Improving Recommendation Diversity with Probabilistic Item Selection

ALEX P. MILLER, Wharton School, University of Pennsylvania
KARTIK HOSANAGAR, Wharton School, University of Pennsylvania

The standard k-nearest neighbor collaborative filtering algorithm is well known to exhibit systematic popularity bias, by which products with higher market shares are disproportionately more likely to be recommended. We propose a novel method for improving the diversity of collaborative filtering algorithms and mitigating this bias. Our proposal works by stochastically recommending items with probabilities proportional to each item's sales or views among similar users. One advantage of our method is that it can be combined with other collaborative filtering techniques. In this work, we specifically evaluate how our method interacts with the classical k-nearest neighbor collaborative filter, a popularity discounted collaborative filter, and a recently proposed algorithm based on probabilistic neighborhood selection. After presenting theoretical evidence for the unbiasedness of our approach, we examine the performance of our algorithm using both simulation and empirical analyses. Whereas previous diversity-improving methods come with a commensurate loss in system performance, our analyses provide evidence that probabilistic item selection is an effective method for increasing recommendation diversity, with essentially no degradation in accuracy. Furthermore, we find that—in contrast to the other methods analyzed in this project—probabilistic item selection exhibits complementarities, suggesting that it can be combined with existing methods for improving recommendation diversity with no loss in performance.

1 Introduction

In online retail and digital media, the goal of recommender systems is to make relevant and useful matches between users and products. Anyone can make relevant recommendations by simply selecting the most popular items in a retailer's catalog; by definition, these items will have large appeal to large numbers of consumers. However, these items are also those that most users are already aware of and would have likely purchased without a recommendation. An important part of the utility of recommendations is their role in aiding product discovery. Thus, successful recommendation systems are often thought of as those strike an optimal balance between discovery and relevance. In the literature on recommender systems, these two objectives are operationalized as recommender *diversity* and *accuracy*.

As researchers have become more aware of the importance of recommender diversity, they have uncovered a systematic bias in the collaborative filtering algorithm, which is among the most widely used methods in commercial recommendation systems. Specifically, the classical k-nearest neighbor collaborative filter is disproportionately more likely to recommend popular items than less popular items. This problem is only exacerbated over time because recommended items are themselves more likely to be purchased—making them even more popular. This leads to an adverse "rich-get-richer" feedback loop that concentrates recommendations among a shrinking set of items—ultimately limiting the diversity of product recommendations and reducing their usefulness as engines of discovery. Furthermore, as recommenders skew product purchases towards popular items, the distribution of sales reflects an inaccurate picture of true consumer preferences. As a result, popularity bias in recommender systems can lead to lower recommendation accuracy and, potentially, harm both sales and consumer welfare [Fleder and Hosanagar, 2009].

In this work, we present a novel technique for mitigating popularity bias in collaborative filters. The standard k-nearest neighbor collaborative filter works by finding a neighborhood of consumers similar to the focal recommendation target, and then recommending the most frequently purchased items in this neighborhood. Rather than deterministically selecting the most popular items purchased by similar users, the algorithm we propose in this study stochastically samples items from this list, with probabilities proportional to the observed frequency of each item.

By introducing a stochastic step into the collaborative filtering algorithm, we predictably increase the diversity of recommendations. However, it is important to ensure that our proposed method does not reduce the overall accuracy of the recommender, i.e., how well recommendations match consumer preferences. To evaluate the performance of our algorithm on both accuracy and diversity measures, we conduct two complementary numerical analyses. First, given the inherent dynamic nature of the problem of popularity bias described above, we conduct a simulation analysis that captures the path-dependent evolution of recommendations through time. Since a primary concern with popularity bias is that initial disparities in sales distributions will grow over time, it is important that our evaluation technique captures the dynamic interactions between consumer choices and recommendation algorithms. However, one may reasonably be concerned about the external validity of pure simulation analysis. For this reason, we conduct a separate empirical analysis of our algorithm on a real-world dataset using standard techniques in the literature on recommender systems and information retrieval.

An appealing characteristic of our proposed method is that it can be layered on top of existing approaches to improve recommender diversity. This allows us to identify whether our algorithm performs essentially the same function as other methods, or if it operates through complementary mechanisms. We compare our method of probabilistic item selection to two existing algorithms that are designed to increase recommendation diversity: popularity discounting and probabilistic neighbor selection. Furthermore, we will combine our method with these alternatives to study whether complementarities exist among the various methods and, if so, determine which approach has the largest effect on performance outcomes.

Our main findings are two-fold: first, probabilistic item selection appears to provide meaningful gains in recommendation diversity with practically no loss in accuracy. Furthermore, we show that combining probabilistic item selection with the other approaches results in meaningful complementarities. Whereas combining other methods can result in lower performance on some outcomes, when our method is interacted with other approaches, we observe essentially a Pareto improvement in performance. This indicates that probabilistic item selection operates through unique mechanisms that can significantly enhance the performance of existing approaches for improving recommender diversity.

2 Background on Product Diversity in Recommender Systems

2.1 Related Literature. Early work on recommender systems predominately focused on improving the *accuracy* of recommender systems, i.e., ensuring that the ranking of recommended products matches consumers' underlying preferences [Breese et al., 1998, Desrosiers and Karypis, 2011, Herlocker et al., 1999]. However, recent developments have started to change the focus of some recent work in the field. First, several studies have shown that diversity in recommendations can improve the user-perceived quality of a recommender system [Ekstrand et al., 2014, McNee et al., 2003]. Second, it was demonstrated in [Fleder and Hosanagar, 2009] that the most popular recommendation algorithms—k-nearest neighbor collaborative filters—reduce aggregate sales diversity by disproportionately recommending popular items. This not only limits the diversity of recommendations provided to any individual user, but it also prevents accurate matches between users and items.

The combination of these insights—that users prefer recommendation diversity and that commonly used algorithms exhibit a systematic popularity bias—has motivated several researchers to propose new recommendation methods with a focus on increasing the diversity of recommendations. One proven method for improving recommendation diversity is to complement collaborative filtering techniques with content based methods. Content based recommenders, that take into account the inherent features of items in a product catalog, are known to increase sales diversity [Brynjolfsson et al., 2011]. Indeed, methods for improving the diversity of collaborative filtering-based systems with item attributes dates back many years [Ziegler et al., 2005]. Such techniques are popular among industry practitioners, however it should be noted that gathering data about item attributes may be expensive or simply infeasible in many environments. As such, improving the diversity of purely collaborative filtering-based techniques remains an important topic of research.

Existing work on this particular topic includes Adomavicius and Kwon [2012], who proposed an item re-ranking method in which the algorithm re-orders all entries above a certain accuracy threshold by inverse popularity. The chosen threshold essentially creates a trade-off between diversity and accuracy, with the goal of increasing the former without significantly reducing the latter. The same authors [Adomavicius and Kwon, 2011] have also proposed an algorithm that optimizes recommender diversity by exploiting the implicit network structure between products and users and interpreting diversity within a graph theoretic max-flow framework. Kaminskas and Bridge [2016] conducted a comprehensive analysis of many re-ranking techniques and studied the correlation and dependence between various diversity-related outcome measures (which they refer to as *serendipity*, *novelty*, *diversity*, and *coverage*).

By rethinking the standard collaborative filter paradigm, in which recommendations are made based on the most popular products among similar users, Said et al. [2012] propose recommending the *least* popular products among the most *dissimilar* users. Another creative approach is put forward by Vargas and Castells [2014], who suggest interpreting the recommendation problem as one of recommending customers to products, rather than the common paradigm of recommending products to customers.

We briefly highlight the work of Adamopoulos and Tuzhilin [2014], who propose a recommendation algorithm which is similar in spirit to the method we develop in this project. In their paper, Adamopoulos and Tuzhilin suggest using a *probabilistic nearest neighbor* technique. This is achieved by stochastically sampling a set of neighbors based on each candidate neighbor's similarity to the focal user. Our approach, in contrast, probabilistically samples *items*, after the focal user's neighborhood has been determined. We will revisit the probabilistic neighborhood algorithm in subsequent sections and discuss more about how our algorithm relates to and complements their approach.

2.2 Measuring Diversity. Each of the aforementioned papers that propose remedies for popularity bias in collaborative filters uses a similar methodological framework: the researchers use historical data from product ratings or purchase databases and evaluate their algorithms by measuring the diversity of recommended products using their new algorithms. This is ultimately a *supply-side* analysis. While recommendation diversity is certainly correlated with sales diversity [Brynjolfsson et al., 2011], it is important to distinguish between them. This is because consumers may selectively accept or reject recommendations, leading to a discrepancy between recommendation and consumption. Furthermore, these studies are *offline* analyses, which allow no interaction between recommendation platform and consumers. One of the primary criticisms of recommendation systems is how they can lead to adverse feedback loops, in which popular products get recommended, making them even more popular. While the increase in supply-side diversity documented in previous studies may help mitigate this problem, it is not immediately clear how the proposed solutions would behave in *dynamic* environments. However, it should be noted that such offline analyses do benefit from strong external validity, by virtue of using data taken directly from real-world environments.

Because this work is focused primarily on analyzing sales diversity and because this phenomenon is fundamentally dynamic in nature, we find it necessary to ensure our methodology accounts for this fact. This motivates our use of both a dynamic stylized model and path-dependent simulation analysis. These allow us to endogenize the dynamics of recommendations and consumer choice in our analysis. At the same time, we recognize that these methods may not directly generalize to real-world environments. For this reason, we will also analyze our proposed algorithm using the prevailing paradigm of offline, empirical evaluation on archival data. In this way, we take advantage of each method's complementary strengths: the dynamism of modeling and simulation, along with the external validity of empirical evaluation. This not only allows us to demonstrate the robustness of our algorithm in dynamic environments, but it also allows us to make an apples-to-apples comparison of our algorithm with the performance of existing approaches designed to improve recommendation diversity.

3 Model

3.1 Collaborative Filter Framework. To aid in describing and contextualizing the algorithm we propose in this paper, we will first introduce a 4-step framework that decomposes the fundamental steps in the traditional collaborative filtering process.

Suppose a firm has I potential customers and makes recommendations from a catalog of J products.[1] Let P represent the $I \times J$ matrix of recorded purchases among these consumers. An individual entry in this matrix, P_{ij}, indicates that consumer c_i has purchased product p_j exactly P_{ij} times in the firm's recorded sales history. Let the index of the focal consumer — i.e., the target of a given recommendation — be represented by index i^*. We identify four steps in the standard collaborative filtering paradigm, describing how the purchase matrix is used to select a specific recommendation for the focal consumer:

1. *Neighbor weighting*: Assign weights $w_i^{i^*} = F(P, i^*, i)$ (for all $i \neq i^*$) according to some function of the purchase history matrix to quantify how "similar" each user c_i is to the focal consumer c_{i^*}.
2. *Neighbor selection*: Use these weights to select a focal neighborhood, i.e., a set of k users $\mathcal{N}^*$ who are similar to c_{i^*}.
3. *Item weighting*: Assign recommendation weights, $r_j^* = G(P, \mathcal{N}^*, j)$, to all $j = 1 \ldots J$ products using some function of the purchase frequencies among users in the focal neighborhood.
4. *Item selection*: Use these recommendation weights to select a product, p^*, to be recommended.

As a concrete case, we describe the classical k-NN collaborative filter in this framework. In the neighborhood weighting step, k-NN applies a similarity function on the purchase vectors to compute a vector of weights between customers. Commonly used similarity functions at this step include cosine similarity and Pearson's correlation coefficient. Neighborhood selection is then

[1] To fix a specific context, we consider a firm that makes individual product recommendations (one at at time), which consumers can either accept or ignore. Note, however, that the following framework could just as well be adapted for "views" rather than purchases (say in the context of a video recommendation service) or situations in which many recommendations are provided simultaneously.

performed by choosing the consumers with the largest k similarity values relative to the focal consumer. Then items are assigned weights by simply counting the frequency with which they have been purchased by consumers in the selected neighborhood. Finally, the item with the largest frequency is selected for recommendation.

This general framework allows us to clearly identify the manipulations of various extensions of the classical collaborative filter. As examples, item re-ranking techniques [Adomavicius and Kwon, 2011] modify the assignment of weights in step 3; the inverse user frequency similarity metric (based on the well-known TF-IDF algorithm) [Breese et al., 1998] would be an alternative neighbor weighting function in step 1.

3.2 Probabilistic Item Selection. In this work, we propose a novel extension of the classical collaborative filter. Specifically, our algorithm introduces a method of *probabilistic item selection* (PI for reference) which has several attractive properties. As described above, in the classical collaborative filter, once the the focal user's k nearest neighbors are identified, recommendations are determined by taking the best-selling item among these neighbors. In contrast, our algorithm will recommend an item with probabilities proportional to the sales of each product in the focal neighborhood. As a result, while the best-selling item in the focal user's neighborhood is most likely to be recommended, other products also have a chance of being recommended. The more popular an item is among the k nearest neighbors, greater the probability of the item being recommended.[2]

Using the notation from the previous section, our probabilistic item selection algorithm randomly chooses an item p^* by sampling items with weights proportional to their sales among users in the focal neighborhood $\mathcal{N}^*$:

$$\Pr(p^* = p_j) \propto \sum_{i \in \mathcal{N}^*} P_{ij}$$

3.3 Theory. *Diversity and Popularity Bias.* By introducing a stochastic item selection step in place of a deterministic one, we will invariably increase recommender diversity. This is because in the classical collaborative filtering method, a focal neighborhood fully determines which item is to be recommended, whereas our method randomly selects a recommended item among many possibilities. However, rather than simply increasing the stochasticity of the recommendation process and *mitigate* popularity bias, our method of probabilistic item selection is in some sense optimally designed to *eliminate* this bias.

In Appendix A, we outline an analytical model of a dynamic recommendation process. We prove that, with our probabilistic item selection algorithm, the limiting behavior of the item market shares converges to the market shares that would be observed in absence of a recommendation algorithm. In other words, our algorithm enhances discovery but does not distort the sales distribution towards more popular items; over time, our algorithm learns the underlying distribution and recommends items with frequencies proportional to the true preference distribution among a focal user's nearest neighbors. This is in contrast to earlier work on pure k-NN collaborative filtering methods that have been proven to exhibit systematic popularity bias [Fleder and Hosanagar, 2009]. Thus, while the classical collaborative filter is known to have distorting effects on aggregate sales diversity, we have theoretical evidence that probabilistic item selection will not share these biases.

Accuracy. While our proposed algorithm will increase recommendation diversity, it is not clear that it can do so without harming recommendation accuracy. Indeed, in some of the literature on recommender diversity, accuracy and diversity are treated as mutually exclusive objectives, in which we must trade off one for the other [Adomavicius and Kwon, 2012]. However, we have some

[2] Note that our method can easily accommodate contexts in which a *list* of recommendations is required. One would simply construct an ordered list by iteratively taking samples *without replacement* until the desired list length is achieved.

reason to suspect that our method may be able to increase diversity without a drop in accuracy. First, consider that the biases caused in dynamic recommendation systems will result in biased input data for any recommendation algorithms. That is, the consumption patterns observed in the presence of recommender bias do not reflect the true, underlying preferences of consumers. By mitigating these biases, our algorithm allows the system to (over time) observe a higher-fidelity pattern of consumption.

Further, we note that our algorithm is similar to the Thompson sampling algorithm from the literature on reinforcement learning and multi-armed bandits [Agrawal and Goyal, 2012]. In particular, both our algorithm and Thompson sampling are designed to stochastically select an action with probabilities proportional to that action's likelihood of being optimal. In that context, stochastic action selection provides a near-optimal balance between exploring new actions and exploiting information about already-observed actions. To be sure, there are important differences between typical bandit settings and the context of recommender systems that should give us caution in generalizing too much between our algorithms: recommender systems often have to choose from action spaces that are orders of magnitude larger than most bandit settings, the data observed by the system is heavily influenced by the choices of users themselves (rather than being fully determined by an algorithm, as is the case in bandit settings), and the existing paradigms for evaluating recommender performance is different than the paradigm of regret in the bandit literature. Nonetheless, we emphasize that there is an existing precedent in related literature for how stochastic action selection—by finding the trade-off between exploration and exploitation—can improve overall system performance.

4 Evaluation

Having provided mathematical and theoretical arguments for why probabilistic item selection can be expected to perform well on both recommendation diversity and accuracy, we now turn to the empirical evidence for this hypothesis.

4.1 Comparison Algorithms. Rather than evaluating the performance of our algorithm in isolation, we will be comparing it to the performance of existing algorithms with similar goals. As a baseline, we will use the the classical k-NN collaborative filter. To motivate the other comparison algorithms in this analysis, we return to the 4-step framework outlined in §3.1. Note that our method operates by increasing diversity at the last step in the collaborative filtering process (this is captured in the name: probabilistic *item selection*). A natural question that arises from our framework is how our technique might compare to diversity-enhancing approaches that operate at other steps in the recommendation process. We outline two such methods that are found in the existing literature on recommendation diversity below.

Probabilistic Neighbor Selection. For our first comparison algorithm, we return to the work of Adamopoulos and Tuzhilin [2014]. In their paper, the authors suggest using a *probabilistic nearest neighbor* algorithm when selecting a focal user's neighborhood in a collaborative filter (k-PN for reference). The comparison of our method with k-PN is useful for several reasons. First, given the ostensible similarity of our approaches, it is not obvious that our method contributes anything novel beyond the k-PN method. Further, the two methods operate by performing similar functions at separate phases in the recommendation process: probabilistic sampling at either the *neighbor* selection phase or the *item* selection phase. By directly comparing each technique separately, we will able to provide insight into which step in the collaborative filtering process has the most influence on recommender performance.

We can operationalize k-PN for our context using the same notation introduced in §3. We first assign weights using the cosine similarity metric between the focal user and all other users:

$$w_i^{i^*} = \text{cosine}(P_{i\cdot} + 1, i^*)$$

Table 1. Existing Algorithms in Collaborative Filter Framework

Algorithm	k-NN	k-PN	P^{-1}	PI (this paper)
Neighbor Weighting	cosine($P_{i\cdot}$, i^*)	cosine($P_{i\cdot} + 1$, i^*)	cosine($P_{i\cdot}$, i^*)	cosine($P_{i\cdot}$, i^*)
Neighbor Selection	Top k	Weighted sample	Top k	Top k
Item Weighting	Frequency in $\mathcal{N}^*$	Frequency in $\mathcal{N}^*$	Frequency in $\mathcal{N}^*$ times P^{-1}	Frequency in $\mathcal{N}^*$
Item Selection	Top 1	Top 1	Top 1	Weighted sample

Note: Gray cells represent each algorithm's point(s) of divergence from classical k-NN

where $P_{i\cdot}$ represents the vector of purchases across the J products by user i. [3] Lastly, we perform an iterative sample of users without replacement to construct a neighborhood of size k. This is done by iteratively sampling k neighbors from the set of all users $\{c_i \mid i = 1 \dots I\}$ into the focal neighborhood $\mathcal{N}^*$. The probability of each user c_i being selected at each step is defined by the "empirical distribution" of similarity weights, which is just each user's share of similarity divided by the aggregate of all user similarities:

$$p(c_i) \propto \begin{cases} 0 & \text{if } i \in \mathcal{N}^* \\ w_i^{i^*} \Big/ \sum_{u=1}^{I} w_u^{i^*} & \text{otherwise} \end{cases}$$

Popularity Discounting. While k-PN represents an algorithm that affects neighbor selection, other algorithms operate at different steps in the collaborative filtering process. Again, to offer some insight into which step has the most influence on recommendation outcomes, we wanted to identify an algorithm that affects the item weighting process. One such algorithm is popularity discounting (P^{-1} for reference). This method is closely related to the inverse user frequency weighting scheme of Breese et al. [1998], who recognized that universally liked items provide less meaningful signals of similarity between users than less popular items. Popularity discounting is motivated by the same concern, but is slightly different in that it operates at the item weighting step rather than the neighbor weighting step in the collaborative filtering process. We choose to study popularity discounting over inverse user frequency in this analysis because it has been shown in prior research to be more effective at mitigating popularity bias [Fleder and Hosanagar, 2009].

The popularity discounting algorithm works as follows: it first assigns item weights by calculating the frequency of items in a focal neighborhood, but it then divides this value by the frequency of items in the total population. The motivation behind this method is to ensure that extremely popular items that are too obvious to be useful as recommendations are down-weighted in proportion to their overall popularity. This algorithm recommends items that are popular among a focal user's nearest neighbors but are not overly popular in the total population. In our model notation, popularity discounting calculates item weights, r_j^*, in the following manner:

$$r_j^* = \left(\sum_{i=1}^{I} P_{ij} \right)^{-1} \sum_{i \in \mathcal{N}^*} P_{ij}$$

Each of the algorithms described so far is succinctly summarized using the 4-step collaborative filtering framework in Table 1. The step at which a given algorithm differs from the classical k-NN collaborative filter is colored in gray, highlighting how each algorithm affects a different phase in the recommendation process.

[3] The addition of 1 ensures that all users have a non-zero probability of being selected into the focal neighborhood.

Table 2. Hybrid Algorithms in Collaborative Filter Framework

Algorithm	$[P^{-1}]$ + [k-PN]	$[P^{-1}]$+ [PI]	[k-PN] + [PI]	$[P^{-1}]$ + [k-PN] + [PI]
Neighbor Weighting	$\text{cosine}(P_{i\cdot} + 1, i^*)$	$\text{cosine}(P_{i\cdot}, i^*)$	$\text{cosine}(P_{i\cdot} + 1, i^*)$	$\text{cosine}(P_{i\cdot} + 1, i^*)$
Neighbor Selection	Weighted sample	Top k	Weighted sample	Weighted sample
Item Weighting	Frequency in $\mathcal{N}^*$ times P^{-1}	Frequency in $\mathcal{N}^*$ times P^{-1}	Frequency in $\mathcal{N}^*$	Frequency in $\mathcal{N}^*$ times P^{-1}
Item Selection	Top 1	Weighted sample	Weighted sample	Weighted sample

Note: Gray cells represent each algorithm's point(s) of divergence from classical k-NN

4.2 Hybrid Algorithms. In the previous sections, we described three extensions of the classical k-NN collaborative filter: probabilistic item selection, probabilistic neighbor selection, and popularity discounting. A natural question is to ask how the various approaches would interact if they were combined. We can view each of the three methods as an independent binary manipulation of the classical k-NN algorithm, resulting in a total of $2^3 = 8$ possible ways to combine the three methods. The independent versions of the methods are already described in Table 1 and we have organized the four interacted "hybrid" algorithms in Table 2. Again, we have highlighted the cells in gray that represent each algorithm's points of divergence from the classical k-NN collaborative filter.

We can reasonably hypothesize that the interacted algorithms will increase recommendation diversity over the classical k-NN algorithm. However, note that there are several factors that are not easy to predict *ex ante*. First, it's not obvious that the mechanisms have truly independent effects on the ultimate diversity of recommendations. Perhaps layering k-PN on probabilistic item selection will result in recommendations that are no more diverse than the independent algorithms. This would indicate that our technique essentially mimics the effects of the k-PN algorithm through alternative mechanisms. However, if we find that the hybrid algorithms outperform the independent algorithms, this indicates the presence of complementarities between the various approaches. Furthermore, by evaluating the full factorial interaction between the methods, we will be able to quantify which method has the most significant impact on the outcomes of interest. Finally, it is critical that in our attempt to increase recommendation diversity we do not compromise on the original goal of recommender systems, which is to help users find products that they actually like. Even if the independent algorithms increase recommendation accuracy, there is no clear way to predict how the interacted methods will perform on accuracy measures.

We now turn to a simulation analysis to study the performance of the various algorithms we have identified in this section. This methodology will allow us to study each of the algorithms on a host of outcomes related to both diversity and accuracy.

5 Simulation Analysis

As mentioned earlier, popularity biases in recommendation systems are fundamentally dynamic in nature and depend on the path-dependent actions of both recommendation platform and consumers. In this section, we set out to evaluate our proposed probabilistic items algorithm using a simulation framework that explicitly allows for dynamic interactions between consumers and the recommendation system. Furthermore, because our simulation includes a model of consumer demand, we will be able to measure several outcome variables that are of interest to both firms and researchers, but very difficult to study effectively using archival data. These include both aggregate and individual-level measures of recommendation diversity and sales diversity; consumer welfare and recommendation accuracy; and overall volume of retailer sales.

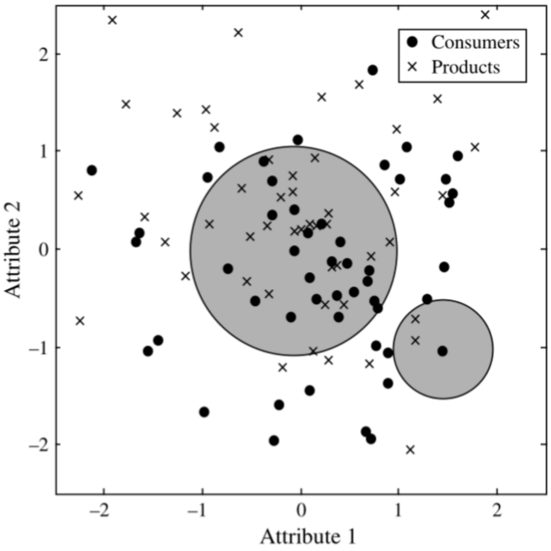


Fig. 1. Sample of Consumers and Products in Attribute Space

While our framework cannot capture all essential features of recommendation environments, it does allow for the most important dynamics of how recommendation algorithms and consumers interact through time. We discuss the most important elements of the analysis below.[4]

5.1 Simulation Design. (i) *Ideal Point Model of Consumers and Products.* The simulation is built around a two-dimensional ideal point model. The dimensions represent two abstract product attributes in this market. The preference of an individual consumer is represented by their position in this space (their "ideal point"); the products available for purchase in this market are also characterized by their coordinates in attribute space. As will be described in more detail below, consumers get higher utility from purchasing products that are close to them, as measured by some metric over this space. This model is based on classical economic models of heterogeneous consumer preference [Hotelling, 1929, Tirole, 1988] and is commonly used in marketing [Elrod and Keane, 1995]. At the beginning of each simulation, we initialize the model by drawing both consumer and product points from i.i.d. standard bivariate normals. In this study we have $I = 50$ consumers and $J = 50$ products. For illustrative purposes, a sample draw of this space for one of our simulations is shown in Figure 1.

(ii) *Awareness.* One of the primary motivations for the use of recommendation systems is that it is not possible for consumers to perform an exhaustive search of the product space. This means that—at least initially—consumers should only be aware of a subset of products in the market. We capture this by only allowing consumers to purchase products in their "awareness set." Initially, this set includes only products that are either close to the consumer in attribute space or close to the origin. Because the generating distribution of the ideal points are zero-centered, the origin represents the area with popular mass appeal. Thus, we assume that consumers are primarily aware of mass appeal products and products close to their preferences. As we will describe below, when a consumer is recommended a product, it is added to their awareness set. At time $t = 0$, awareness sets are initialized by sampling every consumer-product pair according to the following

[4] This model is inspired the work of [Fleder and Hosanagar, 2009]. Readers may refer to that paper for a complementary description of the simulation process.

probability:

$$P(c_i \text{ is aware of } p_j) = \lambda\sigma(0,j) + (1-\lambda)\sigma(i,j)^{\kappa} \tag{1}$$

where $\sigma(i,j)$ is a similarity metric between consumer i and product j in the attribute space; $i = 0$ refers to the origin. λ varies the extent to which consumers are more likely to be aware of central, mainstream products (λ) versus products in their own neighborhood of the attribute space $(1-\lambda)$. κ reduces the radius of the local neighborhood. By limiting this radius, we ensure that customers are not fully aware of all the products that are close by, which allows recommendations to serve a useful role by adding products consumers like (but are not already aware of) into their consideration set.

The awareness region corresponding to items that have a greater than 10% chance of entering into the consideration set of a sample consumer is illustrated by the gray regions in Figure 1. Based on the formula in equation (1), κ controls the relative size of the radii between central and local neighborhoods and λ controls the relative weighting of awareness within each neighborhood.

(iii) *Choice Model.* We model consumer purchase decisions at each time period using the classical discrete choice (or multinomial logit) model [McFadden, 1973]. We define consumer c_i's utility from product p_j at time t as the sum of a deterministic component and a stochastic component: $u_{ijt} = v_{ij} + \varepsilon_{ijt}$. We assume the deterministic component is fully captured by the *similarity* between consumer i and product j, which we define as follows:

$$v_{ij} := similarity_{ij} = -k \log distance_{ij}$$

where $distance_{ij}$ is the Euclidean distance in product space between c_i and p_j. As is consistent with the discrete choice model, we assume the random component of utility is independently and identically distributed as an extreme value random variable. Define consumer i's consideration set at time t as $A_{it} = \{p_j \mid c_i \text{ aware of } p_j \text{ at time } t, \text{for } j = 1 \ldots J\}$. Each consumer's purchase probability distribution, conditional on product awareness, is then given by:

$$P(c_i \text{ buys } p_j \text{ at } t \mid c_i \text{ aware of } p_j \text{ at } t) = \frac{e^{u_{ijt}}}{\sum\limits_{j \in A_{it}} e^{u_{ijt}}}$$

In addition to the products in a consumer's consideration set, an "outside good" (indexed by $j = 0$) is also always included in the choice model. This good is placed at the same distance ($distance_{i0} = 0.75$) for each individual consumer. This value was chosen because it ensures the outside option is a relatively attractive choice for most consumers (on average, the outside good is in the top the 87th percentile of all values in a consumer's choice set).

If consumer c_i selects good p_j on a given choice occasion, this is recorded by incrementing the (i,j) entry in the purchase history matrix, P_{ij}, by one unit; if the consumer chooses the outside good, no changes are made to the matrix.

(iv) *Recommender System.* At the beginning of every time period, a recommendation is determined for the focal consumer based on the purchase history matrix P up to that point in time. When consumer c_i is recommended product p_j, this product is permanently added to their consideration set A_{it} (if it is not already there). We will vary the precise mechanism by which recommendations are made based on the algorithms described in §4. Each algorithm operates on the purchase history matrix P in a given simulation, according to the methods outlined in Tables 1 and 2.

(v) *Recommender Salience.* We assume that if product j has been recommended to consumer i at time t, then the deterministic component of that consumer's utility for that product temporarily

Table 3. Simulation Notation and Configuration

Design Choice	Notation	Value used in simulation
Attribute space dimensions	—	2
Generating distribution	—	Φ
Number of consumers	I	50
Number of products	J	50
Measure of similarity	$\sigma(i, j)$	$e^{(c_i - p_j)^2/\kappa}$
Central vs. local awareness	λ	0.75
Local awareness scaling	κ	3
Effect of distance on utility	k	10
Distance of outside good	$distance_{i0}$	0.75
Salience boost	δ	5
Nearest neighbor similarity	—	cosine
Number of nearest neighbors	—	10
Purchases before recommendations	—	500
Purchases with recommendations	—	500
Total simulations per algorithm	—	1000

increases by an additive factor δ for that time period only: $u_{ijt} = v_{ij} + \delta + \varepsilon_{ijt}$. Thus recommendations have dual effects: making consumers aware of a product and temporarily increasing the salience of that product.

5.2 Simulation Procedure. Many other alternative specifications relating to the number and distribution of consumers and products, the dimensionality of the product space, the various functions used to calculate similarity, the choice model, and the effect of the recommender are considered in [Fleder and Hosanagar, 2009]. Our simulations use the base-case definitions for all the parameter values from that paper, which are concisely summarized in Table 3. Note that, as in that work, alternative specifications for these parameters have been experimented with—e.g., 3, 4, or 5-dimensional attribute space; uniform distribution of ideal points; Pearson correlation similarity metric. These changes affect the precise quantities of the reported results, but the qualitative findings remain unchanged.

A single simulation proceeds as follows: The positions of consumers and products in attribute space are randomly drawn. For each of 500 iterations, every consumer will purchase one product according to the choice model with no recommender system involved. This is a "burn-in" period that provides the recommendation algorithm with a baseline level of sales history to base its initial recommendations on. Then, recommendations are turned on and consumers will undergo 500 more iterations of product purchases in which consumers are given a recommendation in every period. A focal consumer's nearest neighbors are calculated just-in-time and corresponding product recommendations are always based on the most recently available data.

The entire simulation procedure was repeated 1000 times for each of the eight recommendation algorithms, with new consumer-product locations drawn for each simulation. At the end of each simulation, we have an $I \times J$ purchase history matrix corresponding to the actions of consumers through the recommendation phase of the simulation. Using this matrix, we report the mean for each of the outcome measures described below, averaged across all 1000 simulations.

5.3 Outcome Measures. We evaluate the performance of each of the algorithms along three high-level dimensions, each with a separate metric that corresponds to the demand-side and the supply-side. Demand-side refers to what consumers actually purchased and supply-side refers to the products consumers were recommended.

Aggregate Diversity. For measures of aggregate diversity we report on two standard metrics used in the recommender systems literature: First, we report the **Gini coefficient** of the distribution of product sales. Further, we also report a measure of **coverage**, which is the percentage of the all products that are in at least one user's consideration set.[5] Note these metrics provide a measure of demand-side and supply-side aggregate diversity, respectively.

Individual Diversity. We use measures of average unique items bought per person (**AUIBP**) and average unique items aware of per person (**AUIAP**) to capture diversity at the individual level. Again, these correspond to the demand-side (products users bought) and supply-side (products that were recommended) aspects of our model.

Accuracy and Welfare. By the design of the consumer utility function, purchases of products that result in higher utility are closer to the consumer in attribute space. In our context, this proximity in attribute space is precisely what researchers mean when they refer to recommender "accuracy", i.e., how well do recommended products line up with consumers' exogenous preferences. Thus we use the total **average utility** across consumers summed over all purchase occasions as a measure of both accuracy and consumer welfare.

Because our model includes an outside good option in the choice model, a sale does not occur in every simulated choice. Also note that the probability of the outside good being chosen diminishes as the accuracy of a recommender increases: more consumers being aware of more goods that are close to them in attribute space reduces the relative likelihood of the outside good choice. Thus, recommenders with high accuracy will also have a positive effect on sales, which is presumably the primary outcome of interest for retailers. We use the proportion of consumer choices that resulted in the consumption of an "inside" good (i.e, not the outside good p_0) as our measure of **retailer sales**.

5.4 Results.

The full cross tabulation of results is reported in Table 4. This includes the mean of the 1000 simulations performed for each of the eight algorithms across all six outcome variables.[6]

Let us first review the relative outcomes of the four independent algorithms, when they are not combined with any other method. For this section, we limit our analysis to comparing the first four rows in Table 4.

Begin by considering the performance of k-PN (Row 2 in results table) relative to k-NN. Probabilistic neighbor selection appears to improve performance on all outcomes, with the largest gains (compared to other changes in the table) observed on the accuracy outcomes (utility and sales). Interestingly, while coverage appears to increase dramatically from 0.127 to 0.282, the Gini coefficient is only marginally improved from 0.823 to 0.819. While k-PN does unilaterally increase diversity, focusing solely on the supply-side diversity (coverage) appears to overstate its effect on demand-side diversity (the Gini coefficient of the sales distribution).

The outcomes of the popularity discounting algorithm P^{-1} (Row 3) are essentially essentially inverse to k-PN. Specifically, while k-PN only marginally reduced the Gini compared to k-NN, it had moderate to significant improvements on the other outcomes. However, P^{-1} has a significant impact on the Gini coefficient (a decrease from 0.823 to 0.720), and only marginal effects on the

[5] Note that because consideration sets are initialized with a non-empty set of products, this definition of coverage is not the percentage of the catalog that is recommended at least once. However, since all simulations are initialized according to the same process, the relative differences between algorithms on this metric corresponds to those that would be observed for coverage as used elsewhere in the literature.

[6] We omit standard errors to allow for easy comparison across algorithms in our table. However, with 1000 simulations, the standard errors on each variable are very small; for example, standard errors around the Gini coefficient are between 0.0010 and 0.0016. All contrasts described between algorithms discussed in the text are significant at the 5% level.

Table 4. Simulation Results

Row	Algorithm	Gini	Coverage	AUIBP	AUIAP	Utility	Sales
1.	[k-NN]	0.823	0.127	1.70	6.36	8.73	0.701
2.	[k-PN]	0.819	0.282	2.98	14.11	10.20	0.843
3.	[P^{-1}]	0.720	0.145	2.32	7.23	9.23	0.707
4.	[PI]	0.731	0.295	4.48	14.76	10.49	0.800
5.	[P^{-1}] + [k-PN]	0.768	0.137	1.97	6.84	9.14	0.653
6.	[P^{-1}] + [PI]	0.670	0.337	5.08	16.84	10.88	0.811
7.	[k-PN] + [PI]	0.555	0.881	5.85	44.03	13.56	0.904
8.	[P^{-1}] + [k-PN] + [PI]	0.491	0.975	5.84	48.77	14.09	0.924

other outcomes relative to k-NN. This indicates that while popularity discounting does accomplish the goal of mitigating popularity bias, it does not offer much in additional gains to recommendation accuracy, individual recommendation diversity, or overall welfare in our model.

Our proposed algorithm of probabilistic item selection (Row 4) appears to offer significant gains for all outcomes. Probabilistic item selection has very comparable performance to popularity discounting for eliminating popularity bias (0.720 compared to 0.731), and similar performance all other outcomes to probabilistic neighbor selection. Even the most significant differences between probabilistic neighbor selection and probabilistic item selection (apart from the Gini coefficient) are rather modest: k-PN has slightly higher sales values (0.843 compared to 0.800), while PI has slightly higher AUIBP (2.98 compared to 4.48). Thus, the proposed method of probabilistic item selection results in significant gains in aggregate diversity, while maintaining comparable (or better) performance on all other metrics.

We now expand our analysis to consider all eight algorithms, with a particular focus on the interactions between the different methods. Recall from the previous section we observed that probabilistic neighbor selection and popularity discounting have, what would appear to be, complementary strengths: popularity discounting reduced concentration bias, whereas probabilistic neighbor selection had increased performance on the other outcomes. Interestingly, when we combine the two methods (Row 5), we do not get good results. While the combined algorithm has a lower Gini coefficient (0.768) than k-NN (0.823), it does not have a lower coefficient than the independent popularity discounting algorithm (0.720). Furthermore, whereas the independent k-PN algorithm exhibited significant gains among the other metrics compared to k-NN, the combined [P^{-1}] + [k-PN] algorithm is essentially comparable to k-NN, with the combined algorithm's performance on sales actually resulting in a *worse* outcome than k-NN (0.701 compared to 0.653).

Turning to rows 6 and 7 in Table 4, observe that interacting our probabilistic item selection method with either of the other approaches appears to almost unilaterally increase performance. This is not only true in comparison to the baseline k-NN algorithm, but it is also true when compared with any of the three independent approaches. Note that complementarities relative to k-NN can either be super-additive or sub-additive. Specifically, note how k-PN only reduced the Gini by 0.04 compared to k-NN (0.823 − 0.819) and probabilistic item selection reduced the Gini by 0.092 (0.823 − 0.731). However, the change in Gini observed when these two methods are interacted (Row 7, [k-PN] + [PI]) is much larger than the sum of these two independent effects: 0.268 (0.823 − 0.555). This is what we refer to as super-additivity. Alternatively, we find that the interaction of the improvement in Gini between probabilistic item selection and popularity discounting (Row 6, [P^{-1}] + [PI]) is sub-additive: $0.092 + 0.103 = 0.195$ in combined independent differences, compared to $0.823 - 0.670 = 0.153$ in interacted difference.

Lastly, observe that for each of the outcomes measured, the largest gains appear to be driven specifically by our proposed method of probabilistic item selection. One way to demonstrate this is by comparing the algorithm with the three-way interaction (Row 8) to the separate two-way interactions, each of which are missing one of the components of the fully-interacted algorithm (Rows 5, 6, and 7). We focus on consumer utility as a specific example, but the same qualitative findings hold for any of the outcomes. Note that adding popularity discounting ($P_{\cdot j}^{-1}$) to the two-way interaction of the other methods (comparing rows 7 and 8) results in a moderate increase in utility from 13.56 to 14.09 (+0.53). Adding probabilistic neighbor selection (k-PN) to the two-way interaction of the other methods increases utility from 10.88 to 14.09 (+3.21). However, adding our method of probabilistic item selection to the two-way interaction without it, results in the largest increase in utility from 9.14 to 14.09 (+4.95). Another way of framing this result is that taking away probabilistic item selection from any algorithm that already includes it results in larger drops in performance when compared to the drops observed by taking away the other methods.

We emphasize the remarkability of the magnitude of gains observed by the best performing algorithms in *both diversity and accuracy* measures. Particularly when we compare rows 7 and 8 with other rows in the table, we see combining probabilistic neighbor selection and probabilistic item selection dramatically improves performance across the board. These combined algorithms increase both aggregate and individual levels of diversity while simultaneously improving recommendation accuracy.

6 Empirical Evaluation

While we have argued for the value of the simulation analysis for studying the problem of recommendation diversity, simulated models have notable limitations. Though our main results are robust to several alternative assumptions in the simulation model, the previous analysis does not constitute empirical evidence for the utility of our approach. To ameliorate concerns about external validity, we set out to complement our simulation analysis with an empirical evaluation of our probabilistic item selection algorithm.

To do this, we make use of the publicly available LastFM dataset [Cantador et al., 2011]. This dataset contains a record of the number of times a sample of users have listened to each musical artist in their LastFM listening history. (For continuity with the terminology used in the rest of the paper, we'll refer to artists as "items".) We use standard preprocessing techniques to restrict our analysis to those users and items with a significant number of non-zero entries in the dataset [Adomavicius and Kwon, 2009, Manimekalai et al., 2013].[7] Applying this selection procedure to the LastFM dataset results in a history matrix containing data on the number of times 1,830 individual users have listened to 522 musical artists. The density (i.e., the percentage of non-zero user-item pairs) of this dataset is 4.8%.

6.1 Evaluation Methodology. In line with other offline evaluation studies, we employ N-fold cross validation to determine the accuracy of the algorithms in this analysis [Adamopoulos and Tuzhilin, 2014, Cremonesi et al., 2008]. In this context, cross validation takes place at the user level. For each user, we take the set of items with explicit feedback (i.e., the set of items the user has consumed in the past) and partition it into N folds. We then iterate over the partitions by using $N-1$ folds as training data and treating the last fold as test data. In this analysis, we fix N=3.

Accuracy Measure. We will say that a given item in the recommendation list is "relevant" (in the information retrieval sense) if it is in the top 75th percentile of the focal user's observed item frequencies [Basu et al., 1998]. That is, among the set of items that a user has consumed at least

[7] In this project, we chose 30 as the minimum for both of these values, though we have experimented with other values that indicate our results are not affected by this restriction.

once, we assume the top 25% most frequently consumed items would be considered as relevant by that user. Using this approach ensures that relevance is a user-specific construct and not based on any inherent differences between users (i.e., overall number of purchases, number of distinct items consumed, etc.).

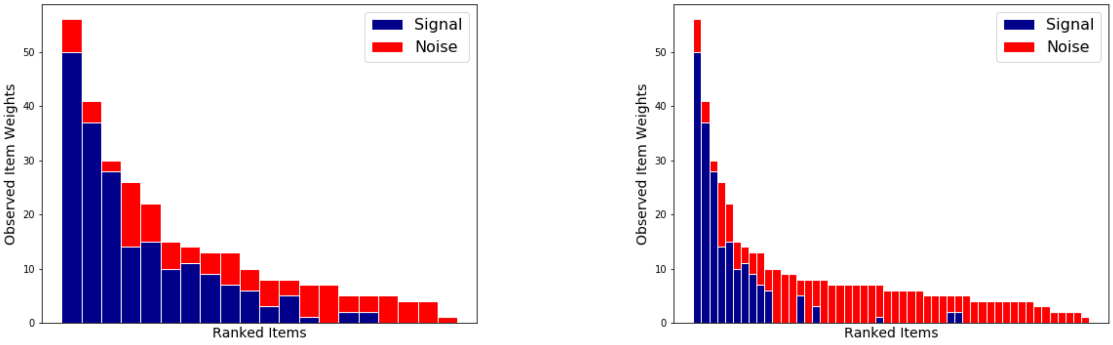
The primary metric we use for accuracy is the normalized discounted cumulative gain (NDCG), which is a prominent measure of relevance in the information retrieval literature [Järvelin and Kekäläinen, 2002, Shani and Gunawardana, 2011]. This measure not only rewards algorithms that recommend relevant items, but it also takes into account the order in which the items are recommended. This is motivated by the notion that a relevant recommendation in the last position of a list is less valuable than a relevant recommendation at the beginning of the list. Using the normalized measure also allows us to average the accuracies across users and make direct comparisons on this measure between algorithms. The values we report in the results below are average NDCG measures across all users and folds.

Diversity Measures. We will again use both the Gini index and coverage to capture measures of aggregate diversity. As discussed earlier, because we are not assuming a model of individual consumption, these are measures of supply-side diversity, i.e., the diversity of items that are recommended. However, this does come with the benefit of not requiring us to impose any assumptions about consumer demand as was necessary in the simulation analysis.

6.2 Practical Considerations. A well-known challenge in working with empirical datasets is dealing with the effects of both sparsity and noise [Herlocker et al., 1999]. Even after purging the data of users or items with a small number of observations, there still remains a significant skew in the distribution of data available for individual users and the practical challenge of handling imperfectly measured preferences. The process of evaluating recommendation algorithms on archival datasets is known to be susceptible to the effects of noise generally, but we also emphasize that our proposed algorithm of probabilistic item selection is particularly sensitive to noise in a predictable way [Amatriain and Pujol, 2015]. In most real-world datasets, the observed item frequencies in a given user's neighborhood have a very heavy tail, in the sense that there is a subset of well-ordered popular items at the head of the distribution, but a very large number of items with low frequency. For example, a large number of items might be consumed just once by one user in the focal user's neighborhood. Each of these items may individually have a low probability of being recommended but may collectively represent a reasonably high probability of being recommended.

Because our algorithm makes recommendations based on frequency shares in the entire distribution of items, even a small amount of noise in the measurement of frequencies can drastically reduce the hit rate of our item sampling algorithm. This is because the noise in the tail of the distribution has an *additive* effect. This relationship between the probability of recommending a top item and the number of items is demonstrated in Figure 2. In this figure, we simulated a scenario in which we imagine item weights for a sample user can be decomposed into a "true" component (or "signal") and a "noise" component. In particular, we have taken observed weights for 20 items (colored in blue) and added i.i.d. Poisson noise (colored in red) and sorted the items by the sum of these weights to simulate what we would see as researchers if the true item weights are observed imperfectly.

In Figure 2(a), we simply plot the observed item weights (sum of true and noise components) for the 20 sampled items. Recommending an item with our stochastic algorithm is akin to throwing a (uniformly random) dart at the bars of the graphs above. The item on which the dart lands is the item that will be recommended. In this scenario, 74% of the time, the dart will land in the signal portion of the graph. In other words, an item will be recommended due to its true item weight in 74% of item selections. In Figure 2(b), we used the same true weights for the 20 items



(a) Signal Ratio with 20 Items: 74%

(b) Signal Ratio with 50 Items: 29%

Fig. 2. Effect of Noise Begins to Dominate as Number of Items Increases

in 2(a); the only difference is that we suppose there 30 more items in the database for which the true weights are zero, but whose weights are still observed with Poisson noise. This is to simulate the scenario in which the universe of items is much larger than those for which a user has a meaningful preference for (as is often the case in digital media and e-commerce recommendation systems). In this simulated world, a dart thrown at the graph will land on a signal component in only 29% of cases. Even though the same absolute amount of signal is present in both graphs, if many items in the tail of the database are observed with a small amount of noise, this noise will accumulate. This additive noise accumulation will systematically bias our stochastic selection algorithm toward items in the tail. In particular, the probability of recommending an item among the user's most preferred items (e.g., in the top decile) decreases monotonically as the number of items grows. This effect is *not* present for sorting-based algorithms such as classical k-NN, which deterministically recommends the items at the top of the ranked item list and are unaffected by noise effects in the tail of the item weight distribution.

Given the particular susceptibility of our probabilistic items algorithm to the presence of noise in large, real-world datasets, we introduce a small modification that keeps the core idea of recommending items stochastically, but with some measures to counteract the effects of noise. The main idea is to limit the number of items from which we sample and then to bias this sample towards those items with higher frequencies. In practice, for a given list length of size l, we probabilistically sample $2l$ items from the most popular $6l$ items in a given user's neighborhood. We then sort this sample of $2l$ items by frequency and recommend the top l items in this list. Note that this modification does introduce extra parameters that must be selected beforehand (specifically, the multipliers of 6 and 2 described here). We do not claim that these values will be universally optimal for any application, but much in the same way that the optimal value of k in k-NN algorithms must be adapted to each dataset, these parameters will also need to be adjusted in other applications. We emphasize, however, that we seek to demonstrate that *some* version of our proposed algorithm can outperform existing methods on diversity measures while maintaining comparable performance on accuracy, not whether the particular parameter values chosen here generalize to all datasets.

6.3 Results. As in Section 5, we report the performance of our algorithm (with the aforementioned modifications) relative to existing methods designed to increase diversity. There are two arbitrary parameters that must be selected in the evaluation process: the number of neighbors to use in the collaborative filtering algorithm (k) and the recommendation list size (l). To establish a base case, we will first consider k=50 and l=25, which are similar to other values for these parameters in the literature [Adamopoulos and Tuzhilin, 2014, Hurley and Zhang, 2011]. After reviewing this case, we will discuss the effect of varying k and l. The results for this base case across the eight algorithms and three outcome metrics are displayed in Table 5.

Table 5. Empirical Results

		NDCG	Gini	Coverage
1.	[k-NN]	0.622	0.613	0.989
2.	[k-PN]	0.614	0.612	0.990
3.	[P^{-1}]	0.428	0.343	1.000
4.	[PI]	0.621	0.566	0.998
5.	[P^{-1}] + [k-PN]	0.397	0.341	1.000
6.	[P^{-1}] + [PI]	0.425	0.289	1.000
7.	[k-PN] + [PI]	0.613	0.567	0.998
8.	[P^{-1}] + [k-PN] + [PI]	0.395	0.284	1.000

As before, let us compare the performance of the independent algorithms first (rows 1-4). The classical k-NN algorithm does have the highest accuracy (0.622) out of any method, but it is worth noting that both the *PI* (0.621) and k-PN (0.614) methods are close behind. The inverse popularity weighting method, P^{-1}, stands out from the rest with a significantly worse performance on accuracy (0.428).

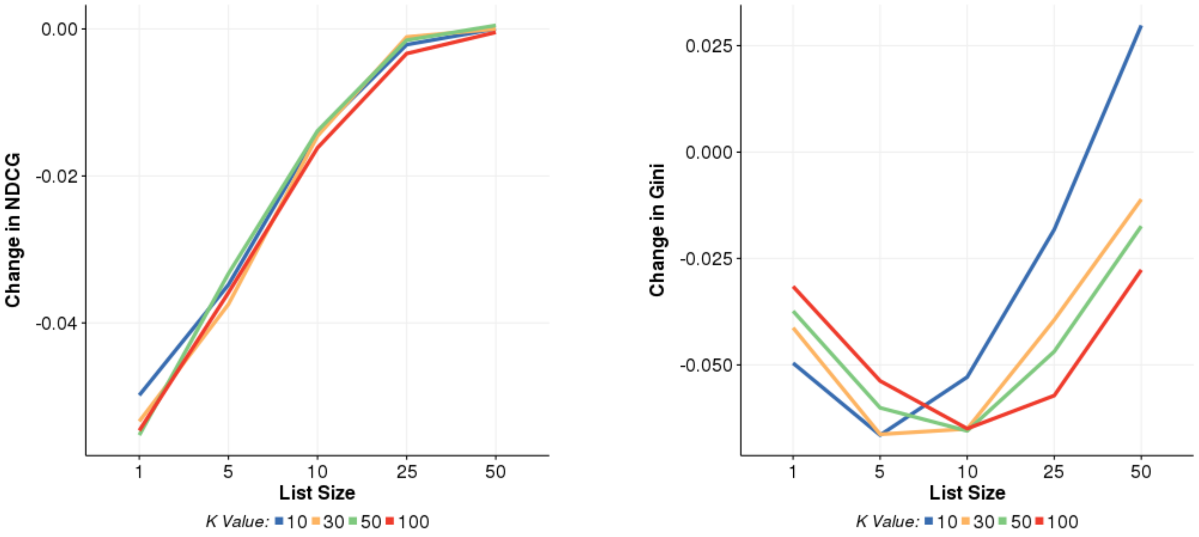
That none of the algorithms are unable to beat k-NN should not be particularly surprising, as it is quite rare for an algorithm in published literature to empirically outperform k-NN on accuracy. Furthermore, it is important to emphasize that the goal of diversity-improving algorithms is to recommend items that user's would not have otherwise encountered in absence of the recommender. Because our evaluation methodology is only able to judge an item as relevant if the user has already found it, these algorithms will be penalized even though they may have recommended a relevant item. This limitation can be overcome by having a model of user preferences, that would allow us to impute utility for items yet-to-be seen by users in our dataset. This is, of course, what we accomplished in our simulation analysis, in which we saw that diversity-improving algorithms resulted *increased* rather than diminished utility. Even with this bias in our evaluation methodology, the accuracy between k-NN and our proposed method of probabilistic items can reasonably be considered comparable for practical purposes.

Turning to the diversity measures, we see relatively little differentiation in terms of coverage, but significant variation in Gini coefficients. We see that, as expected, k-PN is able to increase diversity to a small degree compared to k-NN (slightly lower Gini, slightly higher coverage). Though inverse popularity weighting, P^{-1}, did have the lowest measured accuracy, it also exhibits the most diversity with a relatively small Gini coefficient of 0.343. Lastly, we note that the gains in diversity exhibited by our method of probabilistic items—while not near that of P^{-1}—are quite substantial (0.566) relative to k-NN (0.613) for an algorithm that maintains such high accuracy. Our algorithm seems to be best able to balance accuracy and diversity among all the independent approaches. Turning to the rest of the rows in the table, we see that any algorithm with inverse popularity weighting (rows 3, 5, 6, 8) has a significantly lower accuracy than those without. The probabilistic neighbors and probabilistic items algorithms seem to interact relatively well (row 7), though the performance of this hybrid algorithm is not strictly better than either of the independent approaches.

While we do not observe the near uniform benefits of complementarity that we saw in our simulation analysis, there is still an interesting pattern to find in these empirical results. In particular, we highlight the fact that when probabilistic items is combined with any other approach, the diversity of that approach goes up by a significant amount while maintaining a comparable level of accuracy. This can be seen by comparing rows that only differ by the addition of the probabilistic item selection step in the results table (row pairs (1,4), (3,6), (2,7), and (5,8)). In each of these cases, the change in accuracy is near 0.001 and the drop in Gini is close to 0.05. This same pattern does not hold for the other approaches. Adding inverse popularity weighting to other algorithms results in dramatic decreases in both accuracy and Gini values (on the order of 0.2 and 0.25, respectively). The losses in accuracy observed between rows with and without probabilistic neighbor selection, while small in absolute terms, is always larger than that observed for probabilistic items (typically near 0.01); similarly, while probabilistic neighbors does consistently increase diversity, the change in Gini coefficient is typically quite small (on the order of 0.001). This pattern indicates that our method of probabilistic item selection exhibits the strongest complementarities with other methods for increasing recommendation diversity. For a recommendation method with a given level of accuracy, adding probabilistic item selection to that method can be expected to provide meaningful gains in diversity with negligible losses in accuracy.

6.4 Robustness. We will now examine how the performance of our algorithm varies as we change the values of the input parameters. While the general patterns described above hold for most other portions of the parameter space explored in our research, there are some important exceptions that merit further discussion.

We are primarily interested in understanding how the performance of our probabilistic items selection algorithm changes as we vary the number of neighbors (k) and the recommendation list size (l) in the empirical analysis. In particular, we are concerned most with the *relative* performance of our algorithm, rather than its absolute performance as these parameters change. To visualize how the performance of the probabilistic items algorithm changes, we will plot the *difference* in performance between probabilistic items and the baseline k-NN algorithm, for both accuracy (measured



(a) Change in accuracy, relative to k-NN

(b) Change in diversity, relative to k-NN

Fig. 3. Performance of Probabilistic Items as Neighborhood and List Size Vary

in level changes to NDCG values) and diversity (measured in level changes between Gini coefficients). These graphs are plotted in Figure 3, in which each line represents the change in outcome value between the PI and k-NN algorithms across varying list sizes for a fixed neighborhood size.

Focusing first on the accuracy graph, Figure 3(a), we see a relatively steep drop-off in performance as the list size decreases. This is not particularly surprising for diversity-increasing algorithms, since the margin for error goes down as the list size gets smaller (it is easier to recommend a long-tail item as 1 out of 100 than it is 1 out of 5 without harming accuracy). Interestingly, as the list size approaches 50, probabilistic item selection appears to perform equally well as (if not better than) k-NN.

Turning to the diversity graph, Figure 3(b), we see a non-monotonic relationship between list length and relative performance for all neighborhood sizes. An important feature of this graph is how nearly every change in the Gini observed is smaller than -0.025, indicating that non-trivial improvements in diversity can be found by using probabilistic items for most values of the parameter space. (Recall how the changes in Gini for k-PN were on the order of 0.001.) There is one case in which the Gini is *larger* for probabilistic items relative to k-NN, which is for the edge case of the smallest neighborhood size of $k = 10$ and the largest list size of $l = 50$. Note that this is an extreme situation, in which we are trying to recommend many items from just information provided by a small set of neighbors and, as such, is not a likely set of parameter values to be used in practice. Overall, we can see that probabilistic items provides meaningful gains in diversity with little to no loss in accuracy.

7 Conclusion

In this study, we presented evidence that probabilistic item selection can significantly improve the diversity of collaborative filtering-based recommendation systems. After providing theoretical motivation for why we expected our algorithm to perform well, we then supported this hypothesis with evidence from both simulation and empirical analyses. In our simulation, we attempted to capture the most important characteristics of *dynamic* recommendation systems, in which users and algorithms interact in a path-dependent manner. We then conducted a more traditional, archival analysis of the performance of our algorithm on a real-world dataset that provided empirical evidence for the utility of the probabilistic item selection approach.

Our analysis suggests that probabilistic item selection can be an effective technique for improving recommendation diversity, while maintaining system accuracy. We also compared our approach with other diversity-improving techniques that operate at different steps in the collaborative filtering process. This allowed us to find that our method not only seems to outperform the independent alternatives considered in this paper, but we also found that probabilistic item selection exhibits the most fruitful complementarities with the other approaches. Because our method acts at the very last stage in the recommendation process, our evidence suggests that probabilistic item selection can be applied to other item ranking-based recommendation methods.

References

Panagiotis Adamopoulos and Alexander Tuzhilin. 2014. On Over-specialization and Concentration Bias of Recommendations: Probabilistic Neighborhood Selection in Collaborative Filtering Systems. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, New York, NY, USA, 153–160. DOI:http://dx.doi.org/10.1145/2645710.2645752

Gediminas Adomavicius and YoungOk Kwon. 2009. Toward more diverse recommendations: Item re-ranking methods for recommender systems. In *Workshop on Information Technologies and Systems*. Citeseer.

Gediminas Adomavicius and YoungOk Kwon. 2011. Maximizing aggregate recommendation diversity: A graph-theoretic approach. In *Proc. of the 1st International Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011)*. 3–10.

Gediminas Adomavicius and YoungOk Kwon. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2012), 896–911.

Shipra Agrawal and Navin Goyal. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*. 39–1.

Xavier Amatriain and Josep M Pujol. 2015. Data mining methods for recommender systems. In *Recommender systems handbook*. Springer, 227–262.

Chumki Basu, Haym Hirsh, and William Cohen. 1998. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the The Fifteenth National Conference on Artificial Intelligence (AAAI-98)*. AAAI, 714–720.

John S Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 43–52.

Erik Brynjolfsson, Yu Hu, and Duncan Simester. 2011. Goodbye pareto principle, hello long tail: The effect of search costs on the concentration of product sales. *Management Science* 57, 8 (2011), 1373–1386.

Ivan Cantador, Peter L Brusilovsky, and Tsvi Kuflik. 2011. *Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011)*. ACM.

Paolo Cremonesi, Roberto Turrin, Eugenio Lentini, and Matteo Matteucci. 2008. An evaluation methodology for collaborative recommender systems. In *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on*. IEEE, 224–231.

Christian Desrosiers and George Karypis. 2011. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*. Springer, 107–144.

Michael D Ekstrand, F Maxwell Harper, Martijn C Willemsen, and Joseph A Konstan. 2014. User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 161–168.

Terry Elrod and Michael P Keane. 1995. A factor-analytic probit model for representing the market structure in panel data. *Journal of Marketing Research* (1995), 1–16.

Daniel Fleder and Kartik Hosanagar. 2009. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management Science* 55, 5 (2009), 697–712.

Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*. ACM, New York, NY, USA, 230–237. DOI:http://dx.doi.org/10.1145/312624.312682

Harold Hotelling. 1929. Stability in Competition. *The Economic Journal* 39, 153 (1929), 41–57.

Neil Hurley and Mi Zhang. 2011. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)* 10, 4 (2011), 14.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

Norman Lloyd Johnson and Samuel Kotz. 1977. *Urn Models and Their Application: An Approach to Modern Discrete Probability Theory*. Wiley, New York, NY.

Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* 7, 1, Article 2 (Dec. 2016), 42 pages. DOI:http://dx.doi.org/10.1145/2926720

S Manimekalai, V Kathiresan, and P Sumathi. 2013. Recommendation Diversity using Optimization Techniques and Ranking Method. *International Journal Of Engineering And Computer Science* 2, 09 (2013).

Daniel McFadden. 1973. Conditional logit analysis of qualitative choice behavior. In *Frontiers in Econometrics*, P. Zarembka (Ed.). Wiley, New York, 105–142.

Sean M McNee, Shyong K Lam, Catherine Guetzlaff, Joseph A Konstan, and John Riedl. 2003. Confidence displays and training in recommender systems. In *Proc. INTERACT*, Vol. 3. 176–183.

Alan Said, Benjamin Kille, Brijnesh J Jain, and Sahin Albayrak. 2012. Increasing diversity through furthest neighbor-based recommendation. *Proceedings of the WSDM* 12 (2012).

Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender systems handbook*. Springer, 257–297.

Jean Tirole. 1988. The Theory of Industrial Organization. *MIT Press Books* 1 (1988).

Saúl Vargas and Pablo Castells. 2014. Improving sales diversity by recommending users to items. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 145–152.

Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 22–32.

Online Appendix

Improving Recommendation Diversity with Probabilistic Item Selection

A Appendix

In this section, we present formal theoretical arguments to demonstrate that, in contrast with classical k-nearest neighbor collaborative filtering, our proposed probabilistic item selection algorithm exhibits no systematic popularity bias.

A.1 General Analytical Framework. We consider a set of customers making purchases sequentially. Because our interest is in studying sales concentration (i.e., the relative distribution of market shares between products), our focus is less on how recommenders affect sales volume. Thus, our choice model is conditional on purchase, and only allows agents to choose *which* product to buy. This allows us to isolate choice effects from incidence.[1]

Consider a discrete process in which a set of I consumers, $C = \{c_1, \ldots, c_I\}$, choose among a set of J products, $P = \{p_1, \ldots, p_J\}$ over time. At each time period t, one consumer, $c^t \in C$, purchases one product, $S^t \in P$. This purchase decision is influenced by two factors: a consumer's latent type and the recommendation provided by the platform. Formally, the latent characteristics for consumer $c_i \in C$ are defined by a type vector $\theta_i = (\theta_{i1}, \ldots, \theta_{iJ})$ that represents their purchase probabilities over the J items; the type vector θ_i for the focal consumer c^t at time t is denoted θ^t. The product that is recommended at time t, denoted R^t, is a function of the focal consumer, c^t, and the history of product purchases before time t. The history vector can be defined as $H^t = \{(c^1, S^1), \ldots, (c^{t-1}, S^{t-1})\}$, which simply collects the user-item purchase pairs over time. Concretely, we represent the recommendation function as r and the consumer choice model as f:

$$R^t := r(H^t, c^t) \tag{1}$$

$$S^t := f(R^t, \theta^t) \tag{2}$$

[1] Note that there are real contexts in which incidence is much less of a concern than conditional choice, such as a subscription media service where marginal costs of consumption are zero.

Both r and f can be thought of as either deterministic or stochastic functions. Note r maps the aggregate history vector H^t and consumer index c^t into the space of products (without no dependence on latent consumer preferences θ^t); f maps the recommended product R^t and the consumer type vector θ^t into the product purchased at time t, S^t. The history vector H^{t+1} is then updated to include the pair (S^t, c^t).

Our primary outcome of interest is how recommender systems affect the relative distribution of sales among products. In the model outlined above, market shares at time t can be determined from the record of past sales, $S^1, \ldots, S^{t-1}$. We define the market share vector X^t over the set of J products as follows:

$$\begin{aligned} X^t &= (x_1^t, \ldots, x_J^t) \\ x_j^t &:= \frac{1}{t-1} \sum_{t'=1}^{t-1} \mathbf{I}(S^{t'} = p_j) \end{aligned} \tag{3}$$

where $\mathbf{I}(S^{t'} = p_j)$ is the indicator function for whether product p_j was purchased at time t'. Since market shares at any finite time t will be affected by random fluctuations, the primary analytical quantity we are interested in considering is the limit of market shares as t goes to infinity:

$$X^\infty := \lim_{t \to \infty} X^t \tag{4}$$

A.2 Analytical Assumptions. Much of the setup described in the next section replicates the model from Fleder and Hosanagar (2009), in which the authors proved under similar assumptions that traditional collaborative filters exhibit popularity bias (readers are referred to that paper for an alternative description of this model). Before proceeding, we outline the main assumptions made in our model.

(i) We assume there are only two products w and b (white and black).

(ii) We pre-select a segment of consumers who have been identified to be similar based on their past purchases (possibly from products in other categories). This assumption fixes the set of similar users rather than letting it evolve over time. This model will have no consumer-specific indices, since preferences are considered homogeneous in this segment.

(*iii*) With only two products and one segment of consumers, we define the base-rate consumer preference variables in terms of a single parameter p: $(\overline{\theta}_w, \overline{\theta}_b) = (p, 1-p)$. This vector represents the segment's purchase probabilities for w and b in the absence of recommendations.

(*iv*) We make r a function solely of market shares at time t, $r(X^t)$. This is a common characteristic of many recommender systems, which ignore the chronology of purchases.

The assumption of homogeneity is necessary because nearest neighbor algorithms are intractable to iteratively compute in an analytical model. Even in industrial recommendation systems, these computations are often too intensive for firms to calculate after each purchase. It is not an uncommon practice to only update consumer segments periodically, indicating that our analytical model does have parallel with business practice. Even so, assumptions (*i*), (*ii*), and (*iii*) are eliminated in the simulation analysis described in Section 5 of the main text. That analysis analyzes more realistic market conditions with a large number of products, heterogeneous consumers, and dynamically updated consumer segments.

A.3 Base Case: Deterministic Collaborative Filter. We now specify a functional form for the recommender r, inspired by the way collaborative filters are used in practice. The idea behind collaborative filters, i.e., "people like you bought product Y", is typically operationalized in the following manner: firms find customers similar to the focal customer for which they are providing a recommendation, sort the list of available products by the number of purchases among this segment, and recommend products at the top of this sorted list.

As mentioned, our model considers a pre-selected segment of similar customers that does not evolve with t. In this setting, the market share X^t represents the distribution of product purchases among this focal segment. Thus, recommending the product with highest market share in X^t is precisely the classical collaborative filtering algorithm.

Because our model has only two products, the market share vector can be parameterized by one number; for notational convenience, we use the market share of white as a stand in for X^t and drop the w subscript throughout this example: $X^t = (x^t_w, x^t_b) = (x^t, 1 - x^t)$. Note that $x^t \in [0, {}^1\!/\!_2)$ indicates that b has the highest market share at time t; $x^t \in ({}^1\!/\!_2, 1]$ indicates w has the highest market share. This leads us to the following formal definition of the recommender function r; because we are moving from a generic recommendation algorithm to a specific one, we denote this function with a subscript, r_d, and refer to it as the *deterministic collaborative filter*:

$$\begin{aligned} r_d(x^t) &:= \text{product recommended on occasion } t \mid x^t \\ &= \begin{cases} b, & x^t \in [0, {}^1\!/\!_2) \\ b(1-Z) + wZ, & x^t = {}^1\!/\!_2, \quad Z \sim \text{Bernoulli}({}^1\!/\!_2) \\ w, & x^t \in ({}^1\!/\!_2, 1] \end{cases} \end{aligned} \tag{5}$$

Conditional on x^t, this function is indeed deterministic, except in the special case that product shares are equal. In this edge case, the recommendation is determined by a Bernoulli random variable Z, in which both products have an equal chance of being recommended. We now define a closely related function ρ which represents the probability of w being recommended at time t:

$$\begin{aligned} \rho(x^t) &:= P(r_d(x^t) = w \mid x^t) \\ &= \begin{cases} 0, & x^t \in [0, {}^1\!/\!_2) \\ {}^1\!/\!_2, & x^t = {}^1\!/\!_2 \\ 1, & x^t \in ({}^1\!/\!_2, 1] \end{cases} \end{aligned} \tag{6}$$

Recall from Section A.1, the *influence function* f, which maps product recommendation and underlying consumer preferences into a purchase outcome. To fully describe this function, and to facilitate the intuition behind our model, we now express our model in a Pólya urn framework. Urn models are useful for analyzing stochastic processes and have a long history in a wide range of applications (**?**).

Consider the two urn system of Figure 1. Urn 1 represents the consumers' underlying, time-invariant preferences between the two products; a fraction p of the balls in urn 1 are white, with the remaining fraction $1-p$ of balls being black. Urn 2 represents the dynamic market shares of product purchases over time. At each step t in the process, a ball is drawn randomly from urn 1 (with replacement). This represents the consumer's preference-weighted product choice in the absence of recommendations. Next, a ball is drawn from urn 2 (with replacement) according to the recommendation rule r. This indicates the recommended product. We now introduce an additional parameter π, which is the "strength" of the recommender. The stochastic choice model f is defined by the colors of the balls drawn urns 1 and 2 and this parameter π in the following manner: with probability π, the consumer accepts the recommendation from urn 2 given by $r(x^t)$, and with probability $1-\pi$ the consumer ignores the recommendation and purchases the product drawn from urn 1. An extra copy of the purchased product is then added to urn 2 (effectively updating the recommender's sales history). The process then repeats with new draws from the urns. To initialize this system, 1 ball of each color is placed in urn 2, which ensures that the recommender is not biased toward either product at $t = 1$.

To complete the analytical description of our model, we calculate the probability of each product being purchased at time t as a function of the market shares x^t. Using the choice process described above, we calculate this quantity as the probability that the product

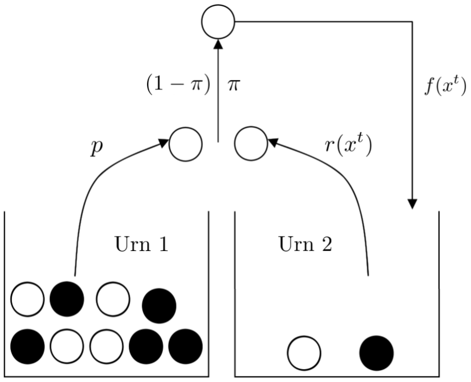


Figure 1: A Two-Urn Model for Recommender Systems

purchased at time t, $f(x^t)$, is white and denote this function as φ:

$$\begin{aligned}\varphi(x^t) &:= P(f(x^t) = w \mid x^t) \\ &= P(w \text{ purchased without recommender influence}) \\ &\qquad + \ P(w \text{ purchased due to recommendation} \mid x^t) \\ &= p(1-\pi) + \rho(x^t)\pi \qquad (7) \\ &= \begin{cases} p(1-\pi), & x^t \in [0, 1/2) \\ \dfrac{p(1-\pi) + [p(1-\pi)+\pi]}{2}, & x^t = 1/2 \\ p(1-\pi)+\pi, & x^t \in (1/2, 1] \end{cases}\end{aligned}$$

A.4 Elimination of Popularity Bias with Stochastic Collaborative Filter. In attempt to mitigate the popularity bias found in classical collaborative filters, we propose a modification of the recommendation function r_d described in (5). We denote this new recommender function by r_s and call it the *stochastic collaborative filter*:

$$r_s(x^t) = b(1-Z) + wZ, \ \text{ where } Z \sim \text{Bernoulli}(x^t) \qquad (8)$$

This recommender has the simple property that the probability of a product being recommended is equal to its market share among the focal segment of consumers. For the white product in our example, we represent this quantity by ρ, which reduces to:

$$\rho(x^t) = P(r_s(x^t) = w) = x^t \qquad (9)$$

Note that since our model starts out with one of each type of product in urn 2, x^t (and therefore $\rho(x^t)$) never takes on the edge values of 0 and 1. This means at every finite step t, the recommendation r_s is determined by a stochastic outcome Z. Relative to the deterministic recommender, this stochastic recommender appears to increase the diversity of recommendations at any finite t, since each product has positive probability of being recommended. However, our primary outcome of interest, the limiting behavior of x^t as $t \to \infty$ remains to be determined. This brings us to the main analytical finding from our model, which is summarized in the following proposition:

Proposition 1. $\lim\limits_{t\to\infty} E[x^t] = p$. *In other words, the limiting market shares of a system with probabilistic item selection equal the market shares that would be observed in the absence of a recommendation algorithm.*

To prove this proposition, we will make use of the following lemmas:

Lemma 1. *Consider a recursive series of the form $v_{t+1} = \alpha_t v_t + \beta_t$, with arbitrary constants α_t, β_t and initial given value v_1. The formula for an arbitrary v_{t+1} can then be written as:*

$$v_{t+1} = \left(\prod_{i=1}^{t} \alpha_i\right) v_1 + \left(\prod_{i=1}^{t} \alpha_i\right) \sum_{j=1}^{t} \frac{\beta_j}{\left(\prod_{i=1}^{j} \alpha_i\right)} \tag{10}$$

Proof. The proof proceeds by induction. We start with the base case $t = 1$. Note in this case, equation (10) reduces to the original series definition:

$$v_2 = \alpha_1 v_1 + \alpha_1 \frac{\beta_1}{\alpha_1} = \alpha_1 v_1 + \beta_1$$

Now assume equation (10) holds for an arbitrary $t = T$. That is, we take it as given that

$$v_{T+1} = \left(\prod_{i=1}^{T} \alpha_i\right) v_1 + \left(\prod_{i=1}^{T} \alpha_i\right) \sum_{j=1}^{T} \frac{\beta_j}{\left(\prod_{i=1}^{j} \alpha_i\right)}$$

By definition, we have

$$v_{T+2} = \alpha_{T+1} v_{T+1} + \beta_{T+1}$$

Substituting in the assumed form for v_{T+1}, results in

$$\begin{aligned}
v_{T+2} &= \alpha_{T+1}\left[\left(\prod_{i=1}^{T}\alpha_i\right)v_1 + \left(\prod_{i=1}^{T}\alpha_i\right)\sum_{j=1}^{T}\frac{\beta_j}{\left(\prod_{i=1}^{j}\alpha_i\right)}\right] + \beta_{T+1}\\
&= \left(\prod_{i=1}^{T+1}\alpha_i\right)v_1 + \left(\prod_{i=1}^{T+1}\alpha_i\right)\sum_{j=1}^{T}\frac{\beta_j}{\left(\prod_{i=1}^{j}\alpha_i\right)} + \left(\prod_{i=1}^{T+1}\alpha_i\right)\frac{\beta_{T+1}}{\left(\prod_{i=1}^{T+1}\alpha_i\right)}\\
&= \left(\prod_{i=1}^{T+1}\alpha_i\right)v_1 + \left(\prod_{i=1}^{T+1}\alpha_i\right)\left[\sum_{j=1}^{T}\frac{\beta_j}{\left(\prod_{i=1}^{j}\alpha_i\right)} + \frac{\beta_{T+1}}{\left(\prod_{i=1}^{T+1}\alpha_i\right)}\right]\\
&= \left(\prod_{i=1}^{T+1}\alpha_i\right)v_1 + \left(\prod_{i=1}^{T+1}\alpha_i\right)\sum_{j=1}^{T+1}\frac{\beta_j}{\left(\prod_{i=1}^{j}\alpha_i\right)}
\end{aligned}$$

This is precisely the equation given in (10). Since the formula holds for an arbitrary $t = T$ and is true for $t = 1$, it follows by induction that the equation is true for all $t \in \mathbb{N}$.

■

Lemma 2. *Consider the same series from Lemma 1 and suppose the following equations hold for some arbitrary constant c and $b, p > 0$:*

$$\prod_{i=1}^{t}\alpha_i \approx ct^{-b}$$

$$\beta_t = \frac{pb}{t}$$

where $\approx$ indicates equality with the addition of a $\mathcal{O}(\cdot)$ term of higher order. Then $\lim_{t\to\infty} v_{t+1} = p$.

Proof. Using equation (10) from Lemma 1 and substituting in the expressions above, we write v_{t+1} as

$$\begin{aligned}
v_{t+1} &= \left(\prod_{i=1}^{t}\alpha_i\right)v_1 + \left(\prod_{i=1}^{t}\alpha_i\right)\sum_{k=1}^{t}\frac{\beta_k}{\left(\prod_{i=1}^{k}\alpha_i\right)}\\
&= v_1ct^{-b} + ct^{-b}\sum_{k=1}^{t}\frac{pb/k}{ck^{-b}} = v_1ct^{-b} + pt^{-b}b\sum_{k=1}^{t}k^{b-1}\\
&= v_1ct^{-b} + pt^{-b}b\left(\frac{t^b}{b} + \mathcal{O}(t^{b-1})\right) = v_1ct^{-b} + p + \mathcal{O}(t^{-1})
\end{aligned}$$

where the summation has been rewritten using the Euler-Maclaurin summation formula.

Because $b > 0$, we see that the first term goes to zero as $t \to \infty$ leaving only p in the limit. ■

With these results, we can now prove the main proposition.

Proof of Proposition 1. As in equation (7), we define $\varphi(x^t)$ as the probability that a consumer purchases w in time period t. Recall that, under the stochastic recommender r_s, the probability of w being recommended at time t was given by $\rho(x^t) = x^t$. This allows us to compute the explicit form for φ:

$$\begin{aligned} \varphi(x^t) &= P(f(x^t) = w \mid x^t) \\ &= p(1-\pi) + \rho(x^t)\pi \\ &= p(1-\pi) + x^t\pi \end{aligned} \tag{11}$$

Since the model is initialized with one of each product included in urn 2 (at time $t = 1$) and a consumer always purchases one product in each time period, the total number of balls in the urn at an arbitrary t is always $t + 1$. Thus the absolute number of w balls in urn 2 at time t is simply the total number times the market share: $(t + 1)x^t$. Conditioning on whether a w or b ball is purchased at time t, this allows us to calculate the market shares of w at $t + 1$:

$$x^{t+1} = \begin{cases} \dfrac{(t+1)x^t + 1}{(t+1)+1} & w \text{ purchased with probability } \varphi(x^t) \\ \\ \dfrac{(t+1)x^t}{(t+1)+1} & b \text{ purchased with probability } 1 - \varphi(x^t) \end{cases} \tag{12}$$

Taking the conditional expectation of x_{t+1} over these probabilities results in:

$$\begin{aligned} E[x^{t+1} \mid x^t] &= \varphi(x^t)\left[\frac{(t+1)x^t + 1}{t+2}\right] + (1 - \varphi(x^t))\left[\frac{(t+1)x^t}{t+2}\right] \\ &= \frac{\varphi(x^t) + (t+1)x^t}{t+2} \end{aligned}$$

After substituting the value of $\varphi(x^t) = p(1-\pi) + x^t\pi$ from equation (11), this becomes:

$$\begin{aligned} E[x^{t+1} \mid x^t] &= \frac{p(1-\pi) + x^t\pi + (t+1)x^t}{t+2} \\ &= \left(\frac{\pi + t + 1}{t+2}\right) x^t + \frac{p(1-\pi)}{t+2} \\ &= \left(1 - \frac{1-\pi}{t+2}\right) x^t + \frac{p(1-\pi)}{t+2} \end{aligned} \tag{13}$$

Comparing this formula with the series described in Lemma 1 and observing that $\prod_{k=1}^{t}\left(1 - \frac{1-\pi}{k+2}\right) \approx ct^{1-\pi}$ applying the result from Lemma 2, we conclude that:

$$\lim_{t\to\infty} E[x^{t+1}] = p$$

■