Strategic Usage in a Multi-Learner Setting

Eliot Shekhtman
Cornell University

Sarah Dean
Cornell University

Abstract

Real-world systems often involve some pool of users choosing between a set of services. With the increase in popularity of online learning algorithms, these services can now self-optimize, leveraging data collected on users to maximize some reward such as service quality. On the flipside, users may strategically choose which services to use in order to pursue their own reward functions, in the process wielding power over which services can see and use their data. Extensive prior research has been conducted on the effects of strategic users in single-service settings, with strategic behavior manifesting in the manipulation of observable features to achieve a desired classification; however, this can often be costly or unattainable for users and fails to capture the full behavior of multi-service dynamic systems. As such, we analyze a setting in which strategic users choose among several available services in order to pursue positive classifications, while services seek to minimize loss functions on their observations. We focus our analysis on realizable settings, and show that naive retraining can still lead to oscillation even if all users are observed at different times; however, if this retraining uses memory of past observations, convergent behavior can be guaranteed for certain loss function classes. We provide results obtained from synthetic and real-world data to empirically validate our theoretical findings.

1 INTRODUCTION

Machine learning (ML) predictions are widely used in today's world, playing an intermediary role between individuals and services in numerous applications. In these settings, predictions rarely come from a single entity—instead, multiple service providers collect data on users and train proprietary models. While this is happening, individuals concurrently choose among these services, making selections according to their own incentives and proportionately creating a downstream impact on the data available to each service. In this broader deployment context, services must deploy learning algorithms that can contend with online data collection and shifting distributions.

An example of this can be found in digital credit services offering small short-term loans to individuals who lack access to conventional banking. This system consists of multiple services, operating largely independently and in an uncoordinated manner. Services here generally approve or deny loans using an automated system, and make initial lending decisions on the basis of an applicant's information using ML models (Francis et al., 2017). These models are trained on historical lending decisions and are updated as each service collects more data. Among other possible factors, individual users in this system are incentivized to select a provider who will approve their loan—selecting among services in order to secure a positive classification.

A large body of work on *strategic classification* (Hardt et al., 2016) studies a model of behavior in which individuals modify their data to achieve positive classifications. This body of work includes the design of decision rules that are robust to anticipated strategic behavior, as well as algorithms for finding such rules through repeated interactions between decision-makers and individuals. However, this model of data manipulation fails to capture a straightforward way in which individuals can express their preferences: simply choosing amongst alternative providers.

In this work, we formalize the problem of *strategic usage*, where individuals vary their participation in various services according to a strategic objective. We study a realizable binary classification setting where individuals seek a positive prediction and services only obtain data from users who select them. While the usage decision is relatively straightforward, the differential access to data present in this setting and the resulting multi-learner dynamics are complex. We show that when services naively update their models with retraining updates, strategic behavior by individuals can cause non-converging oscillations. Follow-

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

ing this realization, we introduce a novel class of retraining updates that make use of memory, which when used can guarantee the convergence of the learning dynamics to an invariant set regardless of initialization. These invariant sets furthermore exhibit favorable conditions: services experience zero loss across users, correctly classifying all users who choose to use them, and users whose true label is negative will elect to leave the system entirely.

The paper is organized as follows. We begin in Section 2 by reviewing the body of related work on strategic classification and usage choices. Our first contribution is introduced in Section 3, formalizing the strategic usage problem setting and notation and expanding on user and service updates with the novel addition of memory. In Section 4 we present our second contribution, being a characterization of the resulting learning dynamics. These results are illustrated in Section 5 with numerical simulations on real and synthetic data, and in Section 6 we conclude with a discussion of implications and directions for future work.

2 RELATED WORK

Strategic Classification Our work is inspired by the setting of *strategic classification*, first proposed by Hardt et al. (2016), in which strategic users manipulate their features in order to receive a positive classification, and a learner or decision maker is tasked with designing a classifier robust to these manipulations. Many works in this setting study complexities such as the distinction between gaming and improvement (Kleinberg and Raghavan, 2020), connections to causal inference (Miller et al., 2020), and the social burden (Milli et al., 2019). As this form of gaming is distinct from the one we study, we do not attempt a comprehensive review of this vast body of work; instead, we highlight work that considers phenomena relevant to our usage setting. One such phenomenon is decision-dependent access to data, which in our setting arises due to user choices between services. Harris et al. (2023) studies an online variant of the strategic classification problem in the presence of "apple tasting" or one-sided feedback, in which labels are only collected for data points that are positively classified. Chien et al. (2023) refer to this phenomenon as algorithmic censoring and explore its implications. In contrast to these works, in the strategic usage setting, both features and labels are unavailable to services that a user does not select. Another phenomenon of interest is repeated interaction between learning algorithms and strategic agents. Dong et al. (2018) present algorithms for an online variant of strategic classification in which data arrives sequentially, each point responding to the currently deployed classifier. Zrnic et al. (2021) study the dynamics of repeated interactions between a decision-maker and a strategic population, and show convergence to a unique equilibrium depending on their relative update frequencies. Interestingly, they show that repeated retraining is sufficient to counteract manipulations when their update frequencies are high enough. Though we study similar repeated retraining dynamics, our analysis differs in that equilibria are not unique.

Endogeneous Distribution Shift The dynamics of repeated interactions between learning algorithms and endogenously shifting populations have also been studied more generally. *Performative prediction*, first introduced by Perdomo et al. (2020), generalizes the setting of strategic classification, modeling a single learner seeking to maximize accuracy subject to an underlying decision-dependent data distribution. They also study the convergence of repeated retraining. Narang et al. (2022); Piliouras and Yu (2022); Wood and Dall'Anese (2022) study a multi-player scenario, where the data distribution depends on the decisions of multiple learners. We also consider decision-dependent distributions to model the varying usage of strategic individuals; however, the mechanics of the dependence violate assumptions necessary to apply previous work in the performative setting, such as distribution smoothness. In contrast the unique equilibrium of performative prediction dynamics, our setting requires a careful convergence analysis in terms of invariant sets rather than single points. The framework of *performative power* introduced by Hardt et al. (2022) studies the ability of services to influence data distributions. Interestingly, they show that in the presence of a choice between competing providers, individuals have no incentive to perform costly manipulations to their features.

Usage Choices A largely separate body of work has investigated the impacts of ML by studying user participation choices. Hashimoto et al. (2018); Zhang et al. (2019) consider sub-populations choosing whether or not to use a single ML model on the basis of accuracy or performance, showing that retention dynamics can lead to the exacerbation of disparate representation found in the population. Ginart et al. (2021); Kwon et al. (2022); Dean et al. (2022) consider users selecting among various services, also on the basis of model accuracy. For such non-strategic usage, these works characterize the convergence of a simple repeated retraining dynamic. We show that when usage decisions are made strategically, naive repeated retraining may fail to converge. Another line of work considers explicit competition for market share between multiple providers, where the market share is determined by users selecting based on performance or accuracy (Gradwohl and Tennenholtz, 2022; Aridor et al., 2020; Jagadeesan et al., 2022; Ben-Porat and Tennenholtz, 2017, 2019). While these works investigate strategic behavior on the part of services, our focus is on strategic behaviors by users. Our setting departs from all works mentioned in this subsection in that we model users who seek positive classifications, rather than merely high accuracy.

3 PROBLEM SETTING

We study the interactions between $n \in \mathbb{N}_+$ individuals, which we refer to as *users*, and $m \in \mathbb{N}_+$ machine learning-based services. Each user $i \in \{1, \dots, n\}$ is represented by features $x_i \in \mathcal{X}$, which encode the information about user i that is available at decision time. For example, in a digital credit example, this information may include data about an applicant's mobile phone usage, financial transactions, location information, and social media use, among other details (Francis et al., 2017). Also associated with each user i is a label $y_i \in \{+1, -1\}$ indicating an outcome of interest. We refer to $+1$ as a positive label, which is generally seen as desirable, and -1 as a negative label, which is generally seen as undesirable. Unlike the features, the label is not visible at decision time. In the simplified digital credit example, the label corresponds to whether an individual has the financial resources to repay a loan on time; however, users need not correspond directly to human individuals. For example, in simulation experiments, we use the Banknote Authentication dataset (Lohweg, 2013). In this setting, a user corresponds to a banknote, the features are defined by a processed image of the banknote, and the label is whether or not the banknote is authentic.

Users interact with services. Each service $j \in \{1, \dots, m\}$ selects a classifier $h_j : \mathcal{X} \to \{+1, -1\}$ from some model class $\mathcal{H}$. This classifier predicts the unseen label of a user, given their features. As with labels, a positive prediction of $+1$ is seen as desirable, while a negative prediction of -1 is seen as undesirable. In the digital credit example, a prediction determines whether a credit service offers a loan to an individual. In the banknote example, a prediction determines whether the financial transaction is accepted.

We study the *realizable* setting, meaning that we assume there exists a classifier $h \in \mathcal{H}$ that perfectly classifies all users simultaneously, $y_i = h(x_i)$ for $i = 1, \dots, m$. This assumption, formally stated in Assumption 3, is consistent with modern machine learning practice: expressive high-parameter model classes and high-dimensional data allow for state-of-the-art dataset interpolation (Zhang et al., 2017). Several works in strategic classification also use this setting (Nair et al., 2022; Ghalme et al., 2021).

Example 1. *For a given feature transformation $\varphi : \mathcal{X} \to \mathbb{R}^d$, the* linear model class *is defined as*

$$\mathcal{H} = \Big\{ h(x) = \begin{cases} +1 & \theta^\top \varphi(x) > 0 \\ -1 & \theta^\top \varphi(x) \le 0 \end{cases} \quad \textit{s.t. } \theta \in \mathbb{R}^d \Big\}.$$

Unlike in a classical supervised machine learning setting, or even in the usual strategic classification setting (Hardt et al., 2016; Zrnic et al., 2021), we do not assume that services have immediate access to data about the users. Instead, the data observed by services depends on the strategic choices of the users. The following sections describe the user behavior and the learning updates of the services.

3.1 Strategic Users

Strategic users seek positive classifications, as these correspond to desirable outcomes. This desire is independent of the user's true label. Unlike prior work on strategic classification, in our setting, users *cannot manipulate their data*, but *can select between different services* and *vary their level of usage*. Thus, the features x_i and label y_i of each user i are fixed. Instead, users strategically adjust their *usage*, denoted by $A_{ij} \in \mathbb{R}_+$ for user i and service j. Each user i selects m usage values $A_{i1}, \dots, A_{im}$. These values are non-negative and real-valued, meaning that the user can modulate the total amount of usage. For example, in the digital credit setting, usage could correspond to the number or amount of loans.

One component of the strategic user objective is the utility from positive classification. This utility $u : \mathcal{X} \times \mathcal{H} \to \mathbb{R}$ depends on the user features and the classifier. We make the following assumptions about the utility:

Assumption 1 (User utility)**.** *For any $h_1, h_2 \in \mathcal{H}$ and $x \in \mathcal{X}$ such that $h_1(x) = -1$ and $h_2(x) = 1$, $u(h_1, x) \le 0 < u(h_2, x)$.*

This assumption states the intuition that users seek positive classification. Furthermore, we allow the utility to depend on the classifier h, not merely the individual classification $h(x)$. This allows users to be sensitive to other qualities of the classifier, such as the margin, i.e. the distance from a negative classification. We illustrate this in the following example:

Example 2. *For a linear model class where classifiers are represented by their weights θ, the* 0-1 utility *is given by $u(x, \theta) = \mathbf{1}\{\theta^\top \varphi(x) > 0\}$. This models users who care only about whether their classification is positive. The linear utility is given by $u(x, \theta) = \theta^\top \varphi(x)$. This models users who care about their margin.*

The benefit that a user i receives from a service j depends on both utility and usage: $A_{ij} u(x_i, h_j)$.

The second component of the user objective depends only on their total usage: $\sum_{j=1}^m A_{ij}$. In particular, higher total usage corresponds to a lower overall objective value. This models an opportunity cost or a penalty for large usages on a per-user basis. In the digital credit example, individuals are disincentivized from applying for loans too often.

Putting these components together, the overall strategic objective for a user i is to maximize

$$\sum_{j=1}^m A_{ij} u(x_i, h_j) - \frac{1}{q} \Big(\sum_{j=1}^m A_{ij} \Big)^q \tag{1}$$

where the power $q > 1$ sets the opportunity cost to increase superlinearly, so the marginal utility of usage is non-increasing. To understand this objective, consider the best

response of a user i for fixed services $h_1, \dots, h_m$. If no service offers a positive classification, then by Assumption 1, no utility will be positive, and thus the best response will be zero usage. If there is a service j offering uniquely maximum utility $u(x_i, h_j) > 0$, then the best response is to ignore any other service, i.e., $A_{ik} = 0$ for all $k \neq j$. The best response usage will be $A_{ij} = u(h_j, x_i)^{\frac{1}{1-q}}$, thus depending on the maximal utility in addition to the power q.

If several services offer equal and maximal utility to the user, then the best response is not unique. It includes any configuration of usage distributed over these equivalent services, with fixed total usage. In other words, users consider only the services offering them a sufficiently positive classification, and their usage magnitude is determined by the magnitude of the utility.

We note that implicit in this behavior is that users have full knowledge of service classifiers. This is a common assumption in strategic classification literature and can be motivated through openly released classifiers or hearsay. In the imperfect knowledge setting, we add that services observe users that they classify as negative, making the game easier for the services.

3.2 Learning Updates for Services

Services deploy classifiers trained on data that they have observed. We take the number of services to be much smaller than the number of users $m \ll n$. To each service, the large and fluid user pool is represented as a *distribution* over features and labels. The distribution observed by a service depends on the usages so that the weight on data point (x_i, y_i) for service j is proportional to the usage A_{ij}. Importantly, this means that when the usage is zero, the service has no information about the datapoint. Users who never choose to use a particular service are essentially invisible to the service.

We model the process of training a classifier by expected loss minimization. The loss $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ quantifies the error of a given classifier on a given data point. We make the following assumption about the loss function.

Assumption 2 (Loss-utility relationship)**.** *For all $h \in \mathcal{H}$, the loss is non-negative, $-y\ell(h, x, y)$ strictly monotonically increases with $u(x, h)$, and there exists $v > 0$ such that $u(x, h) = 0 \implies \ell(h, x, y) = v$.*

For a positive user ($y = +1$), this assumption gives a negative relationship between classification utility and the service's quantification of error: high utility corresponds to low error, so the goals of users and services are aligned. For a negative user ($y = -1$), it is a positive relationship: high utility corresponds to high error, so the user is at odds with the service. The requirement of the existence of a $v > 0$ meanwhile implies that zero loss gives a user the same sign utility as their label.

Example 3. *For a linear model class where classifiers are represented by their weights θ, the* 0-1 loss *is $\ell(\theta, x, y) = \mathbf{1}\{\theta^\top \varphi(x) \cdot y > 0\}$, which corresponds to the 0-1 utility. The* hinge loss *is defined as $\ell(\theta, x, y) = \max\{1 - \theta^\top \varphi(x) \cdot y, 0\}$, which corresponds to the linear utility.*

Recall that as discussed above, realizability means that the model class $\mathcal{H}$ includes a perfect classifier. We now formalize this intuition in terms of the loss function.

Assumption 3 (Realizability)**.** *There exists a classifier $h \in \mathcal{H}$ such that $\ell(h, x_i, y_i) = 0$ for all $i = 1, \dots n$.*

Example 4. *For the linear model class and either the 0-1 loss or the hinge loss, the realizability assumption is satisfied as long as the features $\{\varphi(x_1), \dots, \varphi(x_n)\}$ are linearly separable with a strictly positive margin.*

This assumption is equivalent to ensuring that there exists a classifier that achieves zero expected loss on the entire population of users. For a given user distribution $\mathcal{D}$, the expected loss of a classifier h is $\mathbb{E}_{x,y\sim\mathcal{D}}[\ell(h, x, y)]$. When this distribution is defined for a service j based on usages $A_{1j}, \dots, A_{nj}$, it can be simplified[1] to

$$\sum_{i=1}^{n} \frac{A_{ij}}{\sum_{k=1}^{n} A_{kj}} \ell(h, x_i, y_i) \,. \tag{2}$$

Therefore, when user distributions are defined by usages, the expected loss depends on the usages as well.

Many works on strategic classification considering the setting where users manipulate their features consider a principle-agent game between a single service and multiple users (Hardt et al., 2016). These works focus on showing the existence of a desirable equilibrium, where desirability is defined as correct classifications with respect to true user labels. In our setting, the realizability assumption implies that such good outcomes are possible. However, because we do not model omniscient services—they observe data depending on strategic usage—arriving at such an equilibrium through interactions requires a nontrivial analysis of dynamical and transient behavior.

3.3 Interaction Dynamics

We study the dynamics of repeated interactions between users and services, taking a cue from other lines of work on strategic classification (Zrnic et al., 2021), endogenous distribution shift (Perdomo et al., 2020), and usage dynamics (Dean et al., 2022). In these dynamics, users act according to their strategic objective, while services update their classifiers by retraining. We thus index classifiers and usage by time, and denote the state of the dynamics as

$$H^t = (h_1^t, \dots h_m^t), \quad A^t \in \mathbb{R}^{n \times m}$$

[1] If $\sum_{k=1}^{n} A_{kj} = 0$ for some service, we adopt the convention that for all users i, the fraction $A_{ij}/(\sum_{k=1}^{n} A_{kj}) = 0$.

We consider user best response dynamics, so at time t, user i selects a usage to maximize (1) given models H^t. We allow users to employ any tie-breaking scheme (even a non-deterministic one) when there is not a unique maximizing model. Consider the following joint update:

$$A^t \in \operatorname*{argmax}_{A \in \mathbb{R}_+^{n\times m}} \sum_{i=1}^{n} \left[\sum_{j=1}^{m} A_{ij} u(x_i, h_j^t) - \frac{1}{q} \Big[\sum_{j=1}^{m} A_{ij} \Big]^q \right] \quad (3)$$

We consider this joint update for simplicity of exposition, noting that it is equivalent to any order of independent user updates, due to the separability of the objective.

We consider services that repeatedly retrain their classifiers. The most naive retraining approach minimizes the immediate expected loss (2) given usages observed A^t, i.e. over the current user distribution. Such an approach has been studied in the traditional strategic classification setting (Zrnic et al., 2021; Perdomo et al., 2020), where it has been shown to converge to desirable equilibria. As we will see in Section 4.1, this *memoryless* retraining is a poor fit for the strategic usage setting.

We therefore consider retraining updates which minimize the weighted sum of prior expected losses; that is, the update to models H^{t+1} considers the expected loss (2) induced by $A^t, A^{t-1}, \dots$. By the linearity of expectation, this is equivalent to minimizing the expected loss over a *distribution with memory*. Thus, rather than depending on current usage A^t, classifiers are selected according to an average loss depending on some memory $M^t \in \mathbb{R}_+^{n\times m}$. We define memory as follows, using a discount factor $p \geq 0$:

$$M^t = \frac{A^t}{1+p} + \frac{pM^{t-1}}{1+p} \quad (4)$$

By convention, $M^0 = 0$. The memoryless case is $p = 0$, where the loss depends only on the current timestep. For $p > 0$, prior observations are retained, but with discounted weight, so that timesteps further in the past contribute less. As p increases, the memory is "stronger": prior timesteps have more influence on the loss.

Therefore, the retraining dynamics are defined by a service j selecting a classifier to minimize the expected loss defined by the memory M_t. The simultaneous joint update for services is

$$H^{t+1} \in \operatorname*{argmin}_{H \in \mathcal{H}^m} \sum_{j=1}^{m} \sum_{i=1}^{n} \frac{M_{ij}^t}{\sum_{k=1}^{n} M_{kj}^t} \ell(h_j, x_i, y_i) \,. \quad (5)$$

Due to the separability of the objective over services, this joint update is equivalent to any order of independent service updates. We thus consider the simultaneous update for simplicity of exposition.

We allow classifiers to employ many types of tie-breaking schemes when there is not a uniquely optimal classifier; however, we introduce a requirement of sticky tie-breaking.

Definition 1. *Let there be an update schema where some f^{t+1} is selected to minimize an expected loss as given by $f^{t+1} \in \operatorname{argmin}_{f\in\mathcal{F}} L^t(f)$. The update is called* sticky *when if for a given f^t, $L^{t-1}(f^t) = L^t(f^t)$, then it holds that $f^{t+1} = f^t$.*

For example, when there is a norm defined on $\mathcal{F}$, the minimum-norm update rule is

$$f^{t+1} = \operatorname*{argmin}_{f\in\mathcal{F}} \|f\|_{\mathcal{F}} \quad \text{s.t.} \quad f \in \operatorname*{argmin}_{g\in\mathcal{F}} L^t(g).$$

This update rule satisfies the stickiness property. This requirement is grounded in real-world settings as changing classifiers could often be costly or undesirable, and so when no explicit advantage is derived from replacing a model, it might be more advantageous to retain the old model rather than approve and release a new one.

We consider alternating updates between services and users, where the memory evolves according to (4).

Assumption 4. *Given state (H^t, A^t) at time t, the classifier update to H^{t+1} is* sticky *and satisfies (5), and the usage update to A^{t+1} satisfies (3).*

This means that first the services update $H^t \to H^{t+1}$ based on current usage A^t, and then the users best-respond $A^t \to A^{t+1}$ based on the updated services H^{t+1}. We remark that our analysis techniques could extend to a round-robin of updates involving varied sequences of service and user updates. We include a further discussion in the appendix and focus on the joint updates here for simplicity of exposition and notation.

4 THEORETICAL RESULTS

In this section, we formally introduce our theoretical results concerning the dynamics and convergence of the interaction defined by Assumption 4. We will show that under mild assumptions when the memory is non-zero, the system will reach a desirable point within a finite number of steps. Here, we define desirability from the perspective of accurate classification.

Definition 2. *A state (H, A) is* zero-loss *if all services j satisfy: 1) $A_{ij}\ell(h_j, x_i, y_i) = 0$ for all i and 2) $u(x_i, h_j) \leq 0$ for all i with $y_i = -1$.*

Zero-loss states are desirable from many perspectives. The first condition means that all users with nonzero usage have minimal loss. This means that all services make accurate classifications on the populations they observe. For positive users, it means maximal utility and correctly positive classifications. The second condition means that negative users will receive zero utility and will thus choose not to engage in any service.

Beyond desirability from the perspective of utility and loss, zero-loss points play an important role in understanding the

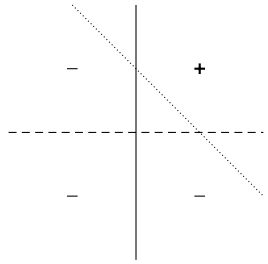

Figure 1: Five datapoint example described in the proof of Proposition 1. Negative points are represented by $-$ and positive points by $+$, where boldface indicates two overlapping points. The dashed and solid lines represent the oscillating classifiers, with the dotted line representing a zero-loss classifier.

convergence of the interaction dynamics. In Section 4.2, we show that for memory parameter $p > 0$, when a state is zero-loss, so are all future states. Thus the zero-loss property defines a set which is *invariant* under the service retraining and user best response updates. In Section 4.3, we further show that the dynamics will reach a zero-loss state in finite time.

All results are presented under Assumptions 1, 2, 3, and 4.

4.1 Importance of Memory

While naive repeated retraining can be a successful strategy for strategic classification problems in the context of data manipulation (Zrnic et al., 2021), in this section we show that it can catastrophically fail.

Proposition 1. *In the memoryless $p = 0$ setting, there exist settings in which the state (H, A) never converges.*

We prove this result by providing an example that leads to oscillations. We construct a dataset and an initial configuration satisfying all assumptions and show that classifiers and usages will fluctuate indefinitely. The dataset is illustrated in Figure 1.

Proof Sketch. We consider a setting of five users and two classifiers, illustrated in Figure 1 and described in detail in Section 5.1. We show that for $p = 0$, there are initial conditions that lead to perpendicular oscillating classifiers (dashed and solid in the figure) rather than zero-loss (for example, dotted). □

In contrast to memoryless updates, when $p > 0$, services accumulate knowledge about data distribution over timesteps. We will show that in our setting, nonzero memory is enough to guarantee convergence to a zero-loss state. Towards that goal, we first make precise the ability of a service to accumulate knowledge.

Lemma 2. *For every user service pair i, j such that $M_{ij}^t > 0$, $\ell(h_j^{t+1}, x_i, y_i) = 0$. Therefore, when $p > 0$, if $A_{ij}^t > 0$ for any timestep t, then for all further timesteps $\tau > t$ it must hold that $\ell(h_j^\tau, x_i, y_i) = 0$.*

The proof of this result, and the missing proofs of all remaining results, are presented in the appendix.

4.2 Characterizing Invariance

So far, we have established that, in order to avoid oscillations, services must have memory $p > 0$ when retraining their classifiers. Now, we turn to questions of convergence. However, unlike many related works, we do not study a setting that necessarily admits fixed points, much less unique fixed points. This occurs because we allow arbitrary user best response updates and assume realizability. This means that it is plausible for users to vary their usage among several different services indefinitely, so long as those users are positive and the services classify them correctly. As a result, some care is required to define the appropriate notion of "convergence." Instead of arguing about fixed points, we turn our attention to an *invariant set* defined by the zero-loss property. The following proposition formalizes the idea of invariance for zero-loss points.

Proposition 3. *If a state (H^t, A^t) is zero-loss at time t, then all future states are zero-loss for all times $\tau \geq t$.*

The proof of this proposition follows from the sticky assumption and users being unable to impart loss on the services without engaging in suboptimal actions. We further establish that the sticky assumption is necessary for classifier tie-breaking in order to guarantee this invariance.

Proposition 4. *Without sticky tie-breaking (Assumption 4), regardless of the value of p, the existence of a zero-loss equilibrium (H^t, A^t) at a timestep t does not guarantee the existence of zero-loss equilibria for further timesteps $\tau > t$.*

Proof Sketch. We consider a setting where a service is initialized with a model only giving positive usage to positive points, with other negative points never seen. This may be observed to be a zero-loss state. At the next timestep, or some future one, the service may feasibly re-sample a new model that gives positive usage to the negative points as they were never observed, resulting in a new state where the zero-loss conditions are violated. □

4.3 Convergence

Finally, we turn to *convergence*: regardless of the initial state of classifiers and users, we show that the interaction dynamics will lead to a zero-loss point within finite time.

Over the course of interactions between services and users, the services collect data. Each time a new user selects a service, the service may respond by updating its classifier—in particular, a change must occur if the new user is incorrectly classified. Towards ensuring convergence to a

zero-loss point, we hope to show that this process terminates. The following lemma presents a sufficient condition for reaching a zero-loss point in terms of the usage update.

Lemma 5. *For any timestep t if there exists no values $M_{i,j}^{t-1} = 0$ such that $A_{i,j}^t > 0$, then (H^t, A^t) is zero-loss.*

The proof of this follows from extending Lemma 2 to the following timestep.

With this lemma in hand, we are ready to prove the main result, which shows that services and users will converge to a zero-loss equilibrium in finitely many steps.

Theorem 6. *Given nonzero memory $p > 0$, there is a finite time $t \in \mathbb{N}$ after which for all $\tau > t$, (H^τ, A^τ) is zero-loss.*

Proof Sketch. We first argue that there are only finitely many timesteps in which the condition in Lemma 5 can occur. Therefore, the system must reach a zero-loss point, and by Proposition 3, the classifiers and usages must continue to be zero-loss. □

5 EXPERIMENTS

We illustrate our results with experiments of simulated strategic usage behavior. We first consider a synthetic data setting similar to that found in the proof of Proposition 1 to illustrate the importance of memory. We then perform experiments on real data in the setting of the Banknote Authentication task (Lohweg, 2013) and the realistic synthetic data of the Bank Account Fraud task (Jesus et al., 2022). These experiments showcase the importance of memory and illustrate the complex and initialization-dependent dynamical behaviors that strategic usage dynamics can induce.

5.1 Five Points Dataset

We begin with a synthetic example with only five points to illustrate oscillation in non-memory cases, with features:

$$\{(1,1),(1,1),(-1,1),(1,-1),(-1,-1)\}$$

and labels $\{1,1,-1,-1,-1\}$ respectively. This dataset is clearly separable by linear classifiers with $\varphi(x) = [x, 1]$ and $d = 3$ (Example 1). We consider users acting strategically according to the linear utility defined in Example 2 with services updating according to the hinge loss defined in Example 3. We consider $m = 2$ services with initial models $\theta = [1, 0, 0]$ and $\theta = [0, 1, 0]$; the dividing hyperplanes are perpendicular to one another, both giving positive classifications to the coincident positive points, but while model 0 gives positive classification to $(1, -1)$ the other gives positive classification to $(-1, 1)$. It may be observed that this setting is identical to that of the proof of Proposition 1 and is illustrated in Figure 1. Note that the fifth user is a negative point who will never choose to use either classifier, and thus will not be seen by them.

For tie-breaking, services choose models by minimizing the norm of θ (subject to achieving zero loss), while users split usage equally between models that assign equal utility to them. We run two experiments with this synthetic dataset: one being memoryless with $p = 0$ and the other using $p = 0.5$. This demonstrates how the memoryless setting may lead to oscillations while the inclusion of memory ensures convergence.

Results are presented in Figure 2, which plots the loss and usage of each service. In the $p = 0$ case, there is clear oscillation in the usages—the users at $(1, -1)$ and $(-1, 1)$ alternate between the two models. In contrast, for $p = 0.5$, the usages converge after the second epoch. Perhaps more illuminatingly the loss drops to zero: this indicates that the services have converged to a zero-loss point (Defintion 2) and no longer change between updates, unlike in the $p = 0$ case where classifiers swap directions each timestep. It must also be noted that the usages of the negative users specifically converge to 0 in the $p > 0$ case. Since the services correctly learn to assign negative classifications to the negative users, even the unseen users at $(-1, -1)$, there is no incentive for nonzero usage from negative users. Differing values for $p > 0$ were tested; however, results were similar; relevant figures are present in the appendix.

5.2 Banknote Authentication

We instantiate a semi-synthetic simulation with real-world data coming from the Banknote Authentication dataset (Lohweg, 2013). This dataset involves a binary classification problem to detect whether a banknote is genuine or forged. Banks are modeled as services and update their forgery-detection classifiers in order to reject forged banknotes while accepting genuine ones. Meanwhile, users are individuals who seek, for practical purposes, banks that allow them to cash in their notes. At the same time, banks update their forgery detection models to keep up with trends in the forgery industry. We remark that our *strategic usage* setting corresponds to the short-term dynamics of banks becoming aware of forged bank notes that are in circulation, while the classical feature manipulation setting corresponds to innovations in forgery techniques.

Features are derived from images of banknote-like specimens; specifically, they are extracted using a wavelet transform tool, resulting in $\mathcal{X} = \mathbb{R}^4$. Each sample additionally comes with a binary label. We apply the following preprocessing: we normalize the mean and variance of the features, and we transform the labels $\{+1, -1\}$.

We simulate services using scikit-learn support vector classification (SVC) (Pedregosa et al., 2011) with a radial basis function (RBF) kernel using radius $\gamma = 1$ and regularization parameter $C = 10^{10}$. This setting corresponds to linear models (Example 1) with an infinite dimensional feature function φ trained with hinge loss (Example 3).

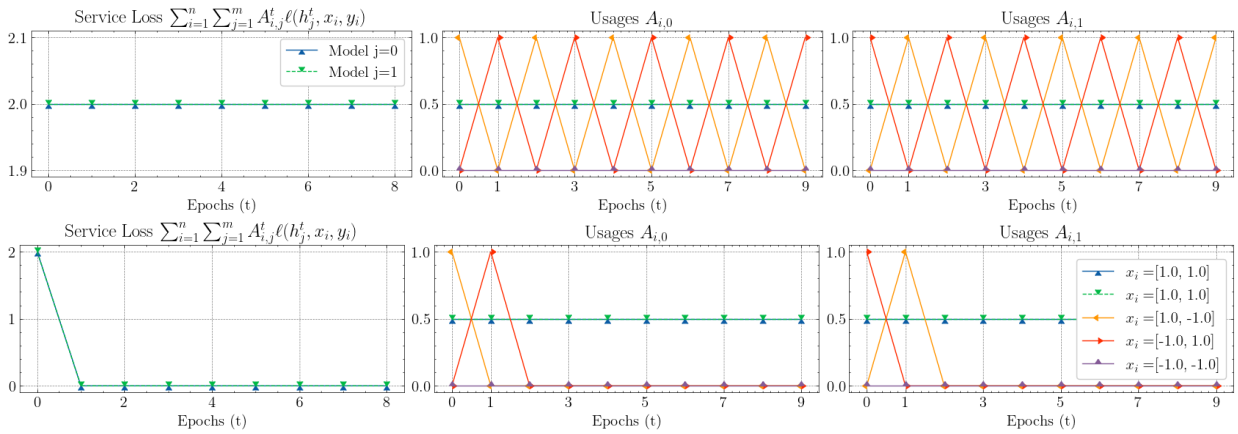

Figure 2: 5-Points dataset; the top three graphs give the $p = 0$ case while the bottom three give $p = 0.5$. Service loss is calculated after the user update but before the service update, and usages are displayed for each of the five points with the middle graphs giving the usages for model $j = 0$ and the right graphs giving the usages for model $j = 1$.

The large value of C corresponds to a small regularization weight, which approximates simply selecting the minimum norm model among all loss minimizers. In the memoryless $p = 0$ setting, services have no negative users in their loss objective at various points in time. This violates the preconditions for the scikit-learn SVC fit function, so in these instances, we preserve weights from the previous timestep.

We initialize this setting by revealing one positive and one negative user at random to each service at a timestep $t = -1$ in order to train models h_j^0 for all $j \in \{1, ..., m\}$. Random seeds for choosing these users are held constant at 100 between runs for consistency, and users tie-break through splitting usage evenly between services that provide equal usage to them.

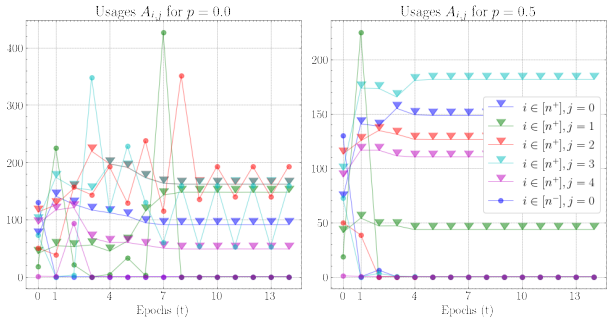

Figure 3: Banknote Authentication dataset; each graph gives the positive and negative usages of each of the five models; triangle markers above the lines indicate positive usage while below the lines indicate negative, with colors giving which model the line refers to. The left graph gives the no-memory $p = 0$ setting, while the graph on the right gives the $p > 0$ setting. Model order, and hence their colors, are meaningless due to the random initialization.

Experiments for various numbers of services, as well as varying the cost power factor q, are presented in the appendix. Figure 3 presents simulation results for $m = 5$ services, plotting the total usage of the positive and negative class over time (i.e., of genuine and forged banknotes in circulation). In the memoryless $p = 0$ case, we observe some complex transients until the tenth timestep, after which point we see oscillation between models $j = 2$ and $j = 3$. The remaining models stopped observing negative points and henceforth stopped updating; however, it is important to note that which models converged and which models continued oscillating is initialization dependent: changing the seed resulted in different dynamics, including changing the values that models' usages converged or oscillated around, in addition to changing the numbers of models converging versus oscillating.

In the nonzero memory case $p > 0$, we once again observe convergence to a zero-loss point, this time after four timesteps—the longer transient period resulting from the larger number of services that users can choose between. At this equilibrium, each model correctly assigns negative classifications to all the negative users, while the positive users are divided between the services. The particular configuration of positive users is an arbitrary result of the random initialization; figures depicting the results of other initializations lead to different final configurations and are presented in the appendix.

5.3 Bank Account Fraud

We include a third dataset as the Bank Account Fraud (BAF) dataset (Jesus et al., 2022). This dataset models the binary classification problem of predicting if a bank account opening is fraudulent or not. We interpret banks as services providing bank accounts, with users being individ-

uals attempting to open accounts. As with the banknote authentication setting, this strategic usage setting would provide the precursor to potential feature manipulation which might occur once fraudulent users realize no bank will provide them with utility.

This dataset suite was designed to stress test the performance and fairness of machine learning models on realistic tabular data generated from a real-world bank account opening fraud dataset. None of the datasets fully satisfy our realizability assumption; we selected Variant V because it was the most separable. We filtered the data points to ensure realizability by fitting a soft-margin support vector machine and removing misclassified points. Each sample has a binary label giving whether the account opening is fraudulent or not, and features include values such as income, customer age, and payment type, exhibiting both real-valued and enumerated types. We transform non-numerical features into one-hot representations before normalizing the mean and variance of all features, and labels are transformed to $\{+1, -1\}$. We present results for a subsample of $10,000$ points due to the large dataset size. Full details of our implementation and preprocessing can be found in the appendix.

We use Scikit-learn SVC (Pedregosa et al., 2011) to simulate services, with a linear kernel and regularization parameter $C = 10^{10}$. Services are initialized in a similar manner to the banknote authentication dataset, and a random seed of 100 is held constant between runs for consistency.

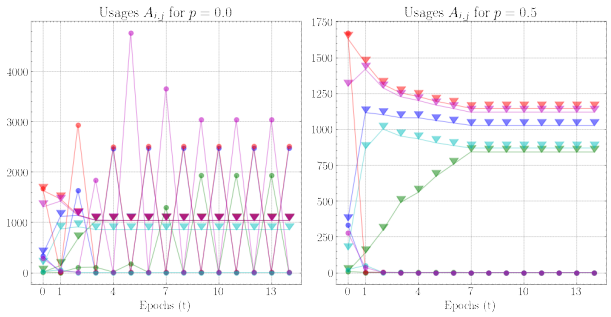

Figure 4: Bank Account Fraud dataset; each graph gives the positive and negative usages of each of the five models. Notation is the same as that for Figure 3.

Experiments for the $p = 0$ and $p > 0$ can be seen in Figure 4, for $m = 5$ services. We once again observe oscillation and non-zero usage for negative users in the memoryless $p = 0$ case, exhibiting the failure of the system to converge to a zero-loss state. The opposite can be seen in the $p > 0$ case, as no change in usage may be observed after the seventh epoch, and all negative users are observed to have left the system. Further experiments on different initializations may be observed in the appendix.

6 CONCLUSION & DISCUSSION

This work focuses on interaction dynamics between strategic users and multiple learners in an interactive setting. We formalize *strategic usage*, in which individuals choose to use services strategically in order to pursue a positive classification. Since usage presents their data to said services, the latter respond by retraining their classifiers, to the potential benefit or detriment of these strategic individuals. We provide sufficient conditions to guarantee a finite-time equilibrium, including the addition of memory to classifier retraining updates. Our work is both an extension of and in contrast to the strategic classification setting, which primarily explores dynamics involving strategic user feature modification in pursuit of goals. We remark that while several works have raised concern as to the adverse social outcomes entailed by strategic classification (Milli et al., 2019; Chen et al., 2020; Hu et al., 2019) our setting ensures that all true positives may receive positive utility at equilibrium.

As the first to study the setting of strategic usage, our work raises several areas for future extension. Though the realizable setting is well motivated for modern ML, it is natural to analyze the setting in which no single classifier can correctly classify the entire dataset. Another natural extension is to consider explicit competition between services, rather than retraining blindly, unaware of the existence of models being deployed by other services in the system. Similarly, we study un-coordinated and short-term user objectives, but an exploration of long-term strategic planning and optimization might give insight into additional real-world phenomena. Finally, it would be interesting to consider the relationship between a finite user pool and an underlying population-level distribution.

Acknowledgements

This work was supported by NSF CCF 2312774, NSF OAC-2311521, a LinkedIn Research Award, and a gift from Wayfair.

References

G. Aridor, Y. Mansour, A. Slivkins, and Z. S. Wu. Competing bandits: The perils of exploration under competition. *arXiv preprint arXiv:2007.10144*, 2020.

O. Ben-Porat and M. Tennenholtz. Best response regression. *Advances in Neural Information Processing Systems*, 30, 2017.

O. Ben-Porat and M. Tennenholtz. Regression equilibrium. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 173–191, 2019.

Y. Chen, J. Wang, and Y. Liu. Strategic recourse in linear classification. *ArXiv*, abs/2011.00355, 2020. URL https://api.semanticscholar.org/CorpusID:226226668.

J. Chien, M. Roberts, and B. Ustun. Algorithmic censoring in dynamic learning systems. *arXiv preprint arXiv:2305.09035*, 2023.

S. Dean, M. Curmei, L. J. Ratliff, J. Morgenstern, and M. Fazel. Multi-learner risk reduction under endogenous participation dynamics. *arXiv preprint arXiv:2206.02667*, 2022.

J. Dong, A. Roth, Z. Schutzman, B. Waggoner, and Z. S. Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70, 2018.

E. Francis, J. Blumenstock, and J. Robinson. Digital credit: A snapshot of the current landscape and open research questions. *CEGA White Paper*, pages 1739–76, 2017.

G. Ghalme, V. Nair, I. Eilat, I. Talgam-Cohen, and N. Rosenfeld. Strategic classification in the dark. In *International Conference on Machine Learning*, pages 3672–3681. PMLR, 2021.

T. Ginart, E. Zhang, Y. Kwon, and J. Zou. Competing ai: How does competition feedback affect machine learning? In *International Conference on Artificial Intelligence and Statistics*, pages 1693–1701. PMLR, 2021.

R. Gradwohl and M. Tennenholtz. Coopetition against an amazon. In *International Symposium on Algorithmic Game Theory*, pages 347–365. Springer, 2022.

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

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

K. Harris, C. Podimata, and Z. S. Wu. Strategic apple tasting. *arXiv preprint arXiv:2306.06250*, 2023.

T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

L. Hu, N. Immorlica, and J. W. Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 259–268, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287597. URL https://doi.org/10.1145/3287560.3287597.

M. Jagadeesan, M. I. Jordan, and N. Haghtalab. Competition, alignment, and equilibria in digital marketplaces. *arXiv preprint arXiv:2208.14423*, 2022.

S. Jesus, J. Pombal, D. Alves, A. Cruz, P. Saleiro, R. P. Ribeiro, J. Gama, and P. Bizarro. Turning the tables: Biased, imbalanced, dynamic tabular datasets for ml evaluation, 2022.

J. Kleinberg and M. Raghavan. How do classifiers induce agents to invest effort strategically? *ACM Transactions on Economics and Computation (TEAC)*, 8(4):1–23, 2020.

Y. Kwon, A. Ginart, and J. Zou. Competition over data: how does data purchase affect users? *arXiv preprint arXiv:2201.10774*, 2022.

V. Lohweg. banknote authentication. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C55P57.

J. Miller, S. Milli, and M. Hardt. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*, pages 6917–6926. PMLR, 2020.

S. Milli, J. Miller, A. D. Dragan, and M. Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019.

V. Nair, G. Ghalme, I. Talgam-Cohen, and N. Rosenfeld. Strategic representation, 2022.

A. Narang, E. Faulkner, D. Drusvyatskiy, M. Fazel, and L. J. Ratliff. Multiplayer performative prediction: Learning in decision-dependent games. *arXiv preprint arXiv:2201.03398*, 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

G. Piliouras and F.-Y. Yu. Multi-agent performative prediction: From global stability and optimality to chaos. *arXiv preprint arXiv:2201.10483*, 2022.

K. Wood and E. Dall'Anese. Stochastic saddle point problems with decision-dependent distributions. *arXiv preprint arXiv:2201.02313*, 2022.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization, 2017.

X. Zhang, M. Khaliligarekani, C. Tekin, et al. Group retention when using machine learning in sequential decision making: the interplay between user dynamics and fairness. *Advances in Neural Information Processing Systems*, 32, 2019.

T. Zrnic, E. Mazumdar, S. Sastry, and M. Jordan. Who leads and who follows in strategic classification? *Advances in Neural Information Processing Systems*, 34: 15257–15269, 2021.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Assumptions are stated and numbered.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] Included in the supplement.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] Assumptions are stated and numbered.
   (b) Complete proofs of all theoretical results. [Yes] Included in the supplement
   (c) Clear explanations of any assumptions. [Yes] Assumptions are discussed.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Missing Proofs

In this section, we restate the main theoretical results, include their full proofs, and introduce a handful of auxiliary results.

A.1 Oscillations and Counter-Examples

In this section, we present examples of dynamics that do not converge or that are not zero-loss invariant.

Proposition 1. *In the memoryless $p = 0$ setting, there exist settings in which the state (H, A) never converges.*

Proof of Proposition 1. Let there exist a set of users with features $X = \{(1,1),(1,1),(-1,1),(1,-1),(-1,-1)\}$ and labels $Y = \{1,1,-1,-1,-1\}$, selecting between two services where both services choose models from the linear model class (Example 1) using feature transformation $\phi(x) = (x, 1)$, using linear utility (Example 2) and hinge loss (Examples 3). Suppose that the initial models are defined through parameters $\theta_1 = [1,0,0]^\top$ and $\theta_2 = [0,1,0]^\top$, that retraining updates tie-break by choosing the minimum norm classifier, and that users tie-break by dividing usage equally between services.

Optimal updates may be calculated by analyzing stable points of repeated gradient updates on the best-response updates found in Equations 3 and 5noting that with $p = 0$, $M^t = A^t$ for any timestep t. This algebra gives us the following update steps:

$$\theta^0 = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}^\top \qquad A^0 = \begin{bmatrix}0.5 & 0.5 & 1 & 0 & 0\\ 0.5 & 0.5 & 0 & 1 & 0\end{bmatrix}^\top$$
$$\theta^1 = \begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 0\end{bmatrix}^\top \qquad A^1 = \begin{bmatrix}0.5 & 0.5 & 0 & 1 & 0\\ 0.5 & 0.5 & 1 & 0 & 0\end{bmatrix}^\top$$
$$\theta^2 = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}^\top \qquad A^2 = \begin{bmatrix}0.5 & 0.5 & 1 & 0 & 0\\ 0.5 & 0.5 & 0 & 1 & 0\end{bmatrix}^\top$$

It may be observed that $\theta^2 = \theta^0 \neq \theta^1$, and $A^2 = A^0 \neq A^1$. Given the memoryless setting in which (H^{t+1}, A^{t+1}) depend only on the previous (H^t, A^t), we may conclude that this system will oscillate for all further timesteps and will not reach a fixed state. □

Proposition 4. *Without sticky tie-breaking (Assumption 4), regardless of the value of p, the existence of a zero-loss equilibrium (H^t, A^t) at a timestep t does not guarantee the existence of zero-loss equilibria for further timesteps $\tau > t$.*

Proof of Proposition 4. Let there exist a set of users with features $X = \{(5),(-5),(0)\}$ and labels $Y = \{1,-1,1\}$, making usage decisions in a system with one service choosing models from the linear model class (Example 1) using feature transformation $\phi(x) = (x, 1)$, using linear utility (Example 2) and hinge loss (Examples 3). Suppose that the model is defined through parameters $\theta = [1,-1]^\top$, with retraining updates tie-breaking stochastically.

Optimal updates may be calculated by analyzing stable points of repeated gradient updates on the best-response updates found in Equations 3 and 5. The first update step can be seen as follows:

$$\theta^0 = \begin{bmatrix}1 & -1\end{bmatrix}^\top \qquad A^0 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top$$

It may be noted that this constitutes a zero-loss equilibrium, as for all users i, $A_i^0 \ell(h^0, x_i, y_i) = 0$, and for the negative point, $\ell(h^0, x_i, y_i) < 1$. Despite this, it is feasible for the following second update step to occur due to H not being constrained by sticky tie-breaking:

$$\theta^1 = \begin{bmatrix}1 & -0.5\end{bmatrix}^\top \qquad A^1 = \begin{bmatrix}1 & 0 & 0.5\end{bmatrix}^\top$$

Here, it can been that for the point (0), $A_i^1 \ell(h^1, x_i, y_i) = 0.25$. As such, state (H^1, A^1) does not constitute a zero-loss equilibrium, proving the claim. □

A.2 Invariance

We next turn to our results about the invariance of zero-loss points. We begin with lemmas characterizing important properties of the service retraining update.

Lemma 2. *For every user service pair i, j such that $M_{ij}^t > 0$, $\ell(h_j^{t+1}, x_i, y_i) = 0$. Therefore, when $p > 0$, if $A_{ij}^t > 0$ for any timestep t, then for all further timesteps $\tau > t$ it must hold that $\ell(h_j^\tau, x_i, y_i) = 0$.*

Proof of Lemma 2. By the separability of Equation 5 on services, we may analyze the best-response update of a particular service j. Let us call H_j^{t+1} the set of models such that the best-response update $h_j^{t+1} \in H_j^{t+1}$.

$$H_j^{t+1} := \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{i=1}^n \frac{M_{ij}^t}{\sum_{k=1}^n M_{kj}^t} \ell(h_j, x_i, y_i)$$

Since terms of the sum where $M_{i,j}^t = 0$ will simply be zero we may simplify the sum, and furthermore in the interest of brevity let us represent $\frac{M_{ij}^t}{\sum_{k=1}^n M_{kj}^t}$ as $r_{i,j}(M^t)$.

$$= \sum_{i \in \{i | M_{i,j}^t > 0\}}^n r_{i,j}(M^t) \ell(h_j, x_i, y_i) \tag{6}$$

By Assumption 3, we have that there exists an $h^* \in \mathcal{H}$ such that $\forall i \in \{1, ..., n\}$, $\ell(h^*, x_i, y_i) = 0$. By the non-negativity of ℓ and M it must hold that $r_{i,j}(M^t)\ell(h_j, x_i, y_i) \geq 0$ for all i, j pairs, and since substituting h^* into expression 6 would return a sum of 0, it must be that 0 is the minimum value of the sum. For all $h \in \mathcal{H}$ such that $\ell(h, x_i, y_i) > 0$ for some user i where $M_{i,j}^t > 0$, the value of the sum would be greater than 0 and therefore h wouldn't minimize the sum so $h \notin H_j^{t+1}$. Finally, we may conclude that for all $h_j \in H_j^{t+1}$, $\ell(h, x_i, y_i) \leq 0$ for all i such that $M_{i,j}^t > 0$.

For the second statement, we may first observe that at any timestep t, if $A_{i,j}^t > 0$ then by Equation 4 and the non-negativity of M, $M_{i,j}^t > 0$. For the $p > 0$ case, if at some timestep t we have that for some user i and service j, $M_{i,j}^t > 0$, then it will hold that $M_{i,j}^{\tau-1} > 0$ for all timesteps $\tau > t$. This may be trivially observed through the non-negativity of A and the memory update given by Equation 4.

By the conclusions drawn above, since $M_{i,j}^{\tau-1} > 0$ we may conclude that $\ell(h_j^\tau, x_i, y_i) \leq 0$. □

Next, we introduce an auxiliary lemma that formalizes the best response behavior of users. This lemma makes rigorous the informal discussion at the end of Section 3.1.

Lemma 7. *For a given user i, let there be a set of services J such that $u_0 = u(x_i, h_j^t) > u(x_i, h_k^t)$ for all services $j \in J$ and other services $k \notin J$. The user best response at time t must satisfy $u_0^{1/(q-1)} = \sum_{j \in J} A_{i,j}^t > \sum_{k \notin J} A_{i,k}^t = 0$.*

Proof of Lemma 7. Analyzing the strategic objective for a user i (1) and denoting it as L_i^u for convenience, we can begin by seeing that given some set of services J such that $u(x_i, h_{j_1}) = u(x_i, h_{j_2}) = C_0$, if we hold $\sum_{j \in J} A_{i,j} = C_1$ constant and $A_{i,k}$ constant for all $k \notin J$ such that $\sum_{k \notin J} A_{i,k} = C_2$ and $\sum_{k \notin J} A_{i,k} u(x_i, h_k) = C_3$, then any distribution of usages between services in J doesn't affect the value of the strategic objective.

$$\begin{aligned}
L_i^u(A, H) &= \sum_{j=1}^m A_{i,j} u(x_i, h_j) - \frac{1}{q} \left(\sum_{j=1}^m A_{i,j} \right)^q \\
&= \sum_{j \in J} A_{i,j} u(x_i, h_j) + \sum_{k \notin J} A_{i,k} u(x_i, h_k) - \frac{1}{q} \left(\sum_{j \in J} A_{i,j} + \sum_{k \notin J} A_{i,k} \right)^q \\
&= C_0 \sum_{j \in J} A_{i,j} + C_3 - \frac{1}{q} \left(\sum_{j \in J} A_{i,j} + C_2 \right)^q \\
&= C_0 C_1 + C_3 - \frac{1}{q} (C_1 + C_2)^q
\end{aligned}$$

Therefore, without loss of generality, let us say all usage in J is concentrated into a service $j \in J$; since $A_{i,j'} = 0$ for all $j' \in J, j' \neq j$ we can drop them from the strategic objective and consider only a service j and services $k \notin J, k \neq j$.

Taking the derivative of $L_i^u(A, H)$ with respect to some $A_{i,j}$, we get gradient $\frac{d}{dA_{i,j}} L_i^u(A, H) = u(x_i, h_j) - \left(A_{i,j} + \sum_{k \neq j} A_{i,k} \right)^{q-1}$. Let us take two services j and k, such that $u(x_i, h_j) > u(x_i, h_l)$ for all $l \neq j$. Service k

is chosen arbitrarily such that $j \neq k$. We shall now enumerate options in a case analysis focusing on the total usage of user i. We shall show that as total usage increases, the model is incentivized to concentrate all usage on the service that provides the highest utility as the derivative of the objective function with respect to other services becomes negative.

Let us say that $\sum_{l=1}^{m} A_{i,l} = 0$. For service j, we have that $\frac{d}{dA_{i,j}} L_i^u = u(x_i, h_j) > 0$; therefore $A_{i,j}$ is below the optimum.

Let us say $\sum_{l=1}^{m} A_{i,l} < u(x_i, h_k)^{\frac{1}{q-1}}$. For service j, we have that $\frac{d}{dA_{i,j}} L_i^u = u(x_i, h_j) - (\sum_{l=1}^{m} A_{i,l})^{q-1} > u(x_i, h_j) - u(x_i, h_k) > 0$; therefore $A_{i,j}$ is below the optimum. For service k, we have that $\frac{d}{dA_{i,k}} L_i^u = u(x_i, h_k) - (\sum_{l=1}^{m} A_{i,l})^{q-1} > u(x_i, h_k) - u(x_i, h_k) = 0$; therefore $A_{i,k}$ is below the optimum.

Let us say $\sum_{l=1}^{m} A_{i,l} = u(x_i, h_k)^{\frac{1}{q-1}}$. For service j, we have that $\frac{d}{dA_{i,j}} L_i^u = u(x_i, h_j) - (\sum_{l=1}^{m} A_{i,l})^{q-1} = u(x_i, h_j) - u(x_i, h_k) > 0$; therefore $A_{i,j}$ is below the optimum. For service k, we have that $\frac{d}{dA_{i,k}} L_i^u = u(x_i, h_k) - (\sum_{l=1}^{m} A_{i,l})^{q-1} = u(x_i, h_k) - u(x_i, h_k) = 0$; therefore $A_{i,k}$ at an optimum.

Let us say $u(x_i, h_k)^{\frac{1}{q-1}} < \sum_{l=1}^{m} A_{i,l} < u(x_i, h_j)^{\frac{1}{q-1}}$. For service j, we have that $\frac{d}{dA_{i,j}} L_i^u = u(x_i, h_j) - (\sum_{l=1}^{m} A_{i,l})^{q-1} > u(x_i, h_j) - u(x_i, h_j) = 0$; therefore $A_{i,j}$ is below the optimum. For service k, we have that $\frac{d}{dA_{i,k}} L_i^u = u(x_i, h_k) - (\sum_{l=1}^{m} A_{i,l})^{q-1} < u(x_i, h_k) - u(x_i, h_k) = 0$; therefore $A_{i,k}$ is above the optimum.

Let us say $\sum_{l=1}^{m} A_{i,l} = u(x_i, h_j)^{\frac{1}{q-1}}$. For service j, we have that $\frac{d}{dA_{i,j}} L_i^u = u(x_i, h_j) - (\sum_{l=1}^{m} A_{i,l})^{q-1} = u(x_i, h_j) - u(x_i, h_j) = 0$; therefore $A_{i,j}$ is at an optimum. For service k, we have that $\frac{d}{dA_{i,k}} L_i^u = u(x_i, h_k) - (\sum_{l=1}^{m} A_{i,l})^{q-1} = u(x_i, h_k) - u(x_i, h_j) < 0$; therefore $A_{i,k}$ is above the optimum.

Let us say $\sum_{l=1}^{m} A_{i,l} > u(x_i, h_j)^{\frac{1}{q-1}}$. For service j, we have that $\frac{d}{dA_{i,j}} L_i^u = u(x_i, h_j) - (\sum_{l=1}^{m} A_{i,l})^{q-1} < u(x_i, h_j) - u(x_i, h_j) = 0$; therefore $A_{i,j}$ is above the optimum. For service k, we have that $\frac{d}{dA_{i,k}} L_i^u = u(x_i, h_k) - (\sum_{l=1}^{m} A_{i,l})^{q-1} < u(x_i, h_k) - u(x_i, h_j) < 0$; therefore $A_{i,k}$ is above the optimum.

From all of this, we can see that the stable equilibrium holds that $A_{i,j} = u(x_i, h_j)^{\frac{1}{q-1}}$, and that for all services $k \neq j$, $A_{i,k} = 0$ because of the non-negativity of A. □

Corollary 8. *For any user i and service j pair such that at timestep t, $u(x_i, h_j^t) \leq 0$, then $A_{i,j}^t = 0$.*

Proof of Corollary 8. By Lemma 7, we have that if there exists some service k such that $u(x_i, h_k^t) > u(x_i, h_j^t)$ then $A_{i,j}^t = 0$. Let us analyze the case where for all users k, $u(x_i, h_j^t) = 0 \geq u(x_i, h_k^t)$. Once again taking the derivative of the strategic objective for a user i (1) with respect to $A_{i,j}^t$ and denoting it as $\frac{d}{dA_{i,j}^t} L_i^u(A^t, H^t)$ for convenience, we can see that if $\sum_{k=1}^{m} A_{i,k}^t > 0$ then $\frac{d}{dA_{i,j}^t} L_i^u(A^t, H^t) < 0$. This indicates that the optimum value of $A_{i,j}^t$ is 0. □

Finally, we prove the main invariance result.

Proposition 3. *If a state (H^t, A^t) is zero-loss at time t, then all future states are zero-loss for all times $\tau \geq t$.*

Proof of Proposition 3. We prove the proposition by showing that if a state (H^t, A^t) is a zero-loss equilibrium, then (H^{t+1}, A^{t+1}) is also a zero-loss equilibrium.

We begin by arguing that $H^{t+1} = H^t$. Consider the retraining objective for service j at t: $\sum_{i=1}^{n} \frac{M_{i,j}^t}{\sum_{k=1}^{n} M_{k,j}^t} \ell(h_j^t, x_i, y_i)$ By Lemma 2, we know that $\ell(h_j^t, x_i, y_i) = 0$ for all users i and services j such that $M_{i,j}^{t-1} > 0$. By zero-loss condition 1 we have that $\ell(h_j^t, x_i, y_i) = 0$ for all users i and services j such that $A_{i,j}^t > 0$. Thus by the definition of memory (4), it must be that $\ell(h_j^t, x_i, y_i) = 0$ for all users i and services j such that $M_{i,j}^t > 0$. We conclude that h_j^t achieves zero retraining loss, and therefore by the definition of sticky tie-breaking (1), $h_j^{t+1} = h_j^t$ for all services j. This implies that the zero-loss condition 2 holds: $0 \geq u(x_i, h_j^t) = u(x_i, h_j^{t+1})$.

We next argue that if A^t is zero loss, so is A^{t+1}. First, consider negative users i with $y_i = -1$. By the zero-loss condition 2 shown in the previous paragraph $u(x_i, h_j^t) \leq 0$. Therefore, by Corollary 8 the best response is $A_{ij}^{t+1} = 0$ for all j and all negative users i, thus ensuring that zero-loss condition 1 holds for negative users i.

Now, consider positive users i with $y_i = +1$. Because $H^{t+1} = H^t$ the user best-response objective remains the same. By monotonicity of ℓ, we have that there exists some value v' such that if $\ell(h, x, +1) = 0$, $u(x, h) = v'$. As we know that $\ell(h_j^t, x_i, y_i) = 0$ for all $A_{i,j}^t > 0$, we have that for all $A_{i,j}^t > 0$, $u(x_i, h_j^t) = v'_{ij}$. Let us say that there exists some positive

user i and service j such that $\ell(h_j^{t+1}, x_i, y_i) > 0$, meaning $u(x_i, h_j^{t+1}) < v'_{ij}$. By Lemma 7, this implies that $A_{i,j}^{t+1} = 0$. As such, for all i, j, if $\ell(h_j^{t+1}, x_i, y_i) > 0$ then $A_{i,j}^{t+1} = 0$; therefore, $A^{t+1}\ell(h_j^{t+1}, x_i, y_i) = 0$ for all users i and services j. This satisfies zero-loss condition 1 for positive users.

Thus we have shown that if a state (H^t, A^t) is a zero-loss equilibrium, then (H^{t+1}, A^{t+1}) is also a zero-loss equilibrium. Through induction, this guarantees that the state (H^τ, A^τ) is a zero-loss equilibrium for all timesteps $\tau > t$. □

A.3 Convergence

With invariance results in hand, we can now prove the main convergence result.

Lemma 5. *For any timestep t if there exists no values $M_{i,j}^{t-1} = 0$ such that $A_{i,j}^t > 0$, then (H^t, A^t) is zero-loss.*

Proof of Lemma 5. Lets say that there exists some timestep t such that there exists no values $M_{i,j}^{t-1} = 0$ such that $A_{i,j}^t > 0$. By Lemma 2 and by the non-negativity of ℓ, this means that $\ell(h_j^t, x_i, y_i) = 0$ for all users i and services j such that $M_{i,j}^{t-1} > 0$. This gives us that for all users i and services j, $\frac{M_{i,j}^{t-1}}{\sum_{k=1}^n M_{k,j}^{t-1}}\ell(h_j^t, x_i, y_i) = 0$ as either $M_{i,j}^{t-1} = 0$ or $\ell(h_j^t, x_i, y_i) = 0$. Since $A_{i,j}^t = 0$ when $M_{i,j}^{t-1} = 0$ and when $M_{i,j}^{t-1} > 0$ it holds that $\ell(h_j^t, x_i, y_i) = 0$, we can similarly conclude that for all i, j, $A_{i,j}^t\ell(h_j^t, x_i, y_i) = 0$. This satisfies the first condition of zero-equilibrium states.

Let us define an L_i^u as follows:

$$L_i^u(A, H) = \frac{1}{q}\left(\sum_{j=1}^m A_{i,j}\right)^q - \sum_{j=1}^m A_{i,j}u(x_i, h_j)$$

We know that for all $M_{i,j}^{t-1} > 0$, $\ell(h_j^t, x_i, y_i) = 0$ by Lemma 2 and the non-negativity of ℓ. This indicates that for all i such that $y_i = -1$, if $M_{i,j}^{t-1} > 0$ then $u(x_i, h_j^t) \le 0$. By Corollary 8, this gives that $A_{i,j}^t = 0$. Since $A_{i,j}^t = 0$ if $M_{i,j}^{t-1} = 0$ as well, this indicates that for all services j, for all users i such that $y_i = -1$, $A_{i,j}^t = 0$.

Let us say for contradiction that for some user i such that $y_i = -1$, for service $j = \text{argmax}_j\, u(x_i, h_j^t)$ it holds that $u(x_i, h_j^t) > 0$. By Lemma 7, it must hold that $A_{i,j}^t > 0$. This poses a contradiction, and therefore we may state that for all i such that $y_i = -1$, for all j, $u(x_i, h_j^t) \le 0$. □

Theorem 6. *Given nonzero memory $p > 0$, there is a finite time $t \in \mathbb{N}$ after which for all $\tau > t$, (H^τ, A^τ) is zero-loss.*

Proof of Theorem 6. We may analyze each timestep t as either a timestep in which there exists some user-service pair i, j such that $A_{i,j}^t > 0$ and $M_{i,j}^{t-1} = 0$, or in which no such pair exists. There are only $n \times m$ such i, j pairs and as such a maximum of nm timesteps where the first condition is satisfied. In the contrary, at any step t where the condition isn't satisfied, by Lemma 5, we have shown that we have reached a fixed point. Therefore, there is a maximum of nm timesteps before the conditions are reached to reach a zero-loss equilibrium. By Proposition 3, this indicates that for all timesteps $\tau \ge nm$, (H^τ, A^τ) constitutes a zero-loss equilibrium. □

We construct examples to indicate the linear dependence of convergence time on n and m. Let us define the model class as $h(x) = \text{sign}(x + \theta)$, with utility as $u(x, h) = \min\{x + \theta, 1\}$ and loss as $\ell(h, x, y) = \max\{0, 1 - y(x + \theta)\}$. We set $q = 2$ and $p > 0$. User tie-breaking is done by stochastically selecting one service to use, and services tie-break by choosing the minimum change classifier between timesteps.

Example 5. *Let there be n evenly spaced positive users, with features:*

$$\{(0), (-0.7), (-1.4), ..., (-0.7)(2 - n), (-0.7)(1 - n)\} : .$$

If $m = 1$ service is instantiated with model $\theta^0 = 0.5$, at every timestep t, user t will receive positive utility and elect for positive usage, pushing the classifier to $\theta^t = 1 - x_t$. This will result in $nm = n$ total timesteps before convergence.

Example 6. *Let there be $n = 1$ user such that $X = \{(0)\}$ and $Y = \{-1\}$. If m services are instantiated with models $\theta_j^0 = j + 1$, at every timestep t, some service j will receive usage from the user, pushing the classifier to $\theta_j^t = -1$, at which point it'll never receive positive usage again. This will result in $nm = m$ total timesteps before convergence.*

A.4 Round Robin Updates

It might not be realistic that users and services update on the same schedule: while prior proofs assume that services conduct one synchronous update based on current usage and users synchronously best respond once based on the updated

services at every timestep, it could be that services only update every several years while users may reallocate usage yearly. Additionally, users and services don't necessarily update synchronously, and one might conduct several updates at a time while others only do one. In this setting, a timestep stops being a feasible metric of progression; instead, we generalize to the concept of a round.

Instead of there being one joint user and one joint service update as in a timestep, we generalize rounds to users and services updating asynchronously and differing numbers of times. We maintain three constraints on this system. First, rounds are divided into alternating user update periods and service update periods, such that only users or services are updating at a time. Second, each round must contain at least one user period and one service period. Finally, each user and each service undergoes at least one best response update in each period.

Proposition 9. *Given nonzero memory $p > 0$, there there are a finite number of rounds $r \in \mathbb{N}$ after which for all $\rho > r$, (H^ρ, A^ρ) is zero-loss.*

Proof of Proposition 9. We shall prove this by showing that a round is functionally equivalent to a set of timesteps that assume stochastic user tie-breaking. Since we have that a zero-loss point will be reached after a finite number of timesteps by Theorem 6, this will imply the existence of a zero-loss point after a finite number of rounds.

As we have already analyzed the result of a service best responding to a set of users, and a user best responding to a set of services, let us analyze the change when services and users undergo multiple consecutive updates.

Given a service j at update $k > 0$, let us denote h_j^k as the best response to memory M_j^k updating on usages A. By the memory update (4), we have that for all users i, $M_{i,j}^k = 0$ if and only if $M_{i,j}^{k+1} = 0$. By Lemma 2, we have that $M_{i,j}^{k+1}\ell(h_j^k, x_i, y_i) = 0$ for all users i. As h_j^k achieves a value of zero for the objective value of the service update, by the non-negativity of loss and M we have that h_j^k is a best-response to M^{k+1}. Sticky updating gives that $h_j^{k+1} = h_j^k$.

Given a user i at update $k > 0$, let us denote A_i^k as the best response to services H. A_i^{k+1} will be a user best response to H; since no other variables are involved in the user objective function, this is equivalent to choosing a new sample A_i^k from the equivalence class of usages that maximizes the user objective function on H.

Due to the separability of the joint user and service updates, these asynchronous updates can be reanalyzed as parts of the joint update. As such, we can collapse all consecutive user updates and all consecutive service updates; without loss of generality, rounds can now be seen as a series of alternating user and service updates. This can be re-indexed as a series of timesteps, each composed of one joint user and one joint service update.

By Theorem 6, this concludes the proof. □

B ADDITIONAL EXPERIMENTS

We present additional experiments on the settings introduced in Section 5.

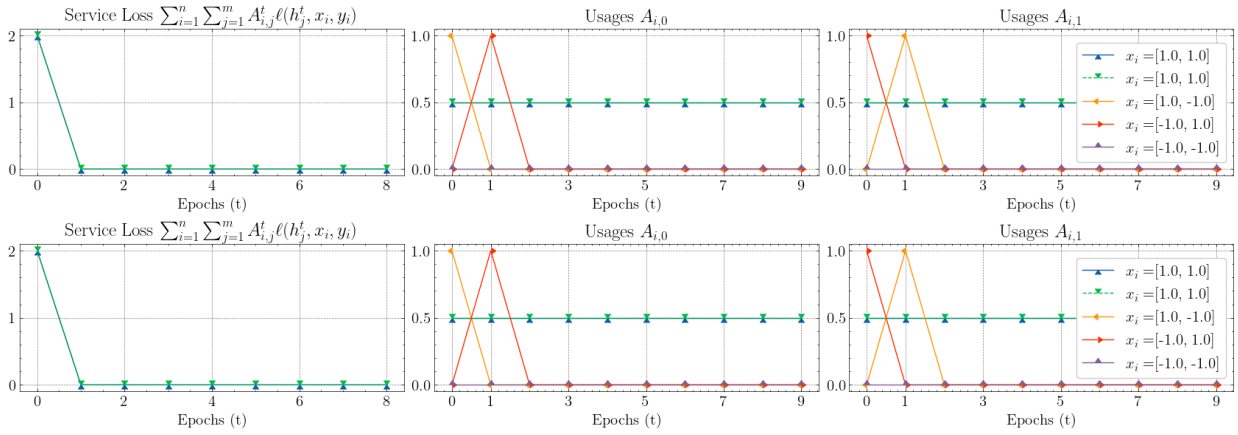

Figure 5: 5-Points dataset; the top three graphs give the $p = 0.1$ case while the bottom three give $p = 1.0$.

Synthetic Dataset In Figure 5, we show that different values of $p > 0$ don't affect the convergence. The top set of plots gives service loss and usages for $p = 0.1$ and the bottom set gives the same for $p = 1$; however, both values of p give the same usages and losses as each other across epochs.

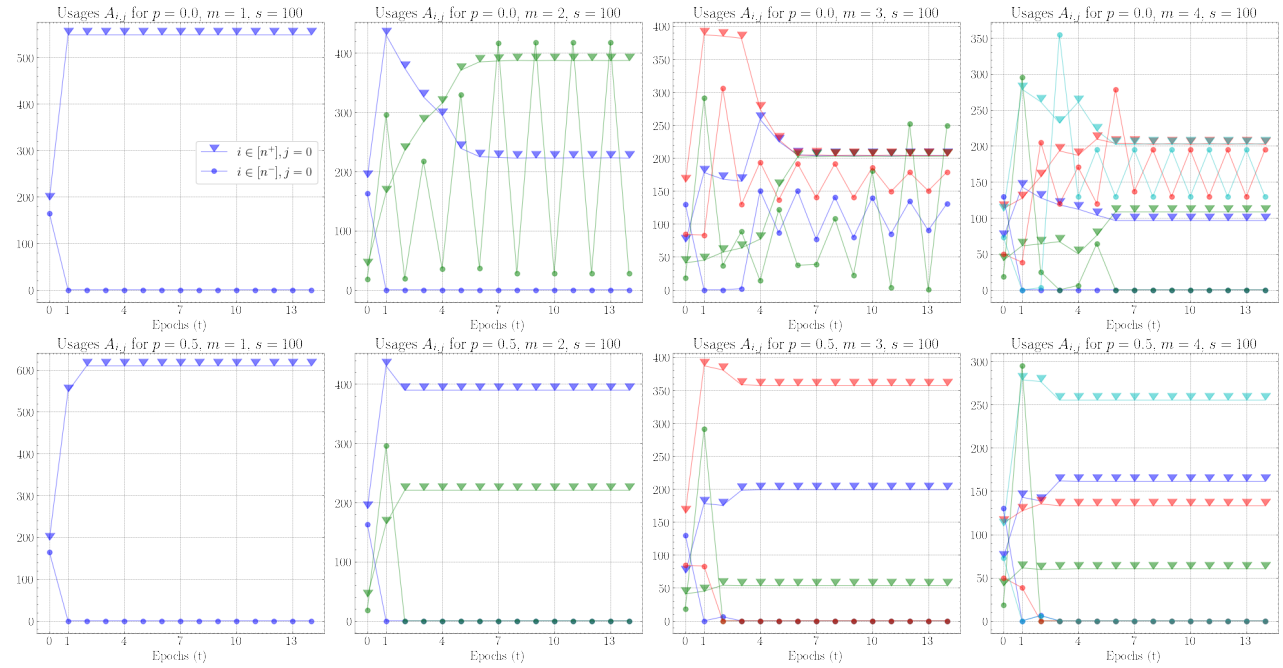

Figure 6: Banknote Authentication dataset; ablating on m. The top four graphs give the $p = 0$ case while the bottom four give $p = 0.5$. In each graph, triangle markers indicate positive usages while circular markers indicate negative usages; colors indicate the different services.

Banknote Authentication Figure 6 demonstrates how varying numbers of services can affect convergence. Plots for $m = 1, 2, 3, 4$ are given, both in the zero memory and nonzero memory cases. This illustrates that as the number of services increases, convergence may take longer due to services interfering with each other and disincentivizing users to reveal themselves to other services through usage. Note that due to the static seed, between graphs models are shown the same initial users when the model is present.

We illustrate the potential for a variety of outcomes depending on the initial seed in Figure 7. Generally, this shows that the initial conditions can have drastic effects on both the time to convergence and the final stable state.

To inspect the effect of cost power factor q on convergence, we provide usages over epochs for varying values of q both with and without memory in Figure 8. No meaningful relationship with convergence is found for $q > 1$.

Bank Account Fraud All experiments were run for $m = 5$ services, with cost power factor $q = 2$. From the realizable subset of the data, we select the first $5,000$ positive and $5,000$ negative points to speed up runtime.

Results under a variety of initial seeds can be seen in Figure 9, including both results with and without memory. These results further corroborate the earlier theoretical findings, demonstrating convergence in the memory case with oscillation commonly occurring in the naïve retraining setting.

B.1 Hardware and Specifications

All experiments were run on an Intel(R) Core(TM) i9-10885H CPU @ 2.40GHz. Our code is available at `https://github.com/eliotshekhtman/strategic-usage`.

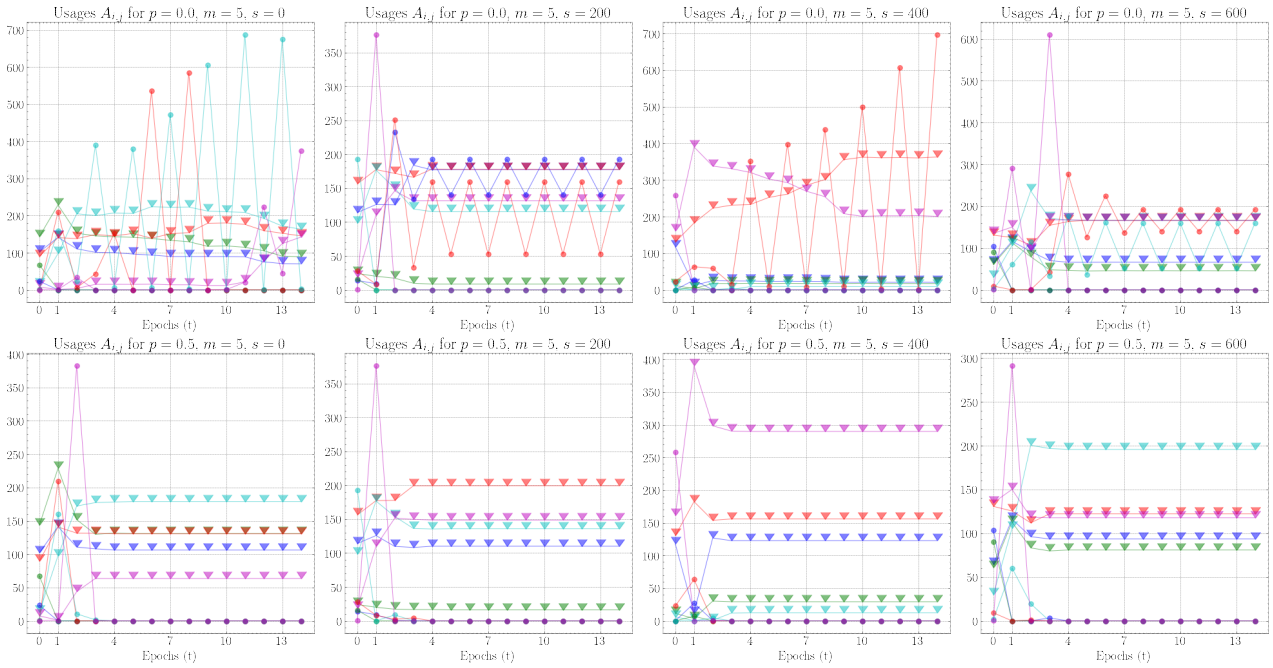

Figure 7: Banknote Authentication dataset; ablating on the seed (s). The top four graphs give the $p = 0$ case while the bottom four give $p = 0.5$. In each graph, triangle markers indicate positive usages while circular markers indicate negative usages; colors indicate the different services.

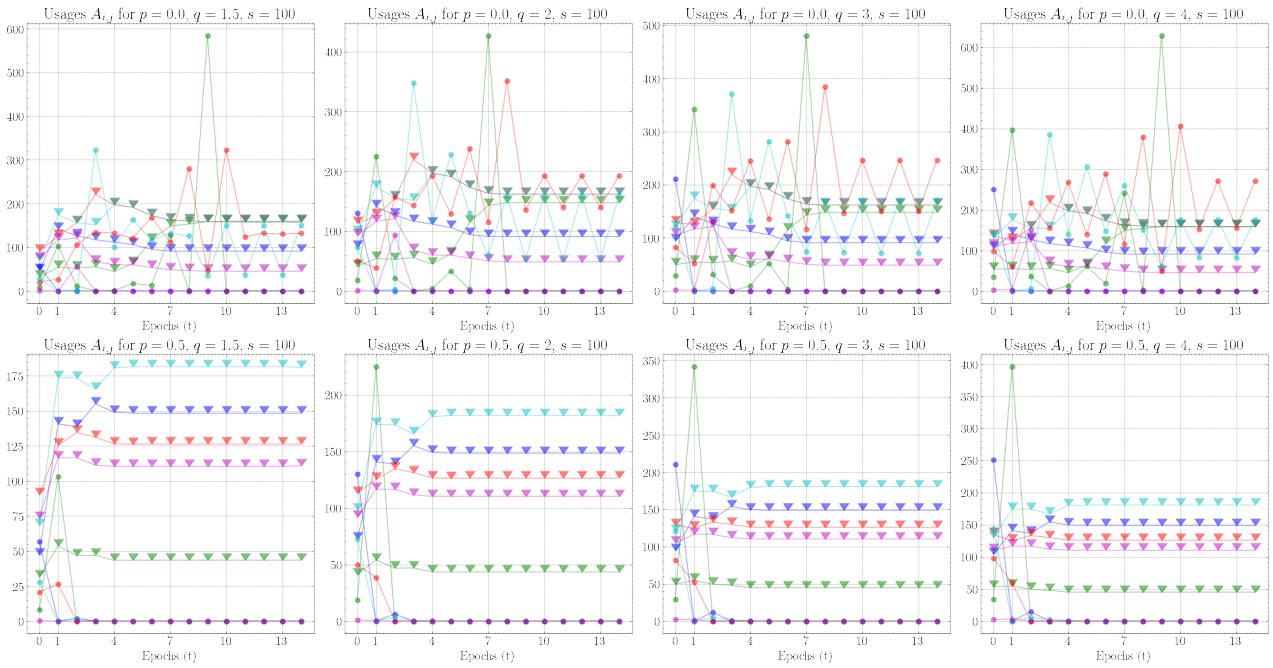

Figure 8: Banknote Authentication dataset; ablating on usage cost power factor q. The top four graphs give the $p = 0$ case while the bottom four give $p = 0.5$. In each graph, triangle markers indicate positive usages while circular markers indicate negative usages; colors indicate the different services.

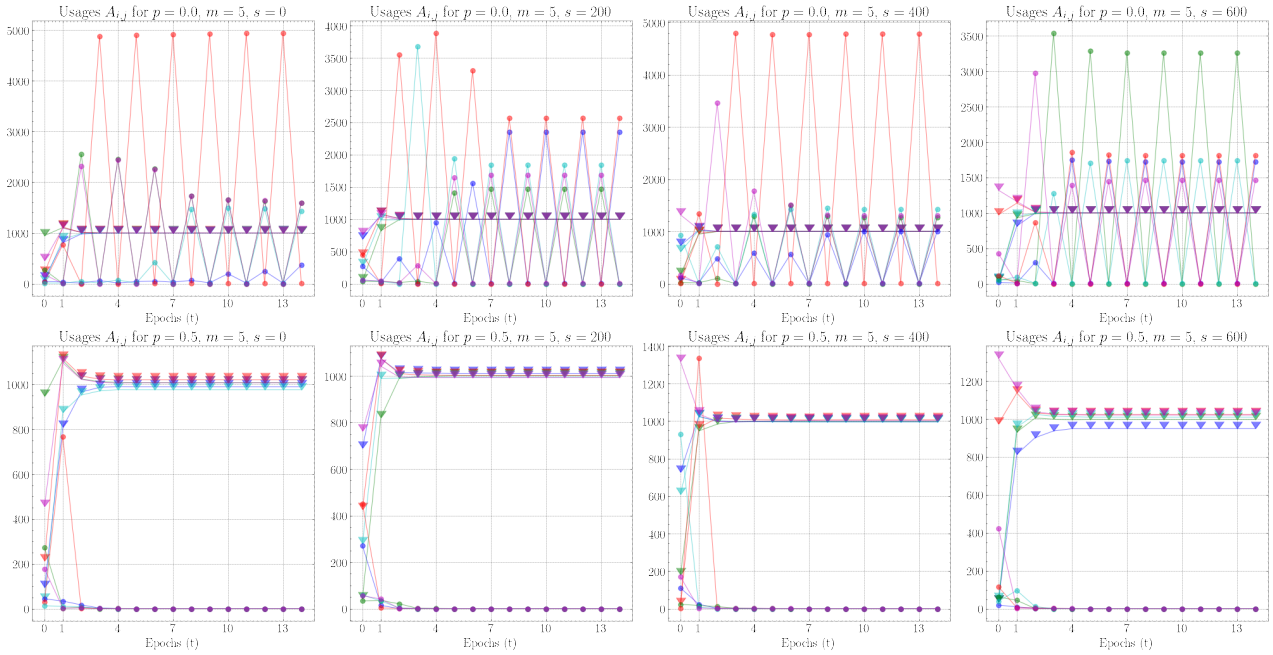

Figure 9: Bank Account Fraud dataset; ablating on the seed (s). The top four graphs give the $p = 0$ case while the bottom four give $p = 0.5$. In each graph, triangle markers indicate positive usages while circular markers indicate negative usages; colors indicate the different services.