Competing AI: How does competition feedback affect machine learning?

Antonio A. Ginart **Eva Zhang** **Yongchan Kwon** **James Zou**
{tginart,evazhang,yckwon,jamesz}@stanford.edu
Stanford University, Palo Alto, CA

Abstract

This papers studies how competition affects machine learning (ML) predictors. As ML becomes more ubiquitous, it is often deployed by companies to compete over customers. For example, digital platforms like Yelp use ML to predict user preference and make recommendations. A service that is more often queried by users, perhaps because it more accurately anticipates user preferences, is also more likely to obtain additional user data (e.g. in the form of a Yelp review). Thus, competing predictors cause feedback loops whereby a predictor's performance impacts what training data it receives and biases its predictions over time. We introduce a flexible model of competing ML predictors that enables both rapid experimentation and theoretical tractability. We show with empirical and mathematical analysis that competition causes predictors to specialize for specific sub-populations at the cost of worse performance over the general population. We further analyze the impact of predictor specialization on the overall prediction quality experienced by users. We show that having too few or too many competing predictors in a market can hurt the overall prediction quality. Our theory is complemented by experiments on several real datasets using popular learning algorithms, such as neural networks and nearest neighbor methods.

1 Introduction

This paper studies what happens when machine learning (ML) predictors compete against each other. ML systems are deployed in ever more ubiquitous applications ranging from commerce to healthcare. It is becoming increasingly common for competing companies in similar markets to use ML to improve their services and attract customers or users. For example, platforms like Yelp[1] and Tripadvisor[2] both use ML to predict user preferences and make personalized recommendations for restaurants and other experiences. A user is more likely to use Yelp over Tripadvisor if they believe Yelp will give them a better recommendation than Tripadvisor (and vice-versa). Many users leave reviews, likes, or other forms of engagement on the platform that they end up using. Finally, the platform can use this feedback as new data to improve their predictive algorithms. The catch is that this form of user data is not an unbiased sample from the general population of users. Rather, it is biased by the fact that users that leave Yelp reviews are more likely to use Yelp more than, say, Tripadvisor.

Competing ML predictors can emerge in diverse settings. Competing search engines predict the most relevant web links given a user's search query. Competing lenders use their ML predictors to assess client credit and offer loan packages. In the ML-as-a-service industry, companies routinely compete to sell their ML algorithms to clients. While the details of the competition vary across settings, a key characteristic is that competition generates temporal dynamics and feedback loops for the learning algorithms. A predictor's performance at one time instance could impact the training data it (or its competitor) observes. Training sets are no longer independent samples from the general population distribution (this is the statistical definition of sampling bias). In turn, this affects the performance and bias of the predictor over time.

In this paper, we propose a model of competing predictors that captures the key features of these interactions and feedback loops. We investigate several common classes of predictors, including neural networks and nearest-neighbor models. Through experiments and theoretical analysis, we demonstrate that competition leads to specialization: while predictors perform better for specific sub-populations, they perform worse on the

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

[1] https://blog.yelp.com/2019/08/yelp-is-releasing-a-new-personalized-app-experience

[2] https://www.tripadvisor.com/engineering/personalized-recommendations-for-experiences-using-deep-learning/

general population distribution compared to when there is no competition. Moreover, we show that the quality-of-service experienced by users in this ecosystem of ML predictors is non-monotonic with respect to the number of competing predictors. The quality-of-service for users is diminished when there are too few or too many competing predictors. There is an optimal number of competing predictors that provides the best quality-of-service for users. This optimal number depends on several factors. One critical factor is how well the users can individually identify the predictor that's best suited for them.

Contributions As ML systems become ever more widely used, often by competing companies, it is increasingly important to model and characterize the effects of competition on ML. This topic is under-explored in ML. We summarize our main contributions as follows:

1. We introduce a novel model for competing predictors, which enables both large-scale experiments and theoretical analysis. Our model is generally useful for exploring statistical, algorithmic, and economic phenomena concerning the feedback dynamics between populations of competing predictors and users.

2. Through empirical and theoretical analysis, we show that user decisions create a feedback loop through which each ML predictor specializes toward a particular sub-population over time; often at the cost of worse performance over the general population of users.

3. We analyze the effect of competition on the quality-of-service for the users. We show that the overall quality can be non-monotonic in the number of competing ML predictors.

2 Model for competing predictors

We assume there is some supervised ML task that requires algorithms to make predictions for users. The prediction task corresponds to a general population distribution $\mathcal{D}$. For $(x,y)\sim\mathcal{D}$ we can think of $x\in\mathcal{X}$ as representing the relevant user attributes or features and $y\in\mathcal{Y}$ as the predictive target. We have k competing predictors, $\{A^{(1)},...,A^{(k)}\}$. Predictor i has an initial batch of training data $D_0^{(i)}$ that are independently and identically distributed (i.i.d.) samples from $\mathcal{D}$. The initial training data $D_0^{(i)}$ corresponds to the data that each predictor starts with—e.g. data from an initial pilot. We typically think of $|D_0^{(i)}|$ as small. We refer to this initial data as *seed data*. Let $D_t^{(i)}$ denote the dataset that the i-th predictor has up to and including time t. $A_t^{(i)}$ is the predictor that is trained on $D_{t-1}^{(i)}$. At each time t, a new sample $(x_t,y_t)\sim\mathcal{D}$ is drawn, representing the t-th user in some user stream. Each predictor outputs $\hat{y}_t^{(i)}=A_t^{(i)}(x_t)$. Then, the user selects one of the k predictors as a *winner*, denoted by w_t. The winning predictor w_t gets the datum (x_t,y_t): $D_t^{(w_t)}=D_{t-1}^{(w_t)}\cup\{(x_t,y_t)\}$ and $D_t^{(i)}=D_{t-1}^{(i)}$ for $i\neq w_t$. We can think of predictors as agents seeking to maximize their query rate and users as agents seeking to maximize the accuracy of the predictor they select. We model predictors with both common parametric and non-parametric ML algorithms. For simplicity, in our experiments and theory we will consider competitions in which predictors are *symmetric*, meaning they use the same learning algorithm. We proceed to describe our user model.

A flexible model for user choice We would like to model how a user chooses among the set of competing predictors. For starters, assume that $\mathcal{Y}$ in the prediction task is categorical and the *prediction quality*, $q(y_t,\hat{y}_t)=\mathbf{1}\{y_t=\hat{y}_t\}$, is binary. We consider the case when users do not have prior biases towards any predictor. Instead, we stipulate that the probability that a user selects predictor should only depend on the tuple $\mathbf{q}_t=(q(y_t,\hat{y}_t^{(1)}),...,q(y_t,\hat{y}_t^{(k)}))$, meaning that user selection probability is only a function of the prediction quality. We denote the user selection operation **SELECT**. The **SELECT** encodes the conditional distribution for w_t over $[k]$ given $\mathbf{q}_t$ where $[k]:=\{1,...,k\}$. We can think of **SELECT** as a randomized operation that outputs the winner, *i.e.*, $w_t=\mathbf{SELECT}(\mathbf{q}_t)$. Equivalently, w_t is a random variable parameterized by $\mathbf{q}_t$. Given that w_t only depends on $\mathbf{q}_t$, the sole parameter that uniquely characterizes a user's choices is the difference in probability that the user selects a correct predictor over an incorrect predictor. We refer to this as the *correctness advantage*, $\mathbf{P}_{\text{ADV}}$, in the system. For any $\mathbf{q}_t$ such that for some $i\neq j$, $y_t=\hat{y}_t^{(i)}$ and $y_t\neq\hat{y}_t^{(j)}$, we define *correctness advantage*[3] as

$$\mathbf{P}_{\text{ADV}}:=\mathbf{Pr}(w_t=i|\mathbf{q}_t)/\mathbf{Pr}(w_t=j\,|\,\mathbf{q}_t).$$

Without loss of generality, we can equivalently use the widely-used softmax parameterization for $\mathbf{P}_{\text{ADV}}$:

$$\mathbf{Pr}(w_t=i\,|\,\mathbf{q}_t)=\frac{1}{Z}e^{\left(\alpha q(y_t,\hat{y}_t^{(i)})\right)}$$

where $Z=\sum_{j\in[k]}\exp\left(\alpha q(y_t,\hat{y}_t^{(j)})\right)$ and thus $\mathbf{P}_{\text{ADV}}=\exp(\alpha)$.

[3] If either $\hat{y}_t^{(i)}=y_t$ or $\hat{y}_t^{(i)}\neq y_t$ for all $i\in[k]$, then a user chooses a competitor uniformly at random.

For simplicity, we use temperature parameter α in lieu of $\mathbf{P}_{\mathrm{ADV}}$ throughout this work; this parametrization does not limit user behavior. To be clear, the user does not necessarily know the true y_t (otherwise there may not be a need for the predictors). Moreover it is not necessary that the user observes all of the predictions $\hat{y}_t^{(i)}$ when making a selection. It is sufficient that the user has some side information on which predictors are likely to be correct. The degree of this correlation can be captured by α. This model is simple and flexible, and it captures the essence of the interaction between predictors and users. We can view the temperature parameter α as indicating how informed the user selections are. When $\alpha = 0$, the user has zero information and uniformly at random selects a predictor. As α increases, the user is more likely to select the algorithm that makes the correct prediction. Therefore, α is a natural metric of *information efficiency* . In many settings, users might be more likely to select a predictor that makes a correct prediction than an incorrect predictor (*i.e.* $\alpha \geq 0$). This might be because users have some private signals or experiences, and also because users typically want to pick the highest quality prediction. Because this is more realistic, we primarily focus on $\alpha \geq 0$ for our experiments and analysis.

For simplicity, we will largely deal with temperature α for the remainder of the paper, while remembering the direct connection between α and the correctness advantage. Another advantage of the softmax parameterization is that it easily generalizes to regression settings by replacing q with any generic loss function ℓ (such as MSE). In the main text of this work, we will assume α is a system constant and thus is fixed for all users. In Appendix B, we further generalize and let α depend on the particular user that is making the selection by sampling each user's α parameter from a standard normal distribution. This reflects that individual users have varying amounts of prior information about the predictors. We found that this yields in highly similar results (refer to Appendix B).

One simplification that we make in our model is that only the selected predictor receives (x_t, y_t). There are several possible modeling variation on this: for example, one could allow the non-selected predictors to add x_t (not y_t) to its database and this could be used for semi-supervised learning. One could also allow the user selection to depend not just on the current predictions but also on predictor reputation. Additionally, one could assume that only some fraction of users actually leave feedback, which would mean that the winner observes y_t only some fraction of the time. These are interesting directions for follow up exploration. In this paper, we make the simplifications in order to capture the key essence due to competition in purely supervised learning.

3 Experiments

We present simulations of competing learners in the supervised (Sec. 3.1) and collaborative filtering settings (Sec. 3.2). We investigate the effects of competition on the predictors and the users and empirically characterize predictor specialization and non-monotonicity of the quality-of-prediction.

3.1 Supervised Learning

We use several popular benchmark datasets for $\mathcal{D}$: `Postures` (Gardner et al., 2014; Dua and Graff, 2017), `Adult Income` (Dua and Graff, 2017), and `FashionMNIST` (Xiao et al., 2017). For `Postures` and `Adult Income` in particular, each datum corresponds to data from one individual, which is particularly appropriate for our motivating competition setting. We explore the effects of different information efficiency value α. For each dataset, we fix a small number of i.i.d. seed samples (order 10^0 - 10^2) and run the simulation for a large number of rounds (order 10^3 to 10^4). We perform our experiments with the widely-used multi-layer perceptron (MLP) as an example of parametric predictors and nearest-neighbors (NN) as an example of non-parametric predictors. In Appendix B we also report similar simulations conducted with a logistic regression model as well as full details of all the experiments. While there are many other classes of predictors to explore, we believe that the standard models used here cleanly capture the key insights.

Competition drives predictor specialization We performed experiments with four competing predictors (similar results are seen for other number of predictors). In Fig. 1 we present heatmaps indicating the accuracy of the four competing predictors on each of the label classes. Red (blue) indicates that a predictor is better (worse) than the average predictor on that class.

When $\alpha = 0$, the user uniformly at random selects a predictor and the lack of competition results in all of the predictors being close to average accuracy. As α increases, we see a clear trend towards greater variations in class-conditional accuracy among the predictors, indicating specialization. A stark example of this can be observed for the competing MLPs on the Postures dataset. For large α, the four predictors specialize over the five classes such that each predictor strongly favors only one particular class (except for predictor 3 which favors two classes). Predictor 0 specializes in detecting `stop`, predictor 2 specializes in `fist`, predictor 3 specializes in `fingers` and predictor 1 is split between `point` and `grab`. Outside of each predictor's specialty class, the performance is low across the board. The predictor specialization not only occurs over the classes but also within the features. An interpretable example of this is for the binary gender

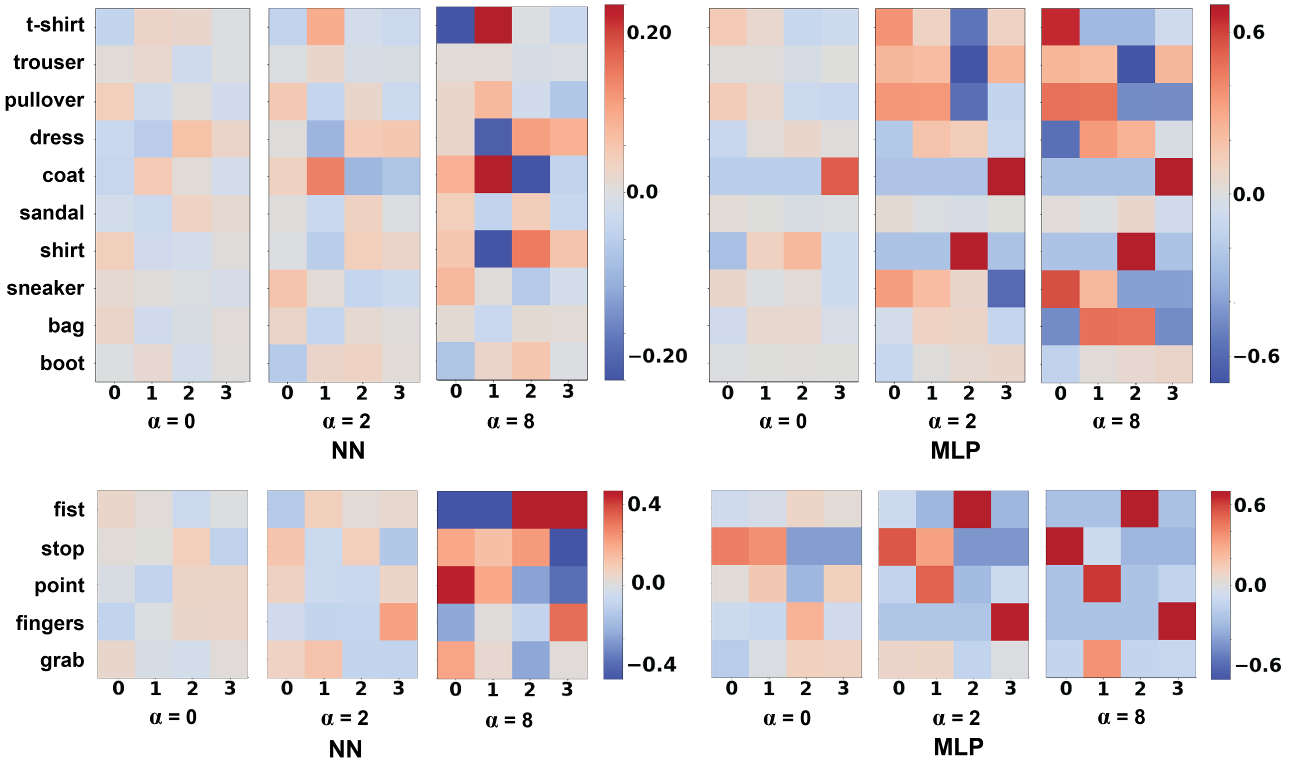

Figure 1: Predictor specialization heatmaps for FashionMNIST (top row) and Postures (bottom row) with NN (left column) and MLP (right column). For each dataset and algorithm we include heatmaps of α at low (0), medium (2), and high (8) values (left to right). Each heatmap is a #(classes) × #(predictors) grid. The ij-th block in a grid indicates the difference between the average class-conditional accuracy for the i-th class and the j-th predictor's class-conditional accuracy for the i-th class. Predictors are indexed by an arbitrary id number and classes are labeled on the left. Red (blue) indicates an accuracy that is higher (lower) than average, and white is average.

feature in the Adult Income dataset (Fig. 2). At $\alpha = 0$ all predictors are close to average accuracy. As α increases, we see that predictor 4 and eventually predictor 7 specialize in males versus females, respectively. This illustrates how competition could lead to ML algorithms that specialized to specific demographic groups.

Larger α creates a positive feedback loop that leads to specialization. Random variation in the initial training batches generates some heterogeneity in the predictors. Users are likely to select the predictor that is best suited for them with large α. This leads that predictor to improve its model specifically for that sub-population.

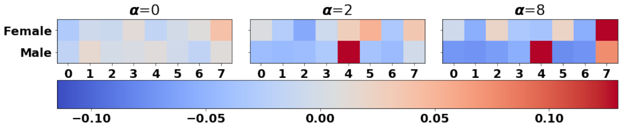

Figure 2: Predictor specialization heatmap for Adult with MLP with 8 competitors. Rows in the grid indicate male vs. female individuals. Red (blue) indicates an accuracy that is higher (lower) than average, and white is average.

In turn, this results in an increased likelihood that members of that sub-population select said predictor. While a common business strategy is for firms to intentionally specialize to particular sub-populations from the onset (Balassa, 1989; Yang and Ng, 2015), the interesting aspect of the phenomena here is that specialization emerges naturally (and unintentionally) due to the competition over data.

We next quantify how the competition affects the predictor's performance on the general population distribution, which is measured as its average accuracy over $\mathcal{D}$. Note that this $\mathcal{D}$ is different from the distribution of data points from a user at any particular time — as we shall see next, the predictor does better on the latter distribution. Fig. 3 measures the change in accuracy over $\mathcal{D}$ compared to the $\alpha = 0$ baseline, which uses the same number of training samples but removes competition.

There is a consistent trend that increasing the information efficiency α at any number of predictors results in lower accuracy on $\mathcal{D}$. The drop in accuracy is largest when there is an intermediate number of predictors. This is because the average number of samples each predictor

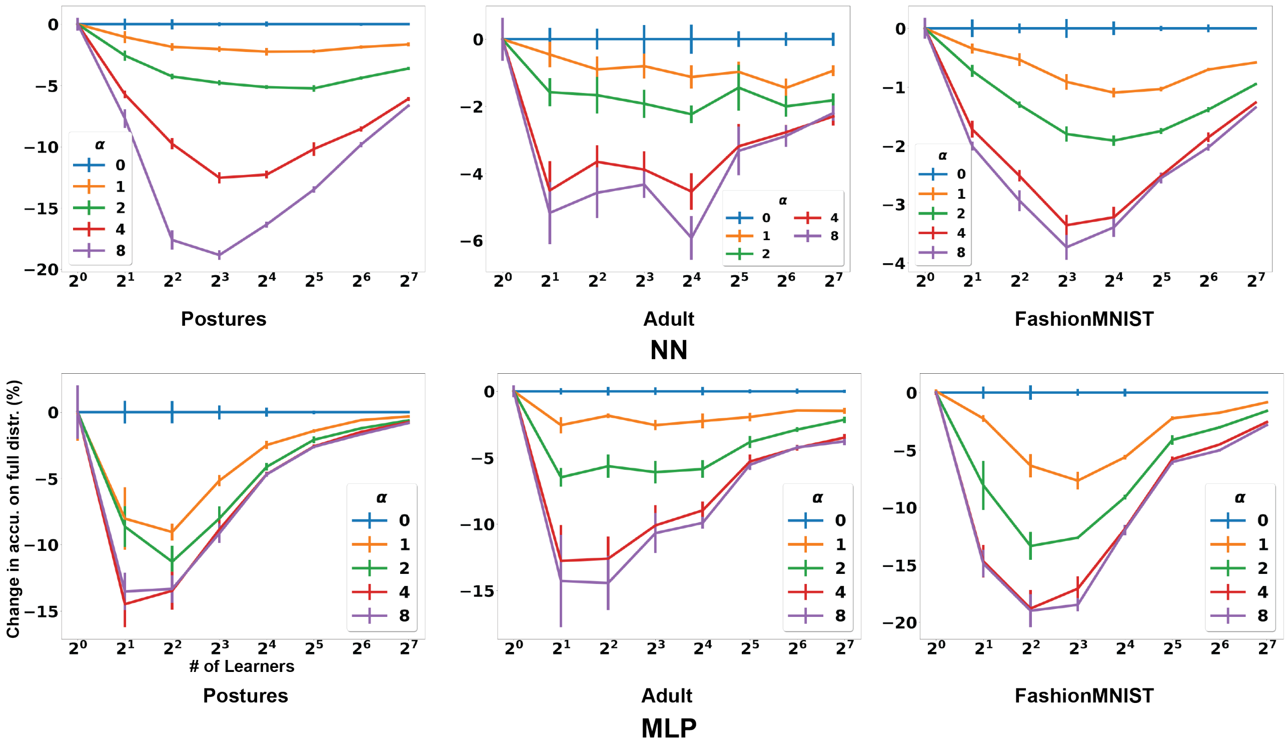

Figure 3: How specialization affects predictor performance: number of predictors (x-axes, log-scale) vs. change in accuracy over general population $\mathcal{D}$ (y-axes, in percentage) for NN and MLP on 3 datasets. To measure the effect of competition, change in accuracy is with respect to a baseline simulation in which winning predictors get an i.i.d. sample instead of the one that selected it to remove selection bias. Confidence intervals are standard error of the mean for 5 replicates.

receives decreases when there are more predictors, since the total number of rounds, or equivalently the total number of samples, is fixed. With fewer data points, there's less feedback to bias the predictor. The decrease in accuracy for the overall distribution could be costly when the company tries to broaden its user-base to the entire $\mathcal{D}$. This is an important consequence of specialization.

Prediction quality for users We shift our focus to analyze the prediction quality experienced by the users. We define the *prediction quality for users* as the average accuracy of the selected predictor averaged over all the rounds of competition: $\frac{1}{T}\sum_{t=1}^{T}\mathbf{1}(\hat{y}_t^{(w_t)} = y_t)$. Fig. 4 shows how this quality varies as the number of predictors (x-axes) and α (different colors) change for NN and MLP applied to three datasets. In each panel, the total number of datapoints (i.e. users) is fixed. Prediction quality for users is consistently higher when users have more information (larger α) when picking the predictor.

Interestingly, we find that the prediction quality for users can be non-monotonic. For example, in Postures data with competing NNs, the highest quality is achieved with 16 competing predictors; having too few or too many predictors decreases quality. The intuition for this phenomenon is as follows. When there is just one predictor, a user has no choice and changing α has no effect. With more predictors and relatively high information efficiency, each user can select the predictor that is likely to be accurate for it, and hence the prediction quality improves. However, when there are too many predictors, each predictor gets fewer training data (recall that the total number of data points is fixed). Hence none of the predictors is very accurate and the overall quality starts to decline. In Sec. 4, we show this phenomena is a mathematical consequence of the learning competition under some mild conditions. The prediction quality over a full range of information efficiencies depicting the monotone increasing and decreasing regimes (for near-infinite and near-zero α) can be found in Appendix B.

3.2 Extension to Collaborative Filtering

Previous experiments capture the setting where each user is a single data point that appears once. Here we experimentally investigate a collaborative filtering extension where each user contributes multiple data points. Collaborative filtering competitions follow the same structure as described in Sec. 2, with the primary differences being a new **SELECT** operation and allowing repeated samples from each user. As before, we have a set of k competing recommenders. We also have a set of m distinct users, $\{u^{(1)},\ldots,u^{(m)}\}$ that are seeking recommendations over a set of r items

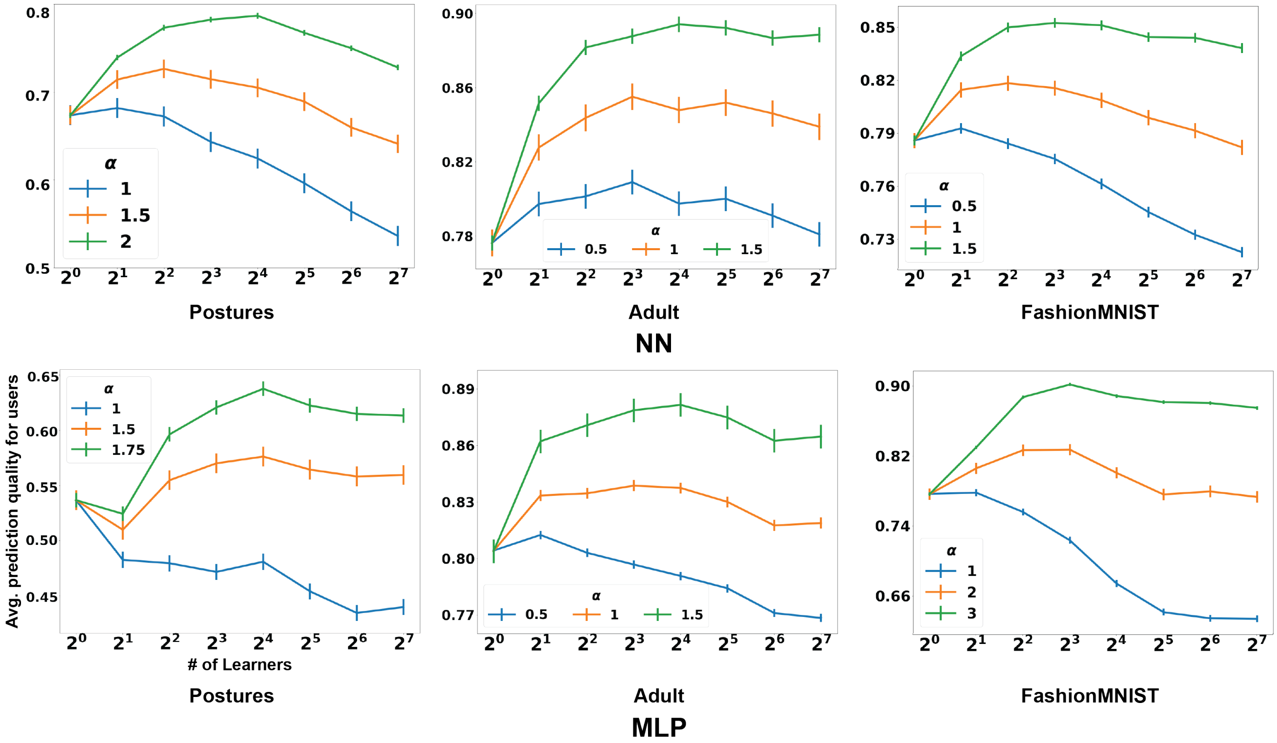

Figure 4: Prediction quality for users: number of predictors (log-scale) vs. avg. prediction quality for users with NN and MLP on 3 datasets. Prediction quality is averaged over all of the rounds in the simulation. Confidence intervals are standard error of the mean for 5 replicates.

(for simplicity, we assume that these items are shared across the recommenders). At each round, a uniformly at random user $u_t \in \{u^{(1)},\dots,u^{(m)}\}$ selects one of k recommenders: $w_t = \mathbf{SELECT}(u_t)$. Then recommender w_t recommends an item for u_t: $A_t^{(w_t)}(u_t) \in [r]$. There is a latent preference matrix $M \in [0,1]^{r \times m}$, where M_{ij} is the probability that user $u^{(j)}$ interacts with the i-th item (pCTR). The "winning" recommender, w_t observes the interaction between a user and item as feedback. Precisely, recommender w_t observes $(x_t, \tilde{y}_t)$ where $x_t := (i,j)$ is simply a pair of the item i and the user j, and $\tilde{y}_t \sim \mathbf{Bernoulli}(M_{ij})$ describes if there is an interaction when item i is recommended to user j. As before: $D_t^{(w_t)} = D_{t-1}^{(w_t)} \cup \{(x_t,\tilde{y}_t)\}$ and $D_t^{(i)} = D_{t-1}^{(i)}$ for $i \neq w_t$.

Users want to maximize the preference scores of the items that they get recommended to them, and recommenders want to maximize the number of queries for items they receive from users. In our experiments, each user keeps track of the quality of past recommendations from each recommender and individually solves a multi-arm bandit (Slivkins, 2019) problem with recommenders as arms when it is their turn to **SELECT**. Each recommender similarly solves an an online matrix factorization problem (Schafer et al., 2007) based on the observed user-item interactions using alternating least-squares (Hastie et al., 2015). We generate M as the product of low-rank factors with i.i.d. Gaussian entries. We run the simulations for 2×10^5 rounds. Appendix B contains the formal description of the model and details about the protocol and implementation.

Fig. 5 shows the collaborative filtering results. Fig. 5 (left) is analogous to Fig. 3; the y-axes quantifies how well each recommender performs over the general population distribution of users. This performance is measured as the expected probability that a randomly selected user decides to interact with the item suggested by this recommender. As in the setting of competing predictors, competition and specialization leads to a decrease in the performance of recommenders for the general user distribution. Fig. 5 (right) is analogous to Fig. 4; the y-axes there is the prediction quality experienced by the users. We find a similar phenomenon as before: having too few or too many recommenders can decrease the quality experienced by users. These collaborative filtering experiments demonstrate that the phenomena that competition leads to algorithmic specialization and that there is a sweet spot for the number of ML models can hold in diverse settings.

4 Theoretical analysis

We carry out theoretical analysis to further understand and support our empirical findings. Here, we assume a binary classification task for simplicity. Complete

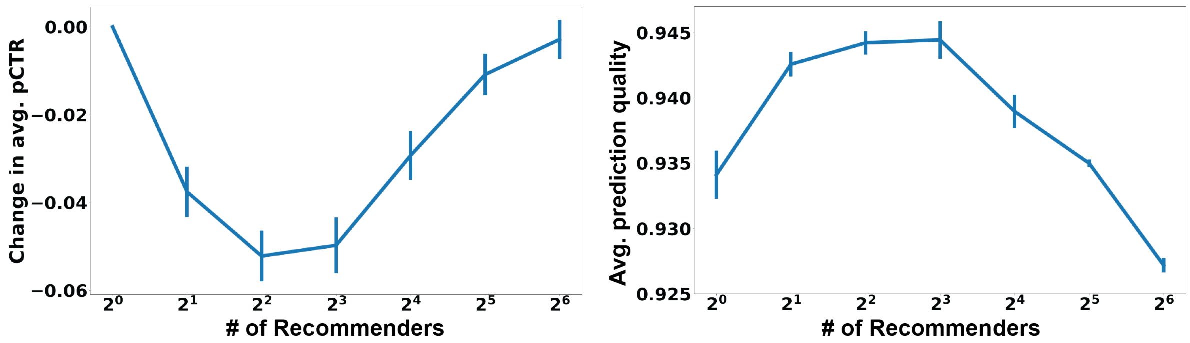

Figure 5: Collaborative filtering competition: Recommender pCTR over general population (left) and avg. prediction quality for users (right) for varying number of recommenders (log-scale). Change in pCTR (left) is with respect to an otherwise identical baseline simulation in which winning recommenders always get an i.i.d. user sample instead of the user that selected it. Prediction quality (right) is averaged out over all the round in the simulation. Confidence intervals are std. error of the mean from 5 replicates.

proofs for all claims are in Appendix C. The analysis in this section can be interpreted as formalizing sufficient conditions for the empirically observed effects of competition to emerge.

4.1 Cost of competition for predictors

Our experiments show that competition causes each predictor to specialize on a sub-population and perform worse on the overall population distribution. We show for simple parametric and non-parametric models that competition results in a gap in the error rates attained by the trained predictors. Let $\mathcal{R}(A;\mathcal{D}) = \mathbf{E}[\mathbf{1}\{A(X) \neq Y\}]$, where $(X,Y) \sim \mathcal{D}$, denote the error rate of a predictor A on samples from the general population. The average error rate of the competing predictors (on $\mathcal{D}$) after t rounds of competition is $\mathcal{R}_t^k = \sum_{i\in[k]} \mathcal{R}(A_t^{(i)};\mathcal{D})/k$, where $A_t^{(i)}$ is predictor i after t rounds of competition as described in Sec. 2 and k is the total number of competitors. The following asymptotic result concerns itself with the perfect information limit $\alpha = \infty$) and holds quite generally for most non-parametric models. Plainly speaking, the theorem says that for certain distributions, the average error rate of competing predictors is not within a constant factor of the error rate of a single predictor.

Theorem 4.1. *Suppose users have perfect information ($\alpha = \infty$) and each predictor is trained using a non-parametric method that is asymptotically a C-approximation (in the usual sense, see Ausiello et al. (2012)) to the Bayes error rate. Then, for any seed set size $s = |D_0|$, there exists $\mathcal{D}$ such that for any $k > 1$, and* , $\lim_{t\to\infty} \frac{\mathcal{R}_t^k}{\mathcal{R}_t^1} = \infty$.

The intuition for Thm. 4.1 is as follows. In the case that Y is deterministic given X, the ML problem is effectively an interpolation. In this case, the Bayes error rate is 0 and this error rate is asymptotically achieved by most non-parametric methods (Tsybakov, 2008) given that they are C-approximations to the Bayes rate. However, when $\alpha = \infty$ in a competition, an unlucky seed set could result in a predictor never achieving 0 error rate, which breaks the C-approximation. Furthermore, this probability can be bounded away from 0 for any finite seed set. Next we show that a risk gap still exists for finite α.

Theorem 4.2. *Suppose $k = 2$ and both predictors use the nearest-neighbor algorithm. Let $s = |D_0|$ be the number of i.i.d. seed samples that each predictor starts with and assume $s \geq 2$. If $\alpha > \log(2)$, then there exists $\mathcal{D}$ such that*

$$\lim_{t\to\infty} \frac{\mathcal{R}_t^2}{\mathcal{R}_t^1} \geq 1 + \frac{1}{54\sqrt{2s}} \left(\frac{8}{9\sqrt{s}}\right)^{s/2} \left(1 - \frac{2}{2+e^{\alpha}}\right)^2$$

The risk ratio decreases quickly in s, indicating that sufficient seed data may be an effective counter-measure for the non-parametric case. Also, the risk ratio grows larger for larger α, which also coincides with intuition.

We next investigate the parametric setting by analyzing an ordinary linear least squares regression. For this analysis, we use the mean squared error to measure expected risk $\mathcal{R}_t^k$. We present two lower bounds. The first holds for any positive information efficiency $\alpha > 0$ and depends on the number of seed samples. The second holds for any finite number of seed samples, but requires the users to have perfect information ($\alpha = \infty$).

Theorem 4.3. *Suppose the data is generated from a linear model $Y = XW + \epsilon$ with $\mathbf{E}(\epsilon|X) = 0$. Assume each predictor uses an ordinary least-squares linear estimator. Let $s \geq 1$ be the number of i.i.d. seed samples each predictor starts with and assume $k \geq 2$. We have the following:*

(i) If $\alpha > 0$ then $\lim_{t\to\infty} \sup_{\mathcal{D}} \frac{\mathcal{R}_t^k}{\mathcal{R}_t^1} \geq 1 + \frac{1}{7056 s^{3/2}}$

(ii) If $\alpha = \infty$ then $\lim_{t\to\infty} \sup_{\mathcal{D}} \frac{\mathcal{R}_t^k}{\mathcal{R}_t^1} \geq \frac{2k}{k+1}$

Thm. 4.3 tells us that when α is large, there is a significant (close to 2×) penalty incurred when there are many competing predictors. When α is small, the penalty is also smaller but does not vanish if the number of seed samples is not too large. Notice that for regression, the worst-case ratio of expected risks vanishes at a low-degree polynomial rate in s. This decays far slower than the exponentially vanishing bound for non-parametric methods. This suggests that seed data may be less helpful in mitigating the cost of competition with parametric methods than with non-parametric.

4.2 Prediction quality for users with competing predictors

We analyze how the number of competing predictors affects the overall prediction quality experienced by users. We want to characterize the dependence of quality on the number of predictors, k, and the information efficiency, α. Recall our notion of empirically measurable *prediction quality for users*: $\frac{1}{T}\sum_{t=1}^{T}\mathbf{1}(\hat{y}_t^{(w_t)}=y_t)$. Here we will be studying theoretically relevant quantities to this random empirical value. We define the *expected prediction quality at time* τ, denoted by $\mathbb{A}_\tau$ as $\mathbb{A}_\tau = \mathbf{E}(\mathbf{1}\{\hat{y}_\tau^{(w_\tau)}=y_\tau\})$. To this end, we will phrase our results in terms of the accuracy, $\mathcal{A}_t^k$, defined by $\mathcal{A}_t^k = 1-\mathcal{R}_t^k$ rather than the risk $\mathcal{R}_t^k$.

Assumptions To make the analysis tractable, we make several natural modeling assumptions that we outline here. We define the following: $\delta = \mathcal{A}_t^1 - \mathcal{A}_t^2$ and $\varepsilon = \mathcal{A}_0^k - \frac{1}{2}$.

1. We are primarily interested in regimes when seed sets are small, which implies that the initial predictors are weak models. Concretely, we assume: $\varepsilon < 1/14$

2. Also, we should have enough data to experience diminishing marginal returns from additional samples. This means that the individual accuracy for one predictor is not much better than the individual accuracy for two predictors each with approximately half as many samples. Concretely, we assume: $0 < \delta < \frac{1}{6}$

3. While we allow the predictors to be correlated, they cannot be extremely correlated. To see why this is necessary, consider the case in which the predictors are perfectly correlated. They always give the same prediction and thus the users derive no benefit from the competition.

4. Finally, we assume that the expected accuracy for a predictor monotonically increases in the data set size. Thus, having more data is better, *on average*, but not necessarily always.

Our result shows that in the regimes described above, there necessarily exists an interval of intermediate information efficiencies, $0 < c_1 < c_2 < \infty$ such that for $\alpha \in (c_1, c_2)$, the optimal number of predictors is neither 1 or ∞. This means that that there is a finite "sweet spot" in the number of competing predictors that produce the best user quality.

Theorem 4.4. *Assume a learning competition at round t under the conditions stated above. Let ρ be the pairwise covariance between two predictors. If we have $\rho < \mathcal{A}_t^k - (\mathcal{A}_t^k)^2 - 6\delta$ then there exists $0 < c_1 < c_2 < \infty$ such that if $c_1 < \alpha < c_2$ then the expected prediction quality for users at round t is maximized by some k^* number of predictors such that $1 < k^* < \infty$. In particular, $c_1 < \log \frac{\mathcal{A}_t^1 - (\mathcal{A}_t^1 - \delta)^2 - \rho}{\mathcal{A}_t^1 - (\mathcal{A}_t^1 - \delta)^2 - \rho - 2\delta}$ and $c_2 > \log \frac{(1-4\varepsilon)\mathcal{A}_t^1}{1-\mathcal{A}_t^1}$.*

To make the result concrete, we instantiate $\mathcal{A}_t^1 \leftarrow 0.9$, $\delta \leftarrow 0.05$, $\epsilon \leftarrow 0.05$ and $\rho \leftarrow 0$. Thm. 4.4 tells us that prediction quality for users at time t is non-monotonic if $0.65 < \alpha < 1.97$. This range of α agrees reasonably well with our empirical measurements. The intuition for the theorem is as follows. Obviously, when α is large having many weak predictors is better for users as the users themselves can take the burden of selecting a correct predictor. When α is not too large, having many weak predictors is not necessarily better for users than having a few smart ones (consider the extreme case of $\alpha \to 0$). However, if α is exactly zero, then having a single predictor is generally better than having even two predictors since the user is not more likely to **SELECT** the correct predictor and the two predictors have split the data. But, there is a sweet spot in α for which the user benefit from being slightly more likely to select the correct predictor outweighs the benefit that a single predictor has in terms of volume of training data. This is due to the near-universal phenomena in ML of diminishing marginal returns in number of training samples.

5 Discussion

Related works In Mixture-of-experts and related ensemble learning methods, multiple predictors work together to train for a prediction task (Masoudnia and Ebrahimpour, 2014; Dietterich, 2000; Zhou, 2012; Opitz and Maclin, 1999). There, the algorithms work together in the ensemble to optimize a common objective, and data can be shared between the algorithms. This differs from our setting where the predictors directly compete over user queries and training data.

Recent literature in multi-agent reinforcement learning (MARL) has largely focused on emergent behaviour in collaborative dynamics between multiple agents (Zhang et al., 2017; Nguyen et al., 2020; Wai et al., 2018; Zhang et al., 2019; Bansal et al., 2017; Baker et al., 2019; Foerster et al., 2017). In the fully-competitive setting, MARLs are typically modeled as zero-sum Markov games, and span a variety of applications such as exploration (Baker et al., 2019; Niroui et al., 2019), control

(Hrabia et al., 2018), and others (Li et al., 2019; Kutschinski et al., 2003). Existing RL approaches in multi-agent competition have studied competitions between two agents (Littman, 1994; Mansour et al., 2017; Aridor et al., 2019) with a focus on the expected equilibrium outcome and agent strategies. In particular, Dong et al. (2019) proposes that the Nash equilibrium for two firms in similarly motivated data acquisition learning game tends toward monopoly at the expense of consumer welfare. We differ from this line of work by explicitly modeling both the predictors and user decisions, incorporating user and sampling biases into our model, and by allowing for any number of predictors and users. This flexibility is critical as we find that the quality of prediction experienced by users heavily depends on the number of competing predictors. Another substantial difference between our analysis and that proposed in Dong et al. (2019) is that ours takes into the account the particular structure of a given supervised learning algorithm. On the other hand, the analysis in Dong et al. (2019) generically assumes learning algorithms can be replaced by black-boxes that simply behave according to canonical minimax error rates.

Another body of work focuses on examining and addressing single-agent direct feedback loops present in sample selection, namely sampling bias (Nie et al., 2018; Shin et al., 2019; Zadrozny, 2004; Liu and Ziebart, 2014; Dudík et al., 2009; Huang et al., 2007, 2006; Cortes et al., 2008; Vella, 1998), but the problem remains under-explored in the case of multi-agent competition. Other forms of a feedback loop in ML systems that have been explored include social media filter bubbles (Sculley et al., 2015), risk assessment (Green and Chen, 2019), and algorithmic policing (Ensign et al., 2017). Dueling algorithms have been explored in Immorlica et al. (2011), though they did not consider any statistical learning settings.

Extensions, limitations and future works This paper proposes a model of competing predictors that enables both empirical and theoretical investigations. We characterize several interesting phenomena, namely how competition leads predictors to specialize and how too little or too much competition can both hurt the quality of prediction experienced by users. The phenomena that we capture, both empirically and theoretically, have not been studied in depth before and are interesting to the general ML community.

Because this is a relatively new direction of research in ML, we make several simplifications that allow the model to capture the essence of competition without overly complicating the insights. Most of our experiments and theory focus on the setting where each user corresponds to a single data point and only appears once. This is reasonable in applications with large populations of users and relatively infrequent repeated interaction. We conduct collaborative filtering experiments in which users and recommenders repeatedly interact over time, and find the phenomena remain. Additional investigation of repeated interactions is a fertile direction for future study.

A simplification we have made is that predictors do not directly interact with other predictors except through their competition over data. In practice, companies behind ML predictors may merge, intentionally differentiate (which could lead to further specialization), or spend money to acquire data. A more general model that captures the full game dynamics would define strategy spaces and payoffs for each predictor and user, and characterize incentive compatible strategies.Finally, we have assumed that the predictor that is selected receives the true label. In practice, there could be additional noise and time lag in the outcome that the predictor observes. This could also be interesting to model.

As prediction algorithms become increasingly widespread, how they interact with each other and the consequences of such competition are very important topics to explore.

Acknowledgements

A.G. is supported by a Stanford Bio-X Fellowship. Y.K. is supported by NIH R01HG010140. J.Z. is supported by NSF CCF 1763191, NSF CAREER 1942926, NIH P30AG059307, NIH U01MH098953 and grants from the Silicon Valley Foundation and the Chan-Zuckerberg Initiative. We would like to thank M. Tiwari and B. He for helpful discussion in the early stages of this work. We would also like to thank A. Grosskopf for editorial help with the manuscript.

References

Aridor, G., Liu, K., Slivkins, A., and Wu, Z. S. (2019). The perils of exploration under competition: A computational modeling approach. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 171–172.

Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., and Protasi, M. (2012). *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. (2019). Emergent tool use from multi-agent autocurricula.

Balassa, B. A. (1989). *Comparative advantage, trade policy and economic development*. Harvester Wheatsheaf New York.

Bansal, T., Pachocki, J., Sidor, S., Sutskever, I., and Mordatch, I. (2017). Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*.

Cortes, C., Mohri, M., Riley, M., and Rostamizadeh, A. (2008). Sample selection bias correction theory. In Freund, Y., Györfi, L., Turán, G., and Zeugmann, T., editors, *Algorithmic Learning Theory*, pages 38–53, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Dong, J., Elzayn, H., Jabbari, S., Kearns, M., and Schutzman, Z. (2019). Equilibrium characterization for data acquisition games.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Dudík, M., Elith, J., Graham, C., Lehmann, A., Leathwick, J., and Ferrier, S. (2009). Sample selection bias and presence-only distribution models: Implications for background and pseudo-absence data. *Ecological applications : a publication of the Ecological Society of America*, 19:181–97.

Ensign, D., Friedler, S. A., Neville, S., Scheidegger, C. E., and Venkatasubramanian, S. (2017). Runaway feedback loops in predictive policing. *CoRR*, abs/1706.09847.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. (2017). Counterfactual multi-agent policy gradients. *CoRR*, abs/1705.08926.

Gardner, A., Duncan, C. A., Kanno, J., and Selmic, R. (2014). 3d hand posture recognition from small unlabeled point sets. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 164–169. IEEE.

Green, B. and Chen, Y. (2019). Disparate interactions: An algorithm-in-the-loop analysis of fairness in risk assessments. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 90–99, New York, NY, USA. Association for Computing Machinery.

Hastie, T., Mazumder, R., Lee, J. D., and Zadeh, R. (2015). Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402.

Hrabia, C.-E., Lehmann, P. M., Battjbuer, N., Hessler, A., and Albayrak, S. (2018). Applying robotic frameworks in a simulated multi-agent contest. *Annals of Mathematics and Artificial Intelligence*, 84(1-2):117–138.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. J. (2007). Correcting sample selection bias by unlabeled data. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press.

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., and Scholkopf, B. (2006). Correcting sample selection bias by unlabeled data. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, page 601–608, Cambridge, MA, USA. MIT Press.

Immorlica, N., Kalai, A. T., Lucier, B., Moitra, A., Postlewaite, A., and Tennenholtz, M. (2011). Dueling algorithms. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 215–224.

Kutschinski, E., Uthmann, T., and Polani, D. (2003). Learning competitive pricing strategies by multi-agent reinforcement learning. *Journal of Economic Dynamics and Control*, 27:2207–2218.

Li, S., Wu, Y., Cui, X., Dong, H., Fang, F., and Russell, S. (2019). Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:4213–4220.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, ICML'94, page 157–163, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Liu, A. and Ziebart, B. (2014). Robust classification under sample selection bias. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 37–45. Curran Associates, Inc.

Mansour, Y., Slivkins, A., and Wu, Z. S. (2017). Competing bandits: Learning under competition. *CoRR*, abs/1702.08533.

Masoudnia, S. and Ebrahimpour, R. (2014). Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293.

Nguyen, T. T., Nguyen, N. D., and Nahavandi, S. (2020). Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Transactions on Cybernetics*, page 1–14.

Nie, X., Tian, X., Taylor, J., and Zou, J. (2018). Why adaptively collected data have negative bias and how to correct for it. In *International Conference on Artificial Intelligence and Statistics*, pages 1261–1269.

Niroui, F., Zhang, K., Kashino, Z., and Nejat, G. (2019). Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *IEEE Robotics and Automation Letters*, 4(2):610–617.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. (2015). Hidden technical debt in machine learning systems. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 2503–2511. Curran Associates, Inc.

Shin, J., Ramdas, A., and Rinaldo, A. (2019). Are sample means in multi-armed bandits positively or negatively biased? In *Advances in Neural Information Processing Systems*, pages 7102–7111.

Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286.

Tsybakov, A. B. (2008). *Introduction to nonparametric estimation*. Springer Science & Business Media.

Vella, F. (1998). Estimating models with sample selection bias: A survey. *The Journal of Human Resources*, 33(1):127–169.

Wai, H., Yang, Z., Wang, Z., and Hong, M. (2018). Multi-agent reinforcement learning via double averaging primal-dual optimization. *CoRR*, abs/1806.00877.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

Yang, X. and Ng, Y.-K. (2015). *Specialization and economic organization: A new classical microeconomic framework*. Elsevier.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 114, New York, NY, USA. Association for Computing Machinery.

Zhang, K., Yang, Z., and Başar, T. (2019). Multi-agent reinforcement learning: A selective overview of theories and algorithms.

Zhang, Y., Zhang, C., and Liu, X. (2017). Dynamic scholarly collaborator recommendation via competitive multi-agent reinforcement learning. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 331–335, New York, NY, USA. Association for Computing Machinery.

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.