W. Yuan, M.Y Yang, Y. Zhuang, Y.C. Juan and C.J. Lin (2016) LIBMF: A Library for Parallel Matrix Factorization in Shared-memory Systems, Journal of Machine Learning Research 17: 1-5

Ciampaglia G.L. and A. Nematzadeh, F. Menczer, and A. Flammini, (2018) How algorithmic popularity bias hinders or promotes quality. Nature, Scientific reports 8, 1:17.

Davis, P., and E. Garces, (2009) Quantitative Techniques for Competition and Antitrust Analysis. Princeton: Princeton University.

Fleder D. and K. Hosanagar, (2009) Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity, Management Science, vol. 55, no. 5, pp. 697-712.

Goldfarb A. and C. Tucker (2019) Digital Economics, Journal of Economic Literature, 57(1), 3-43.

Hagiu, A., and B. Jullien (2011) Why do intermediaries divert search?, The RAND Journal of Economics, 42(2), 337-362.

Jannach D. and M. Jugovac (2019) Measuring the business value of recommender systems, ACM Transactions on Management Information Systems (TMIS), 10(4):1-23.

Lee D. and K. Hosanagar, (2019) How do recommender systems affect sales diversity? a cross-category investigation via randomized field experiment, Information Systems Research, vol. 30, no. 1, pp. 239-259.

Rosen, S. (1981). The Economics of Superstars, American Economic Review. 71: 845-858.

Teh, T.-H., and J. Wright (2020) Intermediation and steering: Competition in prices and commissions, American Economic Journal: Microeconomics.

Zhang, J., Adomavicius, G., Gupta, A. and Ketter, W. (2020). Consumption and Performance: Understanding Longitudinal Dynamics of Recommender Systems via an Agent-Based Simulation Framework. Information Systems Research, 31(1), pp.76-101.