Performative Recommendation: Diversifying Content via Strategic Incentives

Itay Eilat [1]   **Nir Rosenfeld** [1]

Abstract

The primary goal in recommendation is to suggest relevant content to users, but optimizing for accuracy often results in recommendations that lack diversity. To remedy this, conventional approaches such as re-ranking improve diversity by *presenting* more diverse items. Here we argue that to promote inherent and prolonged diversity, the system must encourage its *creation*. Towards this, we harness the performative nature of recommendation, and show how learning can incentivize strategic content creators to create diverse content. Our approach relies on a novel form of regularization that anticipates strategic changes to content, and penalizes for content homogeneity. We provide analytic and empirical results that demonstrate when and how diversity can be incentivized, and experimentally demonstrate the utility of our approach on synthetic and semi-synthetic data.

1. Introduction

Recommendation has become a key driving force in determining what content we are exposed to, and ultimately, which we consume (MacKenzie et al., 2013; Ursu, 2018). But despite the commercial success of modern recommendation systems, a known shortcoming is that recommendations tend to be insufficiently diverse, with content homogeneity becoming more pronounced over time; this has been a long-standing issue in the field for over two decades (Carbonell & Goldstein, 1998; Bradley & Smyth, 2001). Diversity is important in recommendation not only for improving recommendation quality (Vargas & Castells, 2014; Kaminskas & Bridge, 2016) and user satisfaction (Herlocker et al., 2004; Ziegler et al., 2005; McNee et al., 2006; Hu & Pu, 2011; Wu et al., 2018; Dean et al., 2020), but also because a lack of diversity can lead to inequity across content creators, which often hurts the 'long-tail' of non-mainstream suppliers (Yin

et al., 2012; Burke, 2017; Singh & Joachims, 2018; Abdollahpouri et al., 2019b; Mladenov et al., 2020; Wang & Joachims, 2021). From the perspective of the recommendation platform, an inability to diversify content translates into an inability to utilize the full potential that lies in the natural variation of user preferences for promoting system goals (Anderson, 2006; Yin et al., 2012). This has lead to widespread interest in developing methods for making recommendations more diverse (Kunaver & Požrl, 2017).

The common approach for diversifying recommendations is to apply some post-processing procedure that re-ranks the output of a conventionally-trained ranking model—which is optimized for predicting user-item relevance—to be more diverse, for example by traversing the ranked list and removing items which are similar to higher-ranked items (Carbonell & Goldstein, 1998; Ziegler et al., 2005; Sha et al., 2016). This simple heuristic approach has been shown to be quite effective—at least when considering a *given* ranked list, and at one point in time. But recommendation is inherently a dynamic process: here we argue that post-hoc methods may not suffice for promoting diversity in the long run.

To see why, consider that re-ranking (and similar approaches) are designed to diversify the *presentation* of content—not content itself. Presenting diverse content may help in the specific instance it targets, but does not change the pool of available items, nor does it account for any downstream affects of recommendation. Recent work has shown that one drawback of using prediction as the basis for recommendation is that it causes homogenization (Chaney et al., 2018); here we argue that rearranging predicted items to form the appearance of diversity does not remedy this.

As an alternative, we propose to encourage the *creation* of diverse content, so that the set of available items becomes inherently more diverse. In this way, we aim to target the cause—rather than the symptom. Our main observation is that content is shaped by *content creators*, who seek to maximize exposure to their items (Ben-Porat et al., 2020; Hron et al., 2022), and hence likely to modify content in ways which promote their item's predicted relevance (or 'score'). Since the goal of learning is to infer such scores, this gives the learning system leverage in shaping the incentives of content creators. Here we propose to utilize this power to incentivize for the creation of more diverse content.

---

[1]Faculty of Computer Science, Technion – Israel Institute of Technology, Haifa, Israel. Correspondence to: Nir Rosenfeld <nirr@cs.technion.ac.il>.

Towards this, we draw connections to the related literature on strategic learning (Brückner et al., 2012; Hardt et al., 2016), and model content creators as gaining utility from the score given to their items by the learned predictive model. Content creators can then improve their utility by strategically modifying their items to obtain a higher score. Thus, a learned predictive model determines not only what items are recommended to which users—but also creates incentives for content creators, which can promote change (Ben-Porat & Tennenholtz, 2018; Jagadeesan et al., 2022). This provides the system with the potential power to steer the collection of renewing items—over time, and with proper incentivization—towards diversity (Hardt et al., 2022).

To study when and how the system can effectively exercise its power, we cast recommendation as an instance of *performative prediction* (Perdomo et al., 2020), which subsumes and extends strategic learning to a temporal setting where repeated learning causes the underlying data to shift over time. Focusing on retraining dynamics, we study when and how learning can be used to incentivize the creation and preservation of diversity. In retraining, our only means for driving incentivizes derives from *how* we retrain, i.e., from our criterion for choosing the predictive model at each round. Since retraining aims for models that are predictively accurate, our goal will be to provide recommendations that are accurate *and* diverse. But diversity and accuracy can be at odds; hence, we seek to understand how they relate, and to propose ways in which their tradeoff can be optimally exploited.

We begin with a basic analysis demonstrating the mechanisms through which incentives and diversity relate within our setup. We then propose a learning objective that allows to balance ranking accuracy (and in particular NDCG) with diversity through a novel form of regularization, which we use to maximize diversity under accuracy constraints. Our proposed diversity regularizer has two main benefits. First, it is differentiable, and hence can be optimized using gradient methods. Second, it can be applied to *strategically-modified inputs*; this equips our objective with the ability to anticipate the strategic responses of content creators, and hence, to encourage predictive rules that incentivize diversity. Our proposed strategic response operator is also differentiable; thus, and using recent advances in differentiable learning-to-rank, our entire strategic learning objective becomes differentiable, and can be efficiently optimized end-to-end.

Using our proposed learning framework, we empirically demonstrate how properly accounting for strategic incentives can improve diversity—and how neglecting to do so can lead to homogenization. We begin with a series of synthetic experiments, each designed to study a different aspect of our setup, such as the role of time, the natural variation in user preferences, and the cost of applying strategic updates. We then evaluate our approach in a semi-synthetic environ-

ment using real data (Yelp restaurants) and simulated responses. Our results demonstrate the ability of strategically-aware retraining to bolster diversity, and illustrate the importance of incentivizing the creation of diversity. All code is made publicly available at: `https://github.com/itayeilat/Performative-Recommendation`.

1.1. Related work

Diversity in recommendation. The literature on diversity in recommendation is extensive; here we present a relevant subset. Early approaches propose to diversify via re-ranking (Carbonell & Goldstein, 1998; Bradley & Smyth, 2001; Ziegler et al., 2005), an approach that remains to be in widespread use today (Abdollahpouri et al., 2019a). More recent methods include diversifying via functional optimization (Zhang & Hurley, 2008) or integration within matrix factorization (Su et al., 2013; Hurley, 2013; Cheng et al., 2017). Diversity has also been studied in sequential (Kim et al., 2019), conversational (Fu et al., 2021), and adversarial bandit (Brown & Agarwal) settings. The idea of using regularization to promote secondary objectives in recommendation has been applied for controlling popularity bias (Abdollahpouri et al., 2017), enhancing neutrality (Kamishima et al., 2014), and promoting equal opportunity (Zhu et al., 2021). For diversity, Wasilewski & Hurley (2016) apply regularization, but assume that the system has direct control over (latent) item features; this is crucially distinct from our setting in which the system can only indirectly encourage content creators to apply changes.

Strategic learning. There has been much recent interest in studying learning in the presence of strategic behavior. Hardt et al. (2016) propose *strategic classification* as a framework for studying classification tasks in which users (who are the targets of prediction) can modify their features—at a cost—to obtain favorable predictions. This is based on earlier formulations by Brückner & Scheffer (2009); Brückner et al. (2012), with recent works extending the framework to settings in which users act on noisy (Jagadeesan et al., 2021) or missing information (Ghalme et al., 2021; Bechavod et al., 2022), have broader interests (Levanon & Rosenfeld, 2022), or are connected by a graph (Eilat et al., 2022). Since we model content creators as responding to a scoring rule, our work pertains to the subliterature on *strategic regression*, in which user utility derives from a continuous function (Rosenfeld et al., 2020; Tang et al., 2021; Harris et al., 2021; Bechavod et al., 2022), and strategic behavior is often assumed to also affect outcomes (Shavit et al., 2020; Harris et al., 2022). Within this field, our framework is unique in that it considers content creators—rather than end-users—as the focal strategic entities. The main distinction is that this requires learning to account for the *joint* behavior of all strategic agents (rather than each individually), which even for linear score functions results in complex behav-

ioral patters (c.f. standard settings in which linearity implies uniform movement (Liu et al., 2022)). Regularization has been used to control incentives in Rosenfeld et al. (2020); Levanon & Rosenfeld (2021), but in distinct settings and towards different goals (i.e., unrelated to recommendation or diversity), and for user responses that fully decompose.

Performativity and incentives. The current literature on performative learning focuses primarily on macro-level analysis, such as providing sufficient global conditions for retraining to converge (Perdomo et al., 2020; Miller et al., 2021; Brown et al., 2022) or proposing general optimization algorithms (Mendler-Dünner et al., 2020; Izzo et al., 2021; Drusvyatskiy & Xiao, 2022; Maheshwari et al., 2022). In contrast, performativity in our setting emerges from micro-level modeling of strategic agents in a dynamic recommendation environment, and our goal is to address the specific challenges inherent in our focal learning task. Within recommendation, content creators (or 'supplier') incentives have also been studied from a game-theoretic perspective (Ben-Porat & Tennenholtz, 2018; Ben-Porat et al., 2019; 2020; Jagadeesan et al., 2022; Hron et al., 2022). Here, focus tends to be on notions of equilibrium, and the system is typically assumed to have direct control over outcomes (e.g., determining allocations or monetary rewards). Our work focuses primarily on learning, and studies indirect incentivization through a learned predictive rule.

2. Problem Setup

Our setup considers a recommendation platform consisting of m users and n items. Items are described by feature vectors $x_j \in \mathbb{R}^d$, $j \in [n]$, and each item x_j is owned by a (strategic) content creator j. The goal of the system is to learn latent vector representations $u_i \in \mathbb{R}^d$ for each user $i \in [m]$ that are useful for recommending relevant items. As in Hron et al. (2022), we assume all features are constrained to have unit ℓ_2 norm, $\|u\| = \|x\| = 1$ (i.e., lie on the unit sphere). This ensures equal treatment across items (by the system) and users (by content creators), and prevents features from growing indefinitely due to strategic updates.

The system makes recommendations by ranking items for each user i using a personalized score function $f_i(x)$ as:

$$\boldsymbol{r}_i = \text{rank}(f_i(x_1), \ldots, f_i(x_n)) \quad \forall i \in [m] \quad (1)$$

where $f_i(x) = f(x; u_i)$ rely on learned user representation vectors u_i. For a list of items X we denote in shorthand $\boldsymbol{r}_i = \text{rank}(f(X; u_i))$. As in most works on strategic learning (e.g., Hron et al., 2022; Jagadeesan et al., 2022; Carroll et al., 2022), we consider linear score functions $f(x; u_i) = u_i^\top x$. Overall, the goal of the system is to learn good $f_1, \ldots, f_n$ from data, where $u_1, \ldots, u_n$ are the learned parameters.

Learning objective. We measure ranking quality using the standard measure of NDCG evaluated on the top k items,

defined as follows. Consider a list of items with relevance scores $\boldsymbol{y} = (y_1, \ldots, y_n)$. Let $\boldsymbol{r} = (r_1, \ldots, r_n)$ be a ranking, and denote by $\boldsymbol{r}(\ell)$ the index of the ℓth-ranked item in $\boldsymbol{r}$. Then the top-k discounted cumulative gain (DCG) is:

$$\text{DCG}_{@k}(\boldsymbol{y}, \boldsymbol{r}) = \sum_{\ell=1}^{k} \frac{2^{y_{\boldsymbol{r}(\ell)}} - 1}{\log(1 + \ell)} \quad (2)$$

where $2^{y_j} - 1$ measures the 'gain' in relevance from having item $j = r(\ell)$ in the top k, and $\log(1 + \ell)$ 'discounts' its rank. NDCG is then obtained by normalizing relative to the optimal ranking $\boldsymbol{r}^* = \text{argmax}_{\boldsymbol{r}} \text{DCG}_{@k}(\boldsymbol{y}, \boldsymbol{r})$. The primary goal of the system is therefore to learn user representations $\{u_i\}_{i=1}^n$ that optimize average top-k NDCG:

$$\max_{u_1, \ldots, u_m} \frac{1}{m} \sum_{i=1}^{m} \text{NDCG}_{@k}(\boldsymbol{y}_i, \boldsymbol{r}_i) \quad (3)$$

where $\boldsymbol{r}_i$ is the ranking of items for user i according to $f(x; u_i)$. For learning, we will assume that the system has access to relevance labels y_{ij} for some user-item pairs (i, j), and the goal is to generalize well to other pairs.

Item diversity. In addition to ranking accuracy, we will also be interested in measuring and promoting diversity across recommended items. Here we consider intra-list diversity (Ziegler et al., 2005; Vargas & Castells, 2011; Antikacioglu et al., 2019) and focus primarily on cosine similarity as a metric, $\cos(x, x') = \frac{x^\top x'}{\|x\|\|x'\|}$, which is appropriate for comparing unit-norm features (Ekstrand et al., 2014; Hron et al., 2022).[1] See Appendix E.4 for an extension of our approach to entropy-based similarity.

For a list of items $X = (x_1, \ldots, x_K)$ and corresponding ranking $\boldsymbol{r}$, diversity for the top-k items is defined as:

$$\text{div}_{@k}(X, \boldsymbol{r}) = \frac{1}{k(k-1)} \sum_{j,\ell=1}^{k} 1 - \cos(x_{\boldsymbol{r}(j)}, x_{\boldsymbol{r}(\ell)}) \quad (4)$$

which takes values in $[0, 1]$ (Smyth & McClave, 2001).

Recommendation graph. In our setting, each user i is associated with a list of K_i potentially-relevant items, denoted $X_i \subseteq [n]$, and the system's goal is to choose a subset of $k \leq K_i$ items to recommend as a ranked list.[2] Since the same items can appear in multiple lists, it will be useful to consider users and items through a bipartite graph $G = (U, X, E)$, where $(i, j) \in E$ if item j is in user i's list of candidate items. We will also denote $U_j = \{i \in [m] \mid (i, j) \in E\}$ and $m_j = |U_j|$. As we will see, the graph plays a key role in determining the system's potential for encouraging diversity.

---

[1] Since features are normalized, we have $\cos(x, x') = x^\top x'$.
[2] This is also known as second-stage recommendation; see e.g. Ma et al. (2020); Hron et al. (2021); Wang & Joachims (2022).
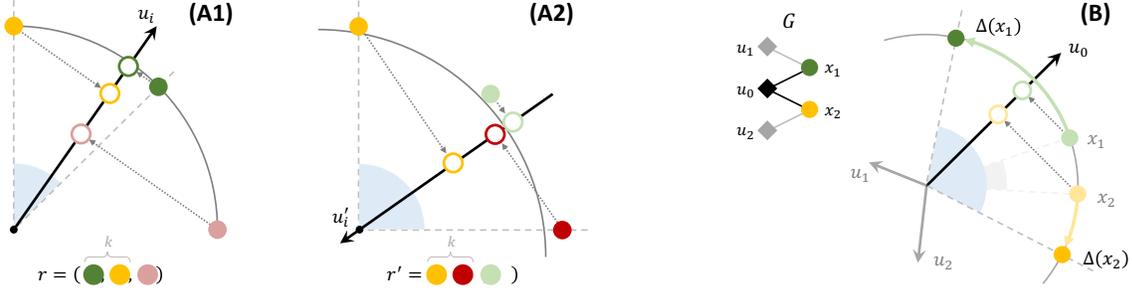
Figure 1. **(A) Tradeoff between accuracy and diversity:** Consider three items with high (●), medium (●), and low (●) relevance. Learned user embedding u_i obtains high $\text{NDCG}_{@2}$ (*projected points*) but low $\text{div}_{@2}$ (*blue sector*) **(A1)**. Conversely, u_i' (note flipped orientation) obtains high diversity, but at the cost of reduced accuracy **(A2)**. **(B) Creating diversity with strategic incentives:** Consider two items and three users connected by the graph G. Initially, both items are similar (*light circles*), and diversity is low. However, u_1 and u_2 incentivize x_1 and x_2, respectfully, to move away from each other, results in a more diverse set for u_0; in this case, without compromising accuracy.

2.1. Strategic content creators

Our key modeling assumption is that items are owned by strategic *content creators* (or 'suppliers') whose aim is to maximize exposure to their items (Hron et al., 2022). Content owners act to increase their item's score $s(x_j)$, which is an average over the scores of potentially relevant users:

$$s(x_j) = \text{avg}\{f(x_j; u_i) \mid i \in U_j\} \quad (5)$$

To preserve equity across content creators, we use spherical averaging over user representations u_i to maintain unit norm[3], which pertains to the following form:

$$s(x_j) = \tilde{v}_j^\top x_j, \quad \text{where } \tilde{v}_j = \frac{v_j}{\|v_j\|}, \ v_j = \frac{1}{m_j}\sum_{i \in U_j} u_i$$

This can be taken to mean that the system reveals normalized scores, so that utility for item j derives from $\tilde{v}_j$ which describes an 'average' user representing all $i \in U_j$.

Following the general formalism of strategic classification (Hardt et al., 2016), we assume content creators can modify their item's features, at a cost, and in response to the learned predictive model. Given a known cost function $c(x, x')$, content creators modify items via the *best response* mapping:

$$x_j^f = \Delta_f(x_j) \triangleq \underset{x':\|x'\|=1}{\text{argmax}} \ s(x') - \alpha c(x_j, x') \quad (6)$$

where the norm constraint ensures that modified items remain on the unit sphere, and throughout we consider quadratic costs, $c(x, x') = \|x - x'\|_2^2$. The scaling parameter $\alpha \geq 0$ will allow us to vary the intensity of strategic updates: when α is small, modifications are less restrictive and so x_j^f can move further away from x_j, and vice versa.

We consider item modification to be 'real', in the sense that changing x can cause y to also change. Following Shavit &

---

[3]To see why normalizing v is important, consider an item j with two users: if u_1, u_2 are close, then v_j will have a similar norm, but if u_1, u_2 are spread out, $\|v_j\|$ can be significantly smaller.

Moses (2019); Rosenfeld et al. (2020), we assume labels are determined by an unknown stochastic ground-truth function f^* which determines personalized relevance for any counterfactual item x as $y = f^*(x; u_i^*)$, where $\{u_i^*\}_{i=1}^m$ are ground-truth user preferences (which are unknown to the learner).

Strategic behavior and diversity. Eq. (6) reveals how the system can drive incentives: since each content creator j acts to make their item more aligned with $\tilde{v}_j$, the system can set the u_i (which together compose all $\tilde{v}_j$) to induce $\tilde{v}_j$-s that vary in their orientation; this incentivizes different content creators to move towards different directions—thus creating diversity, whose potential growth rate is mediated by α. Note that even linear f can incentive different items to move in different directions, since (i) due to norm constraints, items will not necessarily move in the direction of the gradient of f; and more importantly, (ii) since different items appeal to different users, each $\tilde{v}_j$ defines a utility function $s(x_j) = f(x_j; \tilde{v}_j)$ that is distinct for item j. Nonetheless, the $\tilde{v}_j$ are not disjoint; the connectivity structure in G forms dependencies across the u_i, which introduce correlations in how items can jointly move (see Fig. 1 (B)).

2.2. Interaction dynamics

We will be interested in studying how learning affects ranking accuracy and diversity over time. As noted, we focus on retraining dynamics, where at each round t the system retrains its predictive model f^t on current data (x^t, y^t), which in our case, is based on strategic responses to the previous model, $x_j^t = \Delta_{f^{t-1}}(x_j^{t-1})$. We think of item modification as a process which takes time: In the initial portion of round t, users remain to observe x_j^t, on which f^t was trained; but exposure to f^t incentivizes change, and so after some time has passed, content creators publish the modified x_j^{t+1}, for which users provide fresh labels y_{ij}^{t+1}. Once data has been collected for all modified inputs, the system retrains.

3. Learning and Optimization

The conventional approach for learning to recommend relies on training predictive models to correctly rank items by their relevance. Then, to promote diversity, a post-hoc procedure is typically used to re-rank $\boldsymbol{r}$, which in our setting can affect diversity by determining which items appear in the top k.

The main drawback of re-ranking is that its heuristic nature means that diversifying the list might reduce its relevance, and can cause NDCG to deteriorate substantially. As an alternative, here we pursue a more disciplined approach, in which we directly optimize the joint objective:

$$\max_{u_1,\ldots,u_m} \frac{1}{m} \sum_{i=1}^{m} \mathrm{NDCG}_{@k}(\boldsymbol{y}_i, \boldsymbol{r}_i) + \lambda \, \mathrm{div}_{@k}(X_i, \boldsymbol{r}_i) \quad (7)$$

where $\boldsymbol{r}_i = \mathrm{rank}(f(X_i; u_i))$ is the ranking of items in X_i for user i, and λ trades off between accuracy and diversity. With div as regularization, we can tune λ to obtain a desired balance, or maximize diversity under accuracy constraints. In our experiments we tune λ to achieve a predetermined level of NDCG (e.g., 0.9); in this case, regularization serves as a criterion for choosing the most diverse model out of all sufficiently-accurate models. We next describe our approach for optimizing Eq. (7), which sets the ground for our strategically-aware objective.

3.1. Optimization

We propose to optimize Eq. (7) by constructing a differentiable proxy objective, to which we can then apply gradient methods. The key challenge is that Eq. (7) relies on a ranking operator (i.e., for computing NDCG and top-k), which is non-differentiable. Our approach adopts and extends Pobrotyn & Bialobrzeski (2021), and makes use of the differentiable sorting operator introduced in Grover et al. (2019). First, consider NDCG. For the numerator, note that the ranking operator $\boldsymbol{r}(\cdot)$ can be implemented using a corresponding permutation matrix P, i.e., $y_{\boldsymbol{r}(\ell)} = (Py)_\ell$. To differentiate through $\boldsymbol{r}$, we replace P with a 'smooth' row-wise softmax permutation matrix, $\hat{P}$. We denote by $\hat{\boldsymbol{r}}$ the corresponding soft ranking, computed as $\hat{\boldsymbol{r}} = (\hat{P} \odot Q)^\top \mathbf{1}$, where $Q_{ij} = i \, \forall j$, $\odot$ is the Hadamard product, and $\mathbf{1}$ is a vector of 1-s. The denominator for $\mathrm{NDCG}_{@k}$ requires accessing indexes in y using explicit entries in $\boldsymbol{r}$; applying $\hat{\boldsymbol{r}}$ instead gives a weighted combination of y-s, with most mass concentrated at the correct $\boldsymbol{r}(\ell)$ when $\hat{\boldsymbol{r}}$ is a good approximation. The summation term is implemented using a soft top-k operator, which we obtain by applying an element-wise scalar sigmoid to $\boldsymbol{r}$ as:

$$\mathbb{1}\{r \le k\} \approx \sigma_\tau(k - \hat{\boldsymbol{r}}) \quad (8)$$

For the diversity term, note that $\cos$ is naturally differentiable. To make the entire $\mathrm{div}_{@k}$ differentiable, we use the soft top-k operator on item pairs via:

$$\frac{1}{k(k-1)} \sum_{j,j'}^{n} \sigma_\tau(k - \hat{\boldsymbol{r}}_j) \sigma_\tau(k - \hat{\boldsymbol{r}}_{j'}) \left(1 - x_j^\top x_{j'}\right) \quad (9)$$

3.2. Strategically-aware learning

Although Eq. (7) accounts for diversity in learning, it does so *reactively*, in a way that is tailored to the previous time step. Since the learned f incentivizes content creators to modify items, we propose to promote diversity *proactively* by anticipating their strategic responses. To do this, we replace X_i with the anticipated $X_i^f = \Delta_f(X_i) = \{\Delta_f(x)\}_{x \in X_i}$ to get our strategically-aware objective:

$$\max_{u_1,\ldots,u_m} \frac{1}{m} \sum_{i=1}^{m} \mathrm{NDCG}_{@k}(\boldsymbol{y}_i, \boldsymbol{r}_i) + \lambda \, \mathrm{div}_{@k}(X_i^f, \boldsymbol{r}_i^f)$$
$$(10)$$

where $\boldsymbol{r}_i^f = \mathrm{rank}(f(X_i^f; u_i))$ is the anticipated ranking. Eq. (10) optimizes for pre-modification NDCG, but promotes post-modification diversity; this is the mechanism through which learning can incentivize the creation of diversity.[4]

The challenge in optimizing Eq. (10) is twofold: (i) Δ_f is an argmax operator, which can be non-differentiable, and (ii) Δ_f depends on f both internally (by determining utility) and externally (in the top-k operator of $\mathrm{div}_{@k}$). Fortunately, for our modeling choices, Δ_f can be solved in a differentiable closed form. Using KKT conditions, we can derive:

$$\Delta_f(x_j) = \frac{\tilde{v}_j + 2\alpha x_j}{|\tilde{v}_j + 2\alpha x_j|} \quad (11)$$

Proof in Appendix A.1. Plugging Eq. (11) into Eq. (10) gives us our final differentiable strategic learning objective.

4. Diversity via Strategic Incentives

The reliance of the utility of content creators on the learned f provides the system with potential power for shaping incentives. Here we analyze when this potential can materialize in a simplified setting that focuses exclusively on maximizing diversity, for a single time step with no cost restrictions (i.e., set $\alpha = 0$). This removes any constraints on u_i that may arise from accuracy considerations, and serves as a convenient substitute for lengthy strategic dynamics.

Our main object of interest in the analysis is the recommendation graph G, viewed as input to the learning algorithm. As we show, whether diversification through incentivization is possible (or not) depends on properties of the graph. We

---

[4]While in principle it may be possible to also consider future NDCG, note this necessitates reasoning about how deploying f affects future y, which is a challenging causal inference task.

first consider the graph of a single item list X and all related users (i.e., users u_i for which some $x_j \in X_i$ is also in X), and then proceed to general graphs over multiple lists.

We begin with a negative result for single lists.

Proposition 1. *Let $X = \{x_1, x_2\}$. If both items have fully overlapping users (i.e., $U_1 = U_2$), then for any f, $\mathrm{div}(X^f, \boldsymbol{r}^f) = 0$. Hence, diversity cannot be incentivized.*

Proof in Appendix B.1. Prop. 1 shows how similar users induce similar incentives, resulting in $\Delta_f(x_1) = \Delta_f(x_2)$. Extending the result to larger item sets and multiple users is straightforward, and implies the following: if the same list of potentially-relevant items is associated exclusively with the same group of users, then strategic behavior is bound to nullify diversity entirely—regardless of any system efforts. Conversely, Prop. 1 hints that to diversify items, it is necessary to start out with some initial variation in the assignment of users to items. But how much user variation is needed? Our next result shows that for a single item list, when no other considerations are present, minimal differentiation in users is sufficient for obtaining maximal item diversity.

Proposition 2. *Let $X = \{x_1, x_2\}$. If U_1, U_2 differ only in a single user, i.e., if $U_1 = U \cup \{u_1\}$ and $U_2 = U \cup \{u_2\}$ for some U, u_1, u_2 with $|U| > 1$, then there exists an f which obtains maximal diversity, i.e., $\mathrm{div}(X^f, \boldsymbol{r}^f) = 1$.*

The proof is constructive, and appears in Appendix B.2. Prop. 2 shows that, under lenient conditions, incentivization can drive diversity to its greatest possible extend. Generally, and under more realistic considerations (e.g., accuracy and cost constraints), we expect that greater differentiation is likely necessary, even for lower gains in diversity.

We now move to considering graphs for multiple item lists. This introduces dependencies: if some item x_j appears in two distinct but partially-overlapping user lists $X_i, X_{i'}$, then this restricts the possible values that the embeddings $u_i, u_{i'}$ can take. As such, Prop. 2 cannot simply be applied to each list in G independently, since f cannot be tailored to maximize diversity for X_i without affecting $X_{i'}$, for which f may not be optimal. Our final result shows that, despite such dependencies, significant diversity is still attainable.

Proposition 3. *Let $K = 2$, then for any $\epsilon > 0$ and any N, there exists a graph G over N distinct lists and a corresponding f s.t. the average diversity is at least $(1 - \epsilon)(1 - 3/N)$.*

Proof in Appendix B.3, and relies on a construction that simultaneously (i) decouples users across lists, and (ii) permits maximal diversity within lists. Prop. 3 shows that the potential for incentivizing for diversity grows towards the near-optimal value of $1 - \epsilon$ quickly, in terms of the number of lists N—and to the degree that the graph permits.

5. Synthetic Experiments

Our previous section showed that, under favorable conditions, it is theoretically possible to generate significant diversity through incentivization. In this section we empirically demonstrate, in a series of increasingly-complex synthetic tasks, how our strategically-aware learning approach (Eq. (10)) can encourage diversification in practice. Each task is designed to explore a different factor in our setup, and to shed light on how accuracy and diversity trade off as λ is varied. Appendix C includes additional results.

Experimental setup. We set $n = 200, m = 50, k = K_i = 10$ for all i, and fix $d = 2$ so that features x, u can be easily visualized as angles. Item features x are sampled from $\mathcal{N}((1/\sqrt{2}, 1/\sqrt{2}), \sigma_x^2 I)$. We use $f^*(x; u_i^*) = 2^{(u_i^*)^\top x}$, where ground truth user preferences u^* are sampled from $\mathcal{N}((1/\sqrt{2}, 1/\sqrt{2}), \sigma_{u^*}^2 I)$. All features are normalized post-sampling. We use $\sigma_x = 1$ and $\sigma_{u^*} = 0.1$, but in some settings vary them to control dispersion. All results are averaged over 100 random repetitions (when applicable).

5.1. The role of variation in user item lists

Our first experiment investigates the importance of variation across user item lists, which complements Sec. 4. We begin with a graph composed of five mutually-exclusive fully-connected subgraphs, exhibiting full overlap; then, we 'shuffle' N edges across subgraphs, for increasing $N \in \mathbb{N}$—which decreases user overlap (at $N \approx 1000$ edges are approx. uniform). Figure 2 (left) shows NDCG (bottom) and diversity (top) for a range of λ. For $N = 0$ (full overlap), diversity is zero for all λ, in line with Prop. 1. As N grows, diversity increases, but at the cost of reduced NDCG, which is more pronounced for larger λ. Note how only minimal overlap (e.g., $N = 100$ for $\lambda = 1$) suffices for generating considerable diversity, which rises sharply once $N > 0$. This suggests our approach can utilize the capacity for diversity implied by Prop. 2, even under accuracy constraints.

5.2. The role of variation in true user preferences

When learning aims primarily for accuracy, training encourages each u_i to be oriented towards its u_i^*. For diversity, this acts as a constraint which restricts the capacity of f to diversify. Here we study the role of variation in u_i^* as a mediator in this process. Figure 2 (center) shows NDCG (bottom) and diversity (top), both in absolute values and relative to pre-update diversity, for varying σ_{u^*} and for a range of λ. Here we sample edges uniformly, and consider a single time step with $\alpha = 0$. Without regularization ($\lambda = 0$), strategic updates cause diversity to drop to zero, even when user preferences are reasonably dispersed ($\sigma_{u^*} = 1$). However, even mild regularization ($\lambda = 0.05$) suffices for obtaining significant diversity through incentivization, which becomes
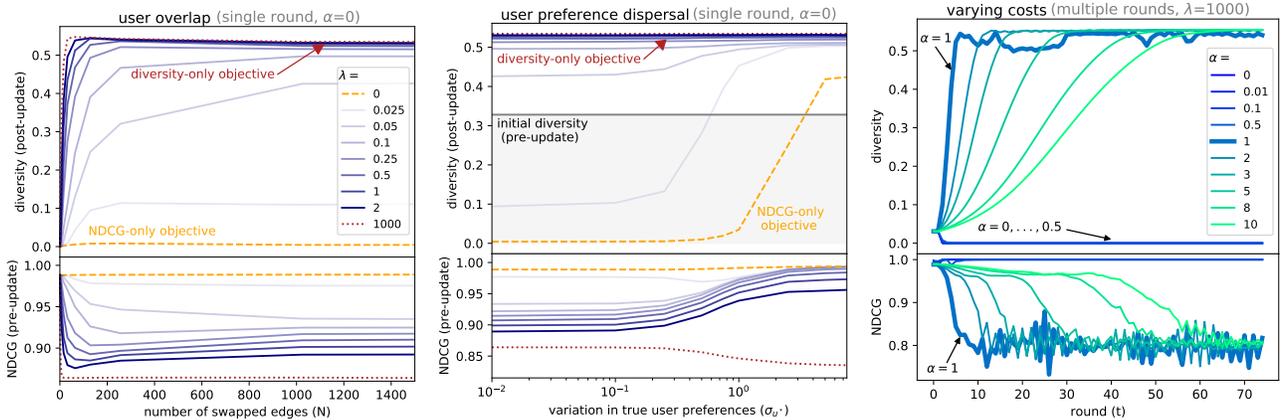
Figure 2. **Synthetic experiments. (Left)** As user overlap decreases (larger N), our approach is quickly able to incentivize diversity; this is pronounced for larger λ. **(Center)** For any dispersion level of true user preferences (u^*), even mild regularization ($\lambda > 0$) suffices for our approach to improve diversity; without it ($\lambda = 0$), diversity may plummet. **(Right)** For low cost scales ($\alpha < 1$), changes are large, and learning is unable to diversify. But once $\alpha \geq 1$, this changes sharply, with Larger α requiring more time for diversity to smoothly form.

more pronounced as λ grows; for all $\lambda > 0.1$, diversity is high, and in effect remains fixed. This suggests our model can effectively utilize natural variation in user preferences. Increased diversity comes at the cost of NDCG, but this diminishes quickly for larger dispersion.

5.3. The role of time vs. modification costs

We now turn to examining the temporal formation of diversity through retraining dynamics, as mediated by modification costs α. When α is large, content creators can only apply small changes to x at each round (and vice versa for small α). On the one hand, small steps suggest that diversity may require time to form; but on the other, note that small steps also allow the system to intervene with high fidelity and direct incentives throughout, and hence to gradually 'steer' behavior towards diversification. Figure 2 (right)
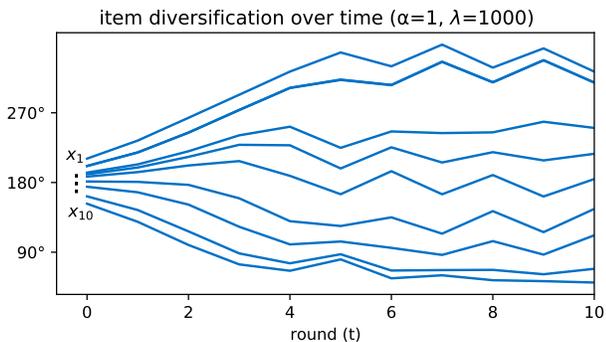
Figure 3. Dispersal of items $x_i \in \mathbb{R}^2$ over time.

shows diversity (top) and NDCG (bottom) for increasing α and over multiple retraining rounds. We set λ to be large so that learning is geared primarily towards diversity. When α is small (here, < 1), diversity quickly drops to zero—as in Prop. 1. In contrast, $\alpha = 1$ exhibits a sharp transition, in which diversity quickly rises. Fig. 3 visualizes for a set of items in $\mathbb{R}^2$ how they quickly become dispersed. Larger α entail similarly high diversity: here the process is slower—since update steps are smaller, but also more stable—since the system has finer control over each step; c.f. $\alpha = 1$, where NDCG fluctuates.

6. Experiments on Real Data

We now turn to evaluating our approach on real data. Here we study how NDCG and diversity evolve over time under different learning methods and experimental conditions. See Appendix D for additional details, and Appendix E for extended results (E.1,E.2), a sensitivity analysis to misspecification of α (E.3), and additional similarity metrics (E.4).

Data. Our experimental setup is based on the restaurants portion of the Yelp dataset[5], which includes user-submitted restaurant reviews. We focus on users having at least 100 reviewed restaurants. For each user i, we construct the list of potential items X_i to include the 40 most popular restaurants of those reviewed by i, which amounts to 236 users and 1,520 restaurants in total. We elicit $d = 43$ restaurant features (e.g., cuisine type, noise level) to be used by the system for learning. We also elicit 'ground truth' user features u_i^* used for optimizing $f^*(x', u_i^*)$, which is trained to

---

[5] https://www.yelp.com/dataset/download
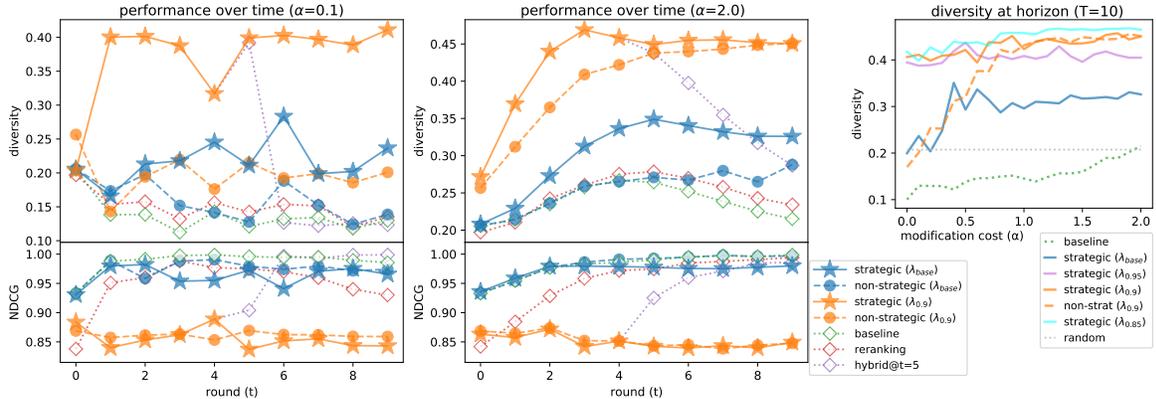
Figure 4. **Experiments on Yelp data. (Left+center)** Diversity and NDCG over rounds for different methods and target NDCG values, for loose ($\alpha = 0.1$; left) vs. restricted ($\alpha = 2.0$; center) strategic updates. Our `strategic` approach consistently achieves significant and sustained improvement in diversity, with minimal loss in accuracy. **Right** Diversity after $T = 10$ rounds for increasing costs scales α.

predict the likelihood that i will review x, interpreted here as relevance y. The labeling function f^* is used only for determining updated relevancies y' for modified items x'; neither f^* nor the u_i^* are known to the learner. We ensure f^* is distinct from learnable functions f by several means: (i) f^* is a fully-connected deep network, whereas f are linear; (ii) f^* is trained using true user features u_i^*, which are unobserved for f; (iii) f^* is trained on considerably more data, and of which the data used for training f is a non-representative subset; and (iv) f is trained on labels that are modified to emphasize highly-rated items. Full details in Appendix D.2.

Setup and learning. We consider top-10 recommendation (i.e., $k = 10$), and evaluate performance over $T = 10$ rounds of retraining and corresponding strategic updates. For training, we assume that at each round the system has access to 30 items per user, randomly selected (out of the 40) per round; of these, a random 20 are used for training, and the remaining 10 are added for validation (tuning λ and early stopping). Test performance is evaluated on all 40 items. Since the test set includes additional items (compared to training), f must learn to generalize well to new content at each step. Since adding items also changes the graph, and since the graph determines strategic responses—f must also learn to generalize to new forms of strategic updates.

Methods and evaluation. In line with our dynamic setup (Sec. 2.2), all methods considered aim primarily at optimizing 'current' NDCG (i.e., at time t maximize NDCG on $x^{(t)}$), but differ in how (and if) they promote diversity. These include: (i) a `non-strategic` approach which regularizes for current diversity on non-strategic inputs $x^{(t)}$ (Eq. (7)), (ii) our `strategic` approach, which regularizes for future diversity on the anticipated $x^{(t+1)} = x^{f_t}$ (Eq. (10)), (iii) an accuracy-only `baseline`, which does not promote diversity (by setting $\lambda = 0$), (iv) `re-ranking` using the popular MMR

diversification procedure (Carbonell & Goldstein, 1998), and (v) a `hybrid@t` approach, which runs `strategic` for $t = 5$ rounds, and then 'turns off' regularization.

Since `non-strategic` and `strategic` are designed to balance NDCG and diversity, for a meaningful comparison, in each experimental condition we fix a predetermined target value for NDCG, tune each method at each round to achieve this target (using λ, on the validation set, and up to tolerance 0.01), and compare the resulting diversity. We use the notation λ_β to mean that λ was tuned for the target β. To compare with `baseline`, in one condition we set λ to the largest value that maintains the same NDCG as the baseline (for which $\lambda = 0$), denoted λ_{base}. Note that due to performativity, data at time t depends on the learned model at time $t-1$. Because our dynamics are *stateful*, results are path-dependent, and so comparisons must be made across full trajectories—and cannot be made independently at each time point.

6.1. Diversity over time

Our first experiment studies how diversity evolves over time for different NDCG targets. Fig. 4 (left) shows diversity (top) and NDCG (bottom) per round for $\alpha = 0.1$, which permits significant (yet restricted) strategic updates. Results show that for `baseline`, which does not actively promote diversity, average diversity decreases over time by roughly 40%. Adding MMR, which re-ranks for diversity, improves only marginally. In the λ_{base} condition, diversity for `non-strategic` improves only slightly compared to `baseline`, mostly at the onset. In contrast, our `strategic` approach is able to consistently maintain (and even slightly increase) diversity over time, albeit at small occasional drops in test NDCG, as compared to `non-strategic` (≤ 0.02).[6]

---

[6]One possible reason for `baseline` to achieve NDCG$\approx$1 is that when diversity is extremely low, items are so similar (and

8
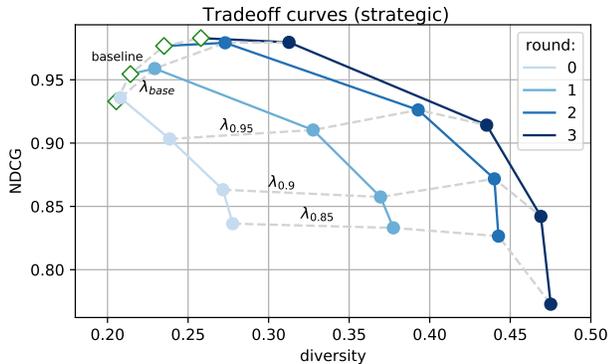
Figure 5. Pareto curves for `strategic`, per round (solid lines) and per λ (dashed).

In the $\lambda_{0.9}$ condition, `non-strategic` is able to preserve roughly 80% of diversity by sacrificing $\sim 13\%$ of the optimal NDCG. Meanwhile, and for the same loss in NDCG, `strategic` is able to quickly *double* the initial diversity—and sustain it, likely through means similar to those observed in Sec. 5.3. Note that sustaining diversity requires to actively promote it throughout; once regularization is switched off (`hybrid`), diversity immediately drops.

Fig. 4 (center) shows results for a similar setting, but using a larger $\alpha = 2$. In comparison, here diversity for all methods improves—but for `baseline` and `MMR`, this is only transient. Results are also smoother, and accumulate—for both improvement and deterioration—which is likely due to the fact that strategic movements are now more restricted. The biggest distinction here is that for $\lambda_{0.9}$, `non-strategic` eventually obtains the same level of diversity as `strategic`, this likely due to α being large: small updates mean that x and x^f are correlated, and so regularizing for $\text{div}(X, \boldsymbol{r})$ also improves $\text{div}(X^f, \boldsymbol{r}^f)$ to some extent. Here the advantage of `strategic` is that it improves much faster.

6.2. The role of modification costs.

To further examine the role of α, we compare performance across a range of cost scales and for multiple target NDCG values. Fig. 4 (right) shows diversity at $T = 10$. As can be seen, all methods benefit from increasing α. However, diversity for `baseline` remains below that of a `random` baseline which recommends k random items. For `strategic`, sacrificing fairly little NDCG ($\lambda_{0.95}$ and below) suffices for gaining significant diversity across *all* α. Results also show how α mediates the gap between `non-strategic` and `strategic`, which increases as α grows.[7]

---

hence similarly relevant) that choosing the top-k becomes trivial.

[7]To avoid clutter we plot `non-strategic` only for $\lambda_{0.9}$, which is comparable to Fig. 4 (left) and (center), but note that other target

6.3. Tradeoffs over time.

Fig. 5 shows Pareto curves for `strategic`, obtained by considering multiple λ and over several rounds. Results show that the entire tradeoff curve between NDCG and diversity improves over time. Paths for each target NDCG depict specific trajectories, and show how diversity can be created without sacrificing accuracy. This holds even when no compromise in accuracy is allowed: even though both `baseline` and λ_{base} maximize accuracy, our `strategic` approach is able to steer content creators towards more diverse content. This is achieved by utilizing the flexibility to choose u_i which improve both accuracy *and* diversity.

7. Discussion

Through the lens of performativity, our paper studies how a learning system can incentivize content creators to collectively form a more diverse inventory of items for recommendation. In this, we challenge the conventional view that diversity is merely a matter of which items to present (and which not), and argue that to fundamentally rectify the predisposition of modern recommendation systems to homogenize content, learning must (i) recognize that content *changes* over time, in part due to the strategic behavior of content creators, and (ii) capitalize on its power shape incentives and steer towards diversity. Our work joins others in taking a step towards studying recommendation systems as complex ecosystems in which users, creators, and the system itself act and react to promote their own goals and aspirations. From a learning perspective, this requires us to rethink the role that predictions play in the recommendation process, and consider its implications on user welfare.

Acknowledgements

References

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

Abdollahpouri, H., Burke, R., and Mobasher, B. Managing popularity bias in recommender systems with personalized re-ranking. In *The thirty-second international flairs conference*, 2019a.

Abdollahpouri, H., Mansoury, M., Burke, R., and Mobasher,

NDCG values exhibit qualitatively similar patterns.

B. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286*, 2019b.

Anderson, C. *The long tail: Why the future of business is selling less of more*. Hachette UK, 2006.

Antikacioglu, A., Bajpai, T., and Ravi, R. A new system-wide diversity measure for recommendations with efficient algorithms. *SIAM Journal on Mathematics of Data Science*, 1(4):759–779, 2019.

Bechavod, Y., Podimata, C., Wu, S., and Ziani, J. Information discrepancy in strategic learning. In *International Conference on Machine Learning*, pp. 1691–1715. PMLR, 2022.

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

Ben-Porat, O., Goren, G., Rosenberg, I., and Tennenholtz, M. From recommendation systems to facility location games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1772–1779, Jul. 2019. doi: 10.1609/aaai.v33i01.33011772. URL https://ojs.aaai.org/index.php/AAAI/article/view/4000.

Ben-Porat, O., Rosenberg, I., and Tennenholtz, M. Content provider dynamics and coordination in recommendation ecosystems. *Advances in Neural Information Processing Systems*, 33:18931–18941, 2020.

Bradley, K. and Smyth, B. Improving recommendation diversity. In *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland*, volume 85, pp. 141–152. Citeseer, 2001.

Brown, G., Hod, S., and Kalemaj, I. Performative prediction in a stateful world. In *International Conference on Artificial Intelligence and Statistics*, pp. 6045–6061. PMLR, 2022.

Brown, W. and Agarwal, A. Diversified recommendations for agents with adaptive preferences. In *Advances in Neural Information Processing Systems*.

Brückner, M. and Scheffer, T. Nash equilibria of static prediction games. In *Advances in neural information processing systems*, pp. 171–179, 2009.

Brückner, M., Kanzow, C., and Scheffer, T. Static prediction games for adversarial learning problems. *The Journal of Machine Learning Research*, 13(1):2617–2654, 2012.

Burke, R. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093*, 2017.

Carbonell, J. and Goldstein, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 335–336, 1998.

Carroll, M. D., Dragan, A., Russell, S., and Hadfield-Menell, D. Estimating and penalizing induced preference shifts in recommender systems. In *International Conference on Machine Learning*, pp. 2686–2708. PMLR, 2022.

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

Cheng, P., Wang, S., Ma, J., Sun, J., and Xiong, H. Learning to recommend accurate and diverse items. In *Proceedings of the 26th international conference on World Wide Web*, pp. 183–192, 2017.

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

Drusvyatskiy, D. and Xiao, L. Stochastic optimization with decision-dependent distributions. *Mathematics of Operations Research*, 2022.

Eilat, I., Finkelshtein, B., Baskin, C., and Rosenfeld, N. Strategic classification with graph neural networks. *arXiv preprint arXiv:2205.15765*, 2022.

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

Fu, Z., Xian, Y., Geng, S., De Melo, G., and Zhang, Y. Popcorn: Human-in-the-loop popularity debiasing in conversational recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 494–503, 2021.

Ghalme, G., Nair, V., Eilat, I., Talgam-Cohen, I., and Rosenfeld, N. Strategic classification in the dark. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Grover, A., Wang, E., Zweig, A., and Ermon, S. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1eSS3CcKX.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pp. 111–122, 2016.

Hardt, M., Jagadeesan, M., and Mendler-Dünner, C. Performative power. *arXiv preprint arXiv:2203.17232*, 2022.

Harris, K., Heidari, H., and Wu, Z. S. Stateful strategic regression. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Harris, K., Ngo, D. D. T., Stapleton, L., Heidari, H., and Wu, S. Strategic instrumental variable regression: Recovering causal relationships from strategic responses. In *International Conference on Machine Learning*, pp. 8502–8522. PMLR, 2022.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

Hron, J., Krauth, K., Jordan, M., and Kilbertus, N. On component interactions in two-stage recommender systems. *Advances in neural information processing systems*, 34: 2744–2757, 2021.

Hron, J., Krauth, K., Jordan, M. I., Kilbertus, N., and Dean, S. Modeling content creator incentives on algorithm-curated platforms. *arXiv preprint arXiv:2206.13102*, 2022.

Hu, R. and Pu, P. Helping users perceive recommendation diversity. In *DiveRS@ RecSys*, pp. 43–50, 2011.

Hurley, N. J. Personalised ranking with diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 379–382, 2013.

Izzo, Z., Ying, L., and Zou, J. How to learn when data reacts to your model: performative gradient descent. In *International Conference on Machine Learning*, pp. 4641–4650. PMLR, 2021.

Jagadeesan, M., Mendler-Dünner, C., and Hardt, M. Alternative microfoundations for strategic classification. In *International Conference on Machine Learning*, pp. 4687–4697. PMLR, 2021.

Jagadeesan, M., Garg, N., and Steinhardt, J. Supply-side equilibria in recommender systems. *arXiv preprint arXiv:2206.13489*, 2022.

Kaminskas, M. and Bridge, D. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7(1):1–42, 2016.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. Correcting popularity bias by enhancing recommendation neutrality. In *RecSys Posters*, 2014.

Kim, Y., Kim, K., Park, C., and Yu, H. Sequential and diverse recommendation with long tail. In *IJCAI*, volume 19, pp. 2740–2746, 2019.

Kunaver, M. and Požrl, T. Diversity in recommender systems–a survey. *Knowledge-based systems*, 123:154–162, 2017.

Levanon, S. and Rosenfeld, N. Strategic classification made practical. In *International Conference on Machine Learning*, pp. 6243–6253. PMLR, 2021.

Levanon, S. and Rosenfeld, N. Generalized strategic classification and the case of aligned incentives. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Liu, L. T., Garg, N., and Borgs, C. Strategic ranking. In *International Conference on Artificial Intelligence and Statistics*, pp. 2489–2518. PMLR, 2022.

Ma, J., Zhao, Z., Yi, X., Yang, J., Chen, M., Tang, J., Hong, L., and Chi, E. H. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, pp. 463–473, 2020.

MacKenzie, I., Meyer, C., and Noble, S. How retailers can keep up with consumers. *McKinsey & Company*, 18(1), 2013.

Maheshwari, C., Chiu, C.-Y., Mazumdar, E., Sastry, S., and Ratliff, L. Zeroth-order methods for convex-concave min-max problems: Applications to decision-dependent risk minimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 6702–6734. PMLR, 2022.

McNee, S. M., Riedl, J., and Konstan, J. A. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pp. 1097–1101, 2006.

Mendler-Dünner, C., Perdomo, J., Zrnic, T., and Hardt, M. Stochastic optimization for performative prediction. *Advances in Neural Information Processing Systems*, 33: 4929–4939, 2020.

Miller, J. P., Perdomo, J. C., and Zrnic, T. Outside the echo chamber: Optimizing the performative risk. In *International Conference on Machine Learning*, pp. 7710–7720. PMLR, 2021.

Mladenov, M., Creager, E., Ben-Porat, O., Swersky, K., Zemel, R., and Boutilier, C. Optimizing long-term social welfare in recommender systems: A constrained matching approach. In *International Conference on Machine Learning*, pp. 6987–6998. PMLR, 2020.

Owen, C. B. *Parameter estimation for the beta distribution.* Brigham Young University, 2008.

Perdomo, J., Zrnic, T., Mendler-Dünner, C., and Hardt, M. Performative prediction. In *International Conference on Machine Learning*, pp. 7599–7609. PMLR, 2020.

Pobrotyn, P. and Bialobrzeski, R. Neuralndcg: Direct optimisation of a ranking metric via differentiable relaxation of sorting. *ArXiv*, abs/2102.07831, 2021.

Qin, L. and Zhu, X. Promoting diversity in recommendation by entropy regularizer. In *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013.

Rosenfeld, N., Hilgard, A., Ravindranath, S. S., and Parkes, D. C. From predictions to decisions: Using lookahead regularization. *Advances in Neural Information Processing Systems*, 33:4115–4126, 2020.

Sha, C., Wu, X., and Niu, J. A framework for recommending relevant and diverse items. In *IJCAI*, volume 16, pp. 3868–3874, 2016.

Shavit, Y. and Moses, W. S. Extracting incentives from black-box decisions. *arXiv preprint arXiv:1910.05664*, 2019.

Shavit, Y., Edelman, B., and Axelrod, B. Causal strategic linear regression. In *International Conference on Machine Learning*, pp. 8676–8686. PMLR, 2020.

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

Smyth, B. and McClave, P. Similarity vs. diversity. In *International conference on case-based reasoning*, pp. 347–361. Springer, 2001.

Su, R., Yin, L., Chen, K., and Yu, Y. Set-oriented personalized ranking for diversified top-n recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 415–418, 2013.

Tang, W., Ho, C.-J., and Liu, Y. Linear models are robust optimal under strategic behavior. In *International Conference on Artificial Intelligence and Statistics*, pp. 2584–2592. PMLR, 2021.

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

Vargas, S. and Castells, P. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pp. 109–116, 2011.

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

Wang, L. and Joachims, T. User fairness, item fairness, and diversity for rankings in two-sided markets. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 23–41, 2021.

Wang, L. and Joachims, T. Fairness in the first stage of two-stage recommender systems. *arXiv preprint arXiv:2205.15436*, 2022.

Wasilewski, J. and Hurley, N. Incorporating diversity in a learning to rank recommender system. In *The twenty-ninth international flairs conference*, 2016.

Wu, W., Chen, L., and Zhao, Y. Personalizing recommendation diversity based on user personality. *User Modeling and User-Adapted Interaction*, 28(3):237–276, 2018.

Yin, H., Cui, B., Li, J., Yao, J., and Chen, C. Challenging the long tail recommendation. *arXiv preprint arXiv:1205.6700*, 2012.

Zhang, M. and Hurley, N. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pp. 123–130, 2008.

Zhu, Z., He, Y., Zhao, X., Zhang, Y., Wang, J., and Caverlee, J. Popularity-opportunity bias in collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 85–93, 2021.

Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pp. 22–32, 2005.

A. Optimization

A.1. Closed form expression for best response Δ_f

Proof. To compute an item's best-response update $\Delta_f(x)$, we must solve the following constrained problem:

$$\max_{x'} v^T x' - \alpha c(x', x) \quad \text{s.t} \quad ||x||_2 = 1$$

where $c(x, x') = ||x' - x||_2^2$. We solve for x' using Lagrangian analysis. First, we square the constraint (which does not change the condition itself). Define the Lagrangian as follows:

$$L(x', \lambda) = v^T x' - \alpha ||x' - x||_2^2 + \lambda[||x'||_2^2 - 1]$$

Next, to find the minimum of L, derive with respect to x', and compare to 0:

$$v - 2\alpha(x' - x) + 2\lambda x' = 0$$
$$v + 2\alpha x = 2\alpha x' - 2\lambda x'$$
$$\frac{v + 2\alpha x}{2\alpha - 2\lambda} = x'$$

Plugging x' into the original constraint gives:

$$\frac{|v + 2\alpha x|}{|2\alpha - 2\lambda|} = 1$$
$$|v + 2\alpha x| = 2\alpha - 2\lambda$$
$$|v + 2\alpha x| - 2\alpha = -2\lambda$$

Finally, plugging λ into the expression for x'_i obtains:

$$\frac{v + 2\alpha x}{|v + 2\alpha x|} = x'$$

B. Proofs

B.1. Proposition 1

Proof. As both items have the same set of users, the normalized average of user embeddings is the same for both items, i.e., $\tilde{v}_1 = \tilde{v}_2 = \tilde{v}$. Since $\alpha = 0$, the best response (Eq. (11)) is given by $\Delta_f(x_j) = \tilde{v}$ for both $j = 1, 2$. Hence, both items are modified in the same way, and so diversity is by definition zero. $\square$

B.2. Proposition 2

We begin with a useful lemma.

Lemma 4. *For any $n > 1$ there exists a set of n unit-norm user embeddings $U = \{u_1, \ldots, u_n\}$ for which:*

$$\sum_{u_i \in U} u_i = \overrightarrow{0}$$

where $\overrightarrow{0}$ is the zero vector.

Proof. We prove for $d = 2$; for $d > 2$, we set the first two coordinates accordingly, and the other $d - 2$ coordinates are set to zero. For $n = 2$, we define $U_2 = \{u_1, u_2\}$ as follows:

$$u_1 = (-1, 0), \quad u_2 = (1, 0)$$

Note that indeed $\sum_{u_i \in U_2} u_i = u_1 + u_2 = (0, 0)$.

For $n = 3$, we define $U_3 = \{u_1, u_2, u_3\}$ as follows:

$$u_1 = (\frac{1}{2}, \frac{\sqrt{3}}{2}), \quad u_2 = (\frac{1}{2}, -\frac{\sqrt{3}}{2}), \quad u_3 = (-1, 0)$$

Here as well $\sum_{u_i \in U_3} u_i = u_1 + u_2 + u_3 = (0, 0)$.

Observation: $\forall n > 1 \; \exists a, b \geq 0$ for which $n = 2a + 3b$.

Thus, for any $n > 1$, we define U to include a copies of U_2 and b copies of U_3. Using this construction, we get $\sum_{u_i \in U} u_i = (0, 0)$, as required. □

We now return to proving the proposition. The general idea is to define two embeddings that are maximally distinct, and ensure all others do not interfere with their orientation.

Proof. First we prove that for $|U| > 1$ diversity is one. According to Lemma 4, there exists a set of user embedding vectors U of size $|U_1 \cap U_2|$ that holds:

$$\sum_{u \in U} u = \vec{0}$$

We define an f that assigns to each user in $|U_1 \cap U_2|$ a vector in U (one to one map). Since the sum of the vectors in U is 0 it holds that $\tilde{v}_1 = u_1$ and $\tilde{v}_2 = u_2$. Since $\alpha = 0$, there are no modification costs, and the strategic response of items according to Eq. (11) is $\Delta_f(x_1) = u_1$ and $\Delta_f(x_2) = u_2$. In order to create maximal diversity between $\Delta_f(x_1)$ and $\Delta_f(x_2)$, we can use any choice of vectors u_1 and u_2 that satisfy $u_1 = -u_2$. □

B.3. Proposition 3

We prove for $d = 2$; for $d > 2$, we set the first two coordinates accordingly, and the other $d - 2$ coordinates are set to zero. This allows us to work with angles between user embeddings u. We use 'x' and 'y' to refer to the corresponding Cartesian components of angles.

Consider the following graph and user vector embeddings. For each $i < N$, define user i's list of candidate items X_i to include the items x_i, x_{i+1}. For user N, the items in X_N are x_N, x_1.

Let $0 < \delta$. Define the following embedding: For every odd $i < N$, set $u_i = 90 - i\delta$; For every even $i < N$, set $u_i = 270 - i\delta$. For $i = N$ we define $u_N = 270 + \delta$. Note that this gives $x_1 = (1, 0)$ since the two users that influence x_1 are u_1 and u_N, whose average gives a vector with coordinates:

$$
\begin{aligned}
\text{x}: \quad & \frac{1}{2}(cos(270 + \delta) + cos(90 - \delta)) = \\
& \frac{1}{2}(cos(-90 + \delta) + cos(90 - \delta)) = \\
& \frac{1}{2}(cos(-(90 - \delta)) + cos(90 - \delta)) = \\
& \frac{1}{2}(cos((90 - \delta)) + cos(90 - \delta)) = \\
& cos(90 - \delta) \\
\text{y}: \quad & \frac{1}{2}(sin(270 + \delta) + sin(90 - \delta)) = \\
& \frac{1}{2}(sin(-90 + \delta) + sin(90 - \delta)) = \\
& \frac{1}{2}(-sin(90 - \delta) + sin(90 - \delta)) = \\
& = 0
\end{aligned}
$$

This item's y value is 0; hence, do due unit norm constraints, we have $x_1 = (1, 0)$.

We now calculate the response of x_i when i is even for all $1 < i < N$. The users associated with x_i are:

$$u_{i-1} = 90 - (i-1)\delta, \qquad u_i = 270 - i\delta$$

We average the two vectors by averaging each coordinate x, y:

$$\text{x}: \quad \frac{1}{2}(cos(270 - i\delta) + cos(90 - (i-1)\delta)) =$$

$$cos(\frac{360 - (2i-1)\delta}{2})cos(\frac{180 - \delta}{2}) =$$

$$cos(180 - \frac{2i-1}{2}\delta)cos(90 - \frac{\delta}{2}) =$$

$$-cos(\frac{2i-1}{2}\delta)sin(\frac{\delta}{2})$$

$$\text{y}: \quad \frac{1}{2}(sin(270 - i\delta) + sin(90 - (i-1)\delta) =$$

$$sin(\frac{360 - (2i-1)\delta}{2})cos(\frac{180 - \delta}{2}) =$$

$$sin(180 - \frac{2i-1}{2}\delta)cos(90 - \frac{\delta}{2}) =$$

$$sin(\frac{2i-1}{2}\delta)sin(\frac{\delta}{2})$$

Normalizing gives:

$$\sqrt{(-cos(\frac{2i-1}{2}\delta)sin(\frac{\delta}{2}))^2 + (sin(\frac{2i-1}{2}\delta)sin(\frac{\delta}{2}))^2} = sin(\frac{\delta}{2})\sqrt{cos(\frac{2i-1}{2}\delta)^2 + sin(\frac{2i-1}{2}\delta)^2} = sin(\frac{\delta}{2})$$

Note this must be positive since it describes a vector length; since $0 < \delta < 180$, this length is $sin(\frac{\delta}{2})$ (and not $-sin(\frac{\delta}{2})$). The normalized vector is:

$$(-cos(\frac{2i-1}{2}\delta), sin(\frac{2i-1}{2}\delta)) = (cos(180 - \frac{2i-1}{2}\delta), sin(180 - \frac{2i-1}{2}\delta))$$

Since $\alpha = 0$, according to Eq. (11),this gives $\Delta(x_i)$.

Next, we calculate the response of x_i when i is odd for all $1 < i < N$. The users associated with x_i are:

$$u_{i-1} = 270 - (i-1)\delta$$
$$u_i = 90 - i\delta$$

We average the two vectors by averaging each coordinate x, y:

$$\text{x}: \quad \frac{1}{2}(cos(270 - (i-1)\delta) + cos(90 - i\delta) =$$

$$cos(\frac{360 + (1-2i)\delta}{2})cos(\frac{180 + \delta}{2}) =$$

$$cos(180 - \frac{2i-1}{2}\delta)cos(90 + \frac{\delta}{2}) =$$

$$cos(\frac{2i-1}{2}\delta)sin(\frac{\delta}{2})$$

$$\text{y}: \quad \frac{1}{2}(sin(270 - (i-1)\delta) + sin(90 - i\delta) =$$

$$sin(\frac{360 - (2i-1)\delta}{2})cos(\frac{180 + \delta}{2}) =$$

$$sin(180 - \frac{2i-1}{2}\delta)sin(-\frac{\delta}{2}) =$$

$$-sin(\frac{2i-1}{2}\delta)sin(\frac{\delta}{2})$$

Normalizing gives:

$$\sqrt{(cos(\frac{2i-1}{2}\delta)sin(\frac{\delta}{2}))^2 + (-sin(\frac{2i-1}{2}\delta)sin(\frac{\delta}{2}))^2} = sin(\frac{\delta}{2})\sqrt{cos(\frac{2i-1}{2}\delta)^2 + sin(\frac{2i-1}{2}\delta)^2} = sin(\frac{\delta}{2})$$

The normalized vector is:

$$(cos(\frac{2i-1}{2}\delta), -sin(\frac{2i-1}{2}\delta)) = (cos(-\frac{2i-1}{2}\delta), sin(-\frac{2i-1}{2}\delta))$$

Since $\alpha = 0$, according to Eq. (11),this gives $\Delta(x_i)$.

We now calculate the diversity for each user $i < N - 1$. User $i = 1$ has items x_1 and x_2 in his list. Thus, diversity is:

$$\frac{1 - x_1^T x_2}{2} = \frac{1}{2}(1 - (1 \cdot (-cos(\frac{4-1}{2}\delta) + 0 \cdot sin(\frac{4-1}{2}\delta)))) = \frac{1 + cos(\frac{3\delta}{2})}{2}$$

For even i, we first calculate the cosine similarity:

$$
\begin{aligned}
x_i^T x_{i+1} = & \\
& -cos(\frac{2i-1}{2}\delta)cos(-\frac{2(i+1)-1}{2}\delta) + sin(\frac{2i-1}{2}\delta)sin(-\frac{2(i+1)-1}{2}\delta) = \\
& -cos(\frac{2i-1}{2}\delta)cos(-\frac{2i+1}{2}\delta) + sin(\frac{2i-1}{2}\delta)sin(-\frac{2i+1}{2}\delta) = \\
& -[cos(\frac{2i-1}{2}\delta)cos(-\frac{2i+1}{2}\delta) - sin(\frac{2i-1}{2}\delta)sin(-\frac{2i+1}{2}\delta)] = \\
& -cos(\frac{2i-1}{2}\delta - \frac{2i+1}{2}\delta) = \\
& -cos(\delta)
\end{aligned}
$$

Diversity is given by:

$$\frac{1 - x_i^T x_{i-1}}{2} = \frac{1 + cos(\delta)}{2}$$

For odd i, cosine similarity is:

$$
\begin{aligned}
x_i^T x_{i+1} = & \\
& -cos(-\frac{2i-1}{2}\delta)cos(\frac{2(i+1)-1}{2}\delta) + sin(-\frac{2i-1}{2}\delta)sin(\frac{2(i+1)-1}{2}\delta) = \\
& -cos(-\frac{2i-1}{2}\delta)cos(\frac{2i+1}{2}\delta) + sin(-\frac{2i-1}{2}\delta)sin(\frac{2i+1}{2}\delta) = \\
& -[cos(-\frac{2i-1}{2}\delta)cos(\frac{2i+1}{2}\delta) - sin(-\frac{2i-1}{2}\delta)sin(\frac{2i+1}{2}\delta)] = \\
& -cos(-\frac{2i-1}{2}\delta + \frac{2i+1}{2}\delta) = \\
& -cos(\delta)
\end{aligned}
$$

Diversity is given by:

$$\frac{1 - x_i^T x_{i-1}}{2} = \frac{1 + cos(\delta)}{2}$$

We have remaining diversity for users $N - 1$ and N. Denote their corresponding diversities by d_{N-1} and d_N. The overall average diversity (Eq. (4)) is:

$$\text{div} = \frac{1}{N}\left(d_{N-1} + d_N + \frac{1 + cos(\frac{3\delta}{2})}{2} + \frac{(N-3)(1 + cos(\delta))}{2}\right)$$
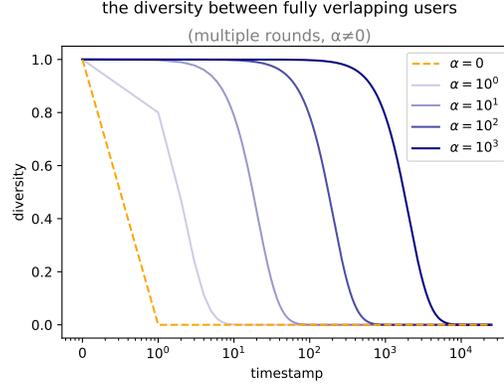
Figure 6. Empirical generalization of Proposition 1 for $\alpha \neq 0$.

Since d_{N-1} and d_N are non-negative, we get:

$$\text{div} \geq \frac{1}{N}\left(\frac{1 + cos(\frac{3\delta}{2})}{2} + \frac{(N-3)(1 + cos(\delta))}{2}\right)$$

Note $\frac{1 + cos(\frac{3\delta}{2})}{2}$ is also non-negative since $cos(\frac{3\delta}{2})$ cannot be less than -1. Hence, we get:

$$\text{div} \geq \frac{(N-3)(1 + cos(\delta))}{2N} = \frac{N-3}{N}\frac{1 + cos(\delta)}{2}$$

Finally, set $\delta = cos^{-1}(1 - 2\epsilon)$; this gives $\epsilon = \frac{1 - cos(\delta)}{2}$, and since $1 - \epsilon = \frac{1 + cos(\delta)}{2}$, we get as required:

$$\text{div} \geq \left(1 - \frac{3}{N}\right)(1 - \epsilon)$$

$\square$

C. Additional Experimental Results: Synthetic Data

C.1. Empirical analysis: Proposition 1 beyond $\alpha = 0$

Proposition 1 states that full user overlap leads to zero diversity, but relies on the assumption that $\alpha = 0$, i.e., that there are no modification costs, and so items can move arbitrarily on the unit sphere. Our motivation for considering $\alpha = 0$ (and a single time step) was that it is a useful proxy for larger α over multiple updates (that do not include retraining). Here we demonstrate empirically that this is indeed the case, i.e., that for $\alpha > 0$, diversity does go to zero over time.

We consider two items, positioned so that they are as far apart as possible: $x_1 = (1, 0)$ and the other at $x_2 = (-1, 0)$. We set $\tilde{v}$, which represents the spherical average of the overlapping users, to a random direction. We then measure how diversity changes over the time. As shown in Figure 6 diversity drops to zero for all α considered.

C.2. Accuracy and diversity over time

As an intermediate step between the experiments in Sec. 5.2 and Sec. 5.3, here we consider how accuracy and diversity trade off over time and for a range of λ, but while keeping $\alpha = 0$. Results are shown in Figure C.2. For $\lambda = 0$ (i.e., learning does not regularize for diversity), resulting NDCG is high and fixed, but diversity is nearly zero. This can be explained by the small value of σ_{u^*}, which implies that after strategic modification, items will be highly similar—which eliminates diversity (as in experiment 5.2), and makes the ranking task easy (since all items are similarly relevant). In the other extreme, when $\lambda = 1000$ (and so diversity is heavily regularized for), we observe that in the first round, diversity increases sharply, whereas NDCG decreases. In the subsequent rounds, diversity remains high and NDCG remains low.

For intermediary λ, results show how the learning allows to balance NDCG and diversity, where diversity is obtained for little loss in NDCG. For larger λ, more NDCG is sacrificed and diversity increases. An interesting phenomena is that lower
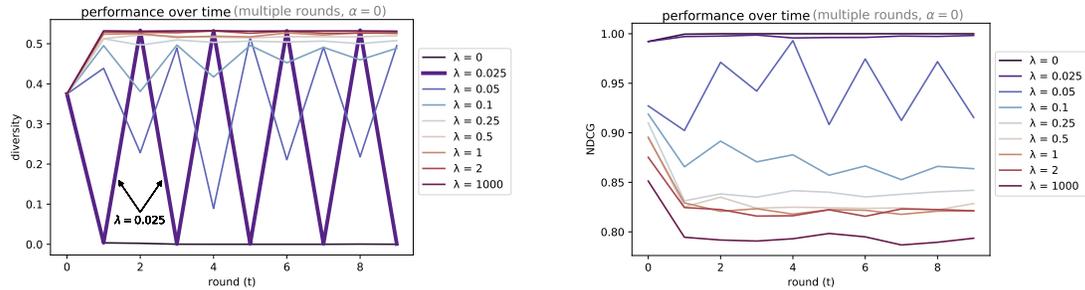
Figure 7. (**Left**) diversity vs number of rounds. (**Right**) NDCG vs number of rounds.
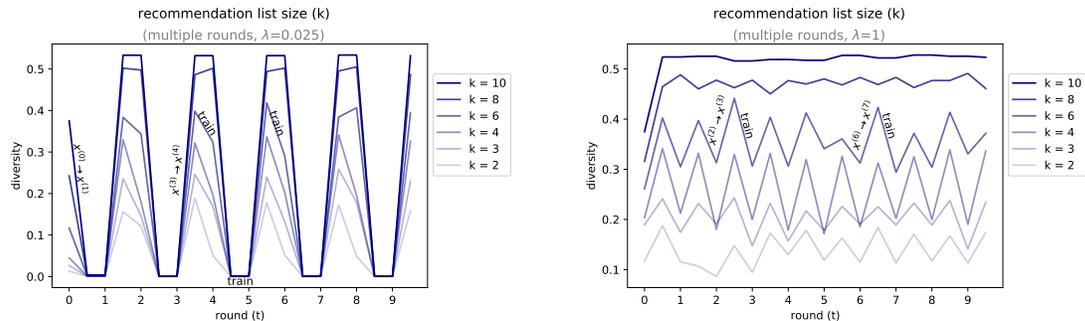
Figure 8. (**Left**) diversity vs number of rounds for $\lambda = 0.025$. (**Right**) diversity vs number of rounds for $\lambda = 1$.

values of λ exhibit periodic behavior. For example, for $\lambda = 0.025$, diversity alternates between very high (even round) and very low (odd rounds). Given how NDCG and diversity relate, we see two possible explanations for this:

(1) When diversity is high, the relevance of items may differ significantly. As a result, even a mild change in the item's ranking can cause NDCG to drop. To avoid this loss of NDCG, one solution is for the model to learn user embeddings u_i that are close to u^*. This, however, causes diversity to be low in the next round.

(2) When diversity is low, item features are very similar, and hence have highly similar relevance values. This implies that most rankings will have similar NDCG. Consequently, learning can encourage diversity without sacrificing current NDCG.

C.3. The role of the number of recommended items k

Here we consider how the size of the recommendation list k impacts the ability of the model to create diversity. We keep all other aspects of the experiment fixed, and vary k. Figure 8 shows results for two values of λ: a low (but positive) value of $\lambda = 0.025$ (left), and a high value of $\lambda = 1$. Overall, results show that higher k enables larger diversity; conversely, when k is small, results show that it is harder for the model to encourage diversity. One possible reason is that due to linearity, the top of the list is likely to includes items that are similar. The main distinction between the low and high λ is how diversity appears over time. As in the previous experiment in Sec. C.2 (in which $k = 10$), we see that $\lambda = 0.025$ exhibits alternating diversity for all k. For $\lambda = 1$, lower k still exhibits some fluctuations, but these become small as k grows.

D. Experimental details

D.1. Data

Our experiments use the Yelp dataset, which is publicly available at https://www.yelp.com/dataset/download.[8]

---

[8]Note Yelp periodically updates their repository; to ensure consistency, we include in our code preprocessed data, used in our experiments, that was parsed from raw data published by Yelp on July 2021.

Items.   Yelp includes data about many business types; of these, our experiment focuses on restaurants. To obtain restaurant entries, we manually identify and select all categories that pertain to restaurants (e.g, 'pizzaria' or 'burger bar'). This results in 22,197 distinct entries.

Features.   For features, we use a subset of the available features that were prevalent, and which were found to be informative for training f^*. We use category information to form additional features by grouping similar categories having similar contextual meaning; for example, the categories 'pizza', 'pasta', 'calzone', etc. are assigned the binary feature 'Italian cuisine'. Overall we use 43 features, which include:

'stars', 'alcohol', 'restaurants good for groups', 'restaurants reservations', 'restaurants attire', 'bike parking', 'restaurants price range', 'has tv', 'noise level', 'restaurants take out', 'caters', 'outdoor seating', 'good for meal-dessert', 'good for meal-late night', 'good for meal-lunch', 'good for meal-dinner', 'good for meal-brunch', 'good for meal-breakfast', 'dogs allowed', 'restaurants delivery', 'japanese', 'chinese', 'india', 'middle east', 'mexican food', 'sweets', 'coffee', 'italian', 'burgers', 'hot dogs', 'sandwiches', 'steak', 'pizza', 'seafood', 'fast food', 'vegan', 'ice cream', 'restaurants table service', 'business accepts credit cards', 'wheel chair accessible', 'drive thru', 'happy hour', 'corkage'.

Users.   As noted, we focus on active users who have contributed at least 100 reviews. One reason is that we have found that including low-activity users in this dataset results in a sparse and disconnected graph, with many isolated items (and corresponding users); this trivializes the task of diversification since most items can be incentivized independently. For each user, we consider the 40 most popular items (of those rated by that user), since by similar reasoning these provide larger overlap, and hence more intricate dependencies across items. In particular, we use the following procedure to construct potential item lists:

Denote by R the set of all restaurants, and by U set of active users.

- for all users $u \in U$, initialize $X_u = \emptyset$

- for all restaurants $r \in R$, initialize U_r to include all users u that have reviewed restaurant r

- while exists $u \in U$ for which $|X_u| < 40$:

    - let r be the restaurant with the most users in U, i.e., $U_r \cap U$ is largest
    - for each user $u \in U_r$:
        - add r to X_u
        - remove r from R
        - if $|X_u| = 40$, then for each r' in x_u, remove u from $U_{r'}$

D.2. Generating counterfactual ground-truth labels (pre-processing)

Since our experiments include modified items x_j^f that do not exist in the data, for learning and evaluation we require means to generate corresponding counterfactual relevance scores y_j^f. To achieve this, prior to the experiment we train a ground truth labeling function, $f^*(x)$, which we query for updated labels throughout the experiment. As noted, we ensure f^* is distinct from (and more powerful than) the predictive models f we learn in the actual experiment.

Data for training f^*.   We train f^* on all data generated by users having at least 50 restaurant reviews; these amount to 1,377 users and 113,852 reviews. Since the original data does not include informative user features, we generate ground-truth user features u_i^* by aggregating for each user i the features of all restaurants reviewed by i. This is similar in spirit to approaches for content-based recommendation. Formally, we define $u_i^* = \frac{1}{|\text{rev}_i|} \sum_{x \in \text{rev}_i} x$, where rev_i is the set of all restaurants that the user i reviewed.

For labels, we consider probabilistic labels $y_{ij} \in [0, 1]$ that describe the likelihood that user i will visit restaurant j, which we interpret as relevance. For each user i, we set $y_{ij} = 1$ for every restaurant j that i reviewed. To obtain negative labels, for each (i, j) pair, we first obtain the geographical location of restaurants j, and then retrieve the closest restaurant j' (in geographical terms) to j that i did *not* review; we then set $y_{ij'} = 0$. This is intended to mimic a setting in which i *could* have went to either j or j' (since they are physically nearby), but chose to go to j.

Architecture. We set f^* to be an MLP with ReLU activations. We use five layers, which we have found to be sufficient for expressing the non-linear relations between user and item features found in the data. The first layer has 86 inputs (43 restaurant features and 43 user features) and $86 \times 2 = 172$ outputs, and for each consecutive layer the output dimension reduces by half.

Training and evaluation. We split the data into train, validation, and test sets, using a 70-20-10 split. We optimize using Adam with a learning rate of 0.01, which gives a reasonable balance between performance and runtime, and used the validation set for early stopping. The final f^* achieves 72% accuracy on the held-out test set.

D.3. Hyper-parameters and tuning (main experiment)

For optimizing predictive models f in each experimental condition, we use Adam and train for a maximum of 200 epochs with learning rate 0.1. For smoothing (see Sec. 3.1), we use temperatures $\tau = 0.1$ for NDCG, $\tau = 1$ for the permutation matrix approximation, and $\tau = 5$ for the soft-k function; all were chosen to be the largest feasible values that permit smooth training. All experiments were run on a cluster of AMD EPYC 7713 machines (1.6 Ghz, 256M, 128 cores).

E. Additional experimental results: real data

E.1. Diversity over time – additional results

Figure 9 includes extended results pertaining to our main experiment in Sec. 6.1 for additional cost scales α.
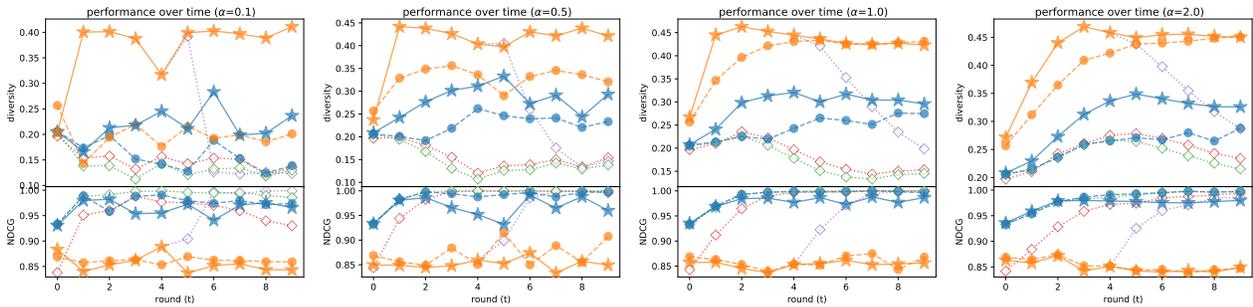
Figure 9. Diversity and NDCG over rounds for different methods and target NDCG values, and for costs $\alpha \in \{0.1, 0.5, 1, 2\}$.

E.2. Tradeoffs over time – additional results

Figure 10 includes extended results for our experiment on tradeoffs over time in Sec. 6.2 for additional cost scales α.
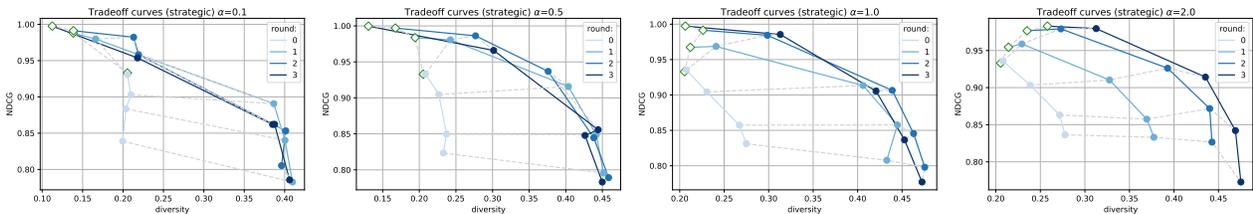
Figure 10. Pareto curves for `strategic` per round (varying λ; solid lines), and vs. `baseline`. Dashed lines show per-λ trajectories.

E.3. Sensitivity to a misspecification of the response model

Our experiments in Sec. 6 consider a setting in which the system has knowledge of the response model Δ, and in particular, of the true cost scale α. In this section we explore the sensitivity of our approach to learning under misspecified α. In particular, in each experimental instance, we train our model on some α, but test it on a different α'. Note that the misspecified α is used throughout all training rounds, and so the effects of misspecification accumulate.

20

Figure 11 (left) shows diversity over rounds on a fixed test $\alpha = 1$, for smaller training $\alpha \in [0.6, 1)$ (blue lines), larger training $\alpha \in (1, 1.4]$ (red lines), and the correct training $\alpha = 1$ (black line). Results show our approach is fairly robust to misspecification, with performance for all train α almost matching the correct one. Figure 11 (right) shows similar results for test $\alpha = 0.5$. Here robustness is preserved in full for smaller $\alpha \in [0.1, 0.5)$, but shows some deterioration in performance for the larger $\alpha \in (1, 1.4]$.
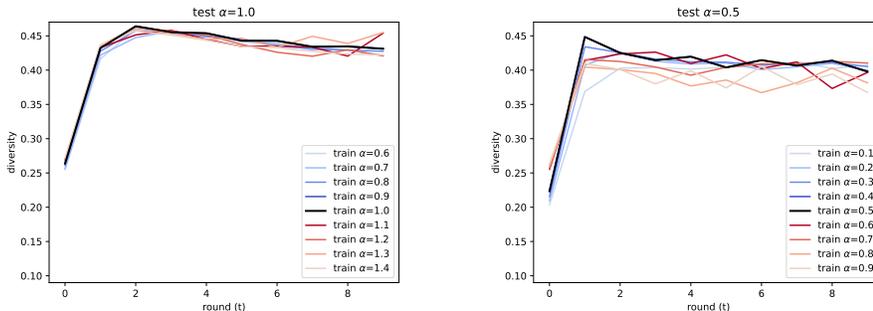
Figure 11. Diversity for misspecified cost scale α, for fixed test α and varying train α.

Complementarily, Figure 12 shows diversity for fixed train α and varying test α. Here, performance is again robust for $\alpha = 1$ (left). However, for the smaller train $\alpha = 0.5$ (right), in which items are subject to more dramatic modifications, mispecification has a significant effect on performance. For smaller test α (blue lines), sever overestimation of α in training (e.g., 0.5 vs. test $\alpha = 0.1$) has a severe negative effect on diversity over time. Interestingly, underestimation of α in training (red lines) results in improved diversity, suggesting that perhaps taking excessive cautionary steps is helpful in this case.
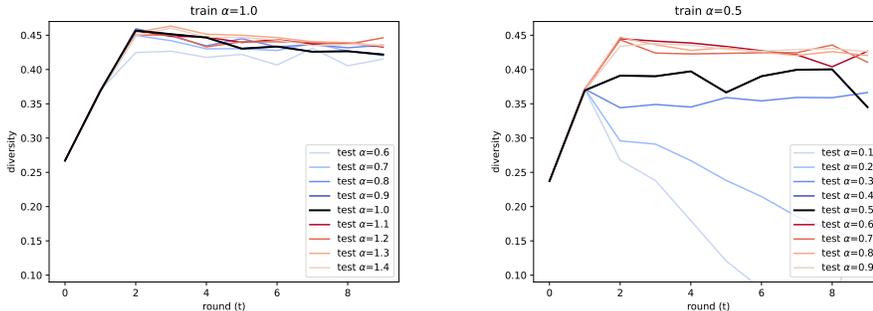
Figure 12. Diversity for misspecified cost scale α, for fixed train α and varying test α.

E.4. Entropy-based diversity regularization

As we state in Sec. 2, our paper focuses predominantly on cosine similarity, which we believe is appropriate for the recommendation environment we consider, and is a popular choice in the literature. Nonetheless, our approach is not restricted to this choice, and in this section we describe how it can be extended to operate on other similarity measures, and in particular, on entroty-based similarity. We then provide some empirical results for this settings.

To begin, note that conventional entropy regularization (e.g., Qin & Zhu (2013)) assumes a Gaussian distribution over feature vectors, and so is not immediately applicable to our setting of unit-norm features. To account for this, we propose a similar measure, but based on the Beta distribution, which is appropriate for inputs in [0,1], and which can apply per feature. The benefits of this measure are that: (i) its parameters can be efficiently estimated using moment matching (Owen, 2008); (ii) both parameter estimates and the differential entropy function are differentiable, and hence permit gradients to pass through; and (iii) entropy can be made to take strategically-modified inputs, and hence allow for strategically-aware optimization.

The Beta distribution is defined by two shape parameters, $a > 0$ and $b > 0$. Let $Z \sim \text{Beta}(a, b)$, and for a given sample of

such Z-s, denote its average by $\bar{Z}$ its standard deviation by S. Then the parameters a and b can be efficiently estimated as:

$$\hat{a} = \bar{Z}(\bar{Z}\frac{(1-\bar{Z})}{S^2} - 1), \qquad \hat{b} = (1 - \bar{Z})(\bar{Z}\frac{(1-\bar{Z})}{S^2} - 1)$$

Our approach is to consider each feature in each item list as deriving from some Beta distribution. Hence, for a given list X and feature i, we first estimate $\hat{a}, \hat{b}$ using $\{x_i\}_{x \in X}$. Note that both estimands are differentiable. Then, we compute entropy for this list and feature as $e_i(X) = \text{entropy}(\hat{a}, \hat{b})$, which admits a differentiable closed form (we used the pytorch implementation[9] that allows to pass gradients). Finally, we define $\text{div}(X) = \frac{1}{d}\sum_{i=1}^{d} e_i(X)$. We can then replace X with X^f, and plug into our objective in Eq. (10), which remains differentiable.

Using this approach, we extend our main experiments to also include entropy-based regularization. Figure 13 shows diversity and NDCG for all methods, when diversity is measured using entropy. Here again we see that strategically-aware methods outperform non-strategic methods across multiple cost scales α, although to a lesser extent than when measuring cosine similarity. Interestingly, optimizing for the incorrect correct measure (here, cosine; blue and orange) performs as well as when the correct measure (i.e., entropy; red and green) is optimized.
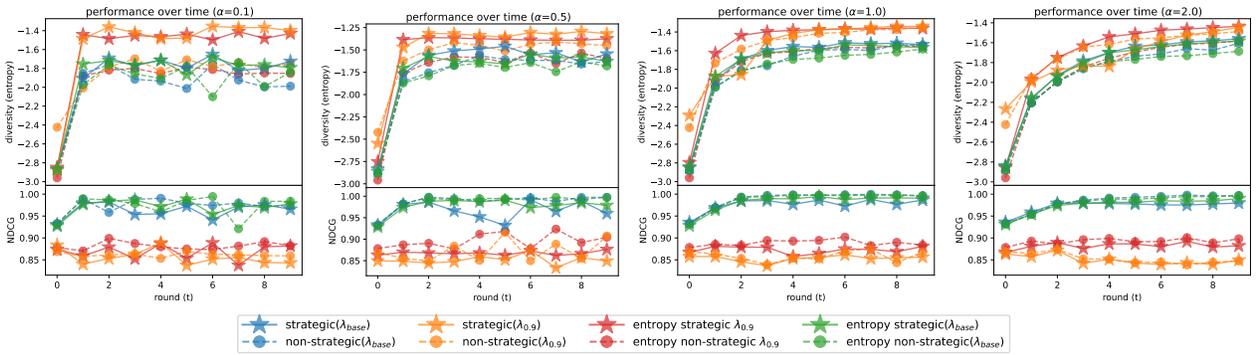
Figure 13. Experiments using entropy-based similarity as a diversity measure.

Finally, we rerun our original experiment using cosine similarity as a measure of diversity, but considering also methods that optimize entropy-based similarity (red and green). Here we see that misspecified diversity regularization is useful, but to a lesser extent than the correct form of regulariztaion. Nonetheless, the importance of awareness to strategic behavior remains to be more important (in terms of performance) than applying the correct regularizer.
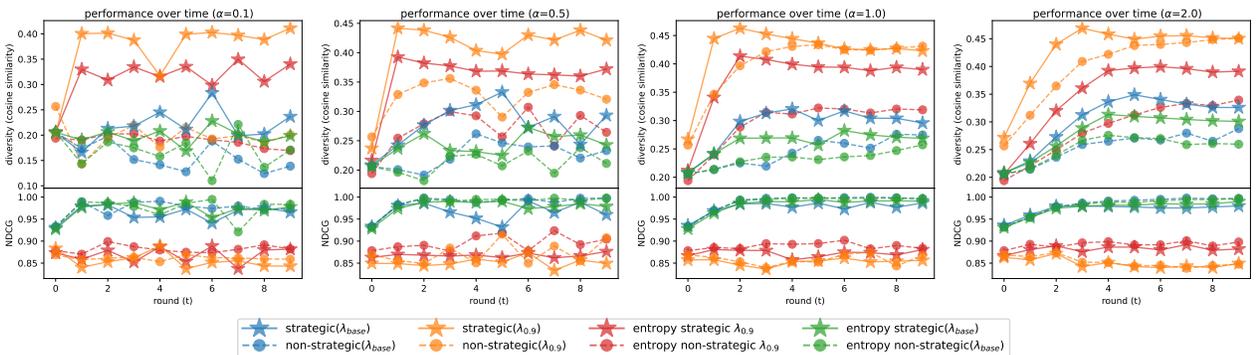
Figure 14. Experiments using cosine similarity as a diversity measure, but including also methods that optimize entropy-based diversity.

---

[9] https://pytorch.org/docs/stable/distributions.html