Stateful Performative Gradient Descent

Anonymous Authors[1]

Abstract

A recent line of work has focused on training machine learning (ML) models in the performative setting, i.e. when the data distribution reacts to the deployed model. The goal in this setting is to compute a model which both induces a favorable distribution and performs well on the induced distribution, thereby minimizing the test loss. Previous work on finding an optimal model assumes that the data distribution immediately adapts to the deployed model. In practice, however, this may not be the case, as the population may take time to adapt to the model. In this work, we propose an algorithm for minimizing the performative loss even in the presence of these effects.

1. Introduction

A recent line of work initiated by (Perdomo et al., 2020) has sought to study how to effectively train machine learning (ML) models in the presence of performative effects. Performativity describes the scenario in which our deployed model or algorithm effects the distribution of the data or population which we are studying. Such effects can be expected when our model is used to make consequential decisions concerning the population, e.g. if we deploy a classifier for automatic disease diagnosis in a clinical setting. As ML becomes ever more ubiquitous across fields, considering these performative effects also grows in importance.

The goal of model training under performative distribution shift is to minimize the model's loss *on the distribution it induces.* Recently, (Izzo et al., 2021) proposed a "meta-algorithm" (performative gradient descent or PerfGD) to accomplish this when the induced data distribution depends only on the deployed model. This amounts to assuming that the data distribution immediately adapts to the deployed model, irrespective of any other conditions. In practice, such a model of performative effects may be overly simplistic. It is likely that the induced distribution will depend not only on the deployed model, but also some notion of the "state" that the population was in when the model was deployed. This notion was introduced by (Brown et al., 2020), who also gave algorithms for finding locally stable models in this setting. A notion of optimality in this setting is the minimization of the *long-term* performative loss—that is, finding a model which minimizes the average risk over an infinite time horizon, assuming that we keep deploying that same model.

1.1. Our contributions

In this work, we introduce an algorithm for minimizing the performative risk in the stateful setting of (Brown et al., 2020). Our algorithm is similar in spirit to that of (Izzo et al., 2021), in that it amounts to estimating an appropriate gradient and using it to perform gradient descent. However, unlike (Izzo et al., 2021), we no longer have even direct sample access to the distribution that we care about (the "long-term" induced distribution), and this added technical challenge makes previous algorithms for optimizing the performative risk ineffective. Indeed, the only way to apply previous approaches directly is to wait for many time steps after each model deployment so that the induced distribution stabilizes to its long-term limit. Our algorithm overcomes this limitation by "simulating" waiting, without actually needing to do so. We show theoretically that this method accurately captures the behavior of the long-term distribution. Experiments confirm our theory and also show its improvement over existing methods which are not specifically adapted to the stateful setting.

1.2. Related work

(Perdomo et al., 2020) introduced the performative prediction framework to the ML community. They gave two simple algorithms—repeated risk minimization (RRM) and repeated gradient descent (RGD)—which converge to a stable point.

(Brown et al., 2020) introduced the "stateful" extension of performative distribution shift and showed that the RRM and RGD procedures introduced by (Perdomo et al., 2020) converge to long-term stable points.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

(Izzo et al., 2021) gave the first method (PerfGD) for computing the performative optimum. Their method assumes that the induced distribution belongs to a known parameterized family such as a Gaussian with fixed variance, but the dependence of the parameter (in this case the mean) on the model is unknown. They showed that estimating the gradient of the performative loss reduces to estimating the derivative of this parameter with respect to the deployed model, then used this to do (approximate) gradient descent on the performative loss. Under convexity assumptions on the performative loss, they showed that PerfGD converges to an approximate minimizer.

(Miller et al., 2021) also studied optimizing the (stateless) performative loss. The authors quantified when the performative loss is convex and propose using black-box derivative-free optimization methods to find the performative optimum.

A related line of work studies the setting of strategic classification (Hardt et al., 2015), which is a subclass of the general performative setting. In this setting, it is assumed that individual datum react to a deployed model by a best-response mechanism, inducing a population-level distribution shift. (Dong et al., 2017) considered optimizing the performative risk in an online version of this problem and for a certain class of best-response dynamics.

Lastly, the original performative optimization problem can be thought of as a derivative-free optimization (DFO) (Flaxman et al., 2005) problem with a noisy function value oracle. In the stateful case, however, we no longer even have an unbiased noisy oracle for the function we wish to optimize (the long-term performative risk), making black-box DFO algorithms ineffective.

2. Problem setup and notation

We consider a generalization of the original performative prediction problem (Perdomo et al., 2020). Rather than having the distribution map $\mathcal{D}$ depend only on our model parameters θ, we instead let the distribution observed at time t (denoted ρ_t) depend on both our deployed parameters θ_t and the previous distribution ρ_{t-1}:

$$\rho_t = \mathcal{D}(\theta_t, \rho_{t-1}).$$

Note that this setting is strictly more general than the original setup, and captures the fact that in practice, it is unlikely that the population we are modeling will immediately snap to a new distribution upon deployment of a new model. In general, it will take the distribution some time to adapt. This generalization was introduced by (Brown et al., 2020), who referred to it as "stateful" performativity.

Under reasonable regularity conditions on $\mathcal{D}$, if we define $\theta_t \equiv \theta$ for all t, then there exists a limiting distribution $\mathcal{D}^*(\theta) = \lim_{t\to\infty} \rho_t$. (See Claim 1 of (Brown et al., 2020) for sufficient conditions.) That is, $\mathcal{D}^*(\theta)$ describes the limiting distribution if we continue to deploy θ for all time steps t. If we define the long-term performative loss $\mathcal{L}^*(\theta) = \mathbb{E}_{\mathcal{D}^*(\theta)}[\ell(z;\theta)]$, then a sensible goal is to compute the long-term optimum

$$\theta^*_{\mathrm{OPT}} \triangleq \operatorname*{argmin}_{\theta\in\Theta} \mathcal{L}^*(\theta),$$

where Θ is some set of admissible model parameters. This is similar to the problem addressed in (Izzo et al., 2021), except now we do not even have direct sample access to $\mathcal{D}^*$.

For simplicity, we will restrict our attention to the case where ρ_t is a Gaussian with fixed covariance Σ. (We remark that our techniques should be viewed more as a "meta-algorithm" whose details can easily be generalized to other parametric distributions.) In this setting, ρ_t is completely determined by its mean $\mu_t \in \mathbb{R}^d$, and rather than the distribution map $\mathcal{D}$, we can equivalently consider the mean map m, where

$$\mu_t = m(\theta_t, \mu_{t-1}).$$

We then have $\rho_t = \mathcal{N}(m(\theta_t, \mu_{t-1}), \Sigma)$. Analogously to the long-term distribution assumption, we will assume that for every fixed θ and any starting μ, there is a long-term mean $\mu^*(\theta) = \lim_{k\to\infty} m^k(\theta, \mu)$, where $m^0(\theta,\mu) = \mu$ and $m^k(\theta,\mu) = m(\theta, m^{k-1}(\theta,\mu))$ for $k \geq 1$.

We define the *decoupled performative loss* $L(\theta,\mu) \triangleq \mathbb{E}_{\mathcal{N}(\mu,\Sigma)}[\ell(z;\theta)]$. Given a fixed value of μ_{t-1}, the performative loss when we deploy θ_t at time t can be written as

$$L_t = L(\theta_t, m(\theta_t, \mu_{t-1})).$$

Lastly, we will use $\nabla_i f$ to denote the gradient of a function f with respect to its i-th argument. So for instance, $\nabla_1 \mathcal{L}(\theta; m(\theta,\mu_t))$ means the gradient of $\mathcal{L}(\theta; m(\theta,\mu_t))$ *only with respect to the θ appearing in the first argument* (before the semicolon) even though θ appears in $\mathcal{L}$ in the second argument as well.

2.1. Previous algorithms and their limitations

RRM and RGD (Brown et al., 2020) showed that RRM converges to a stable point in the long run. In general, since the stateless performative problem is a subclass of the stateful one, we can find examples where a stable point can be arbitrarily far from an optimal point. (See §2.2 of (Izzo et al., 2021).)

Naive PerfGD If we use the one-step mean $m(\theta,\mu)$ as a surrogate for the long-term mean, then we can naively apply PerfGD from (Izzo et al., 2021). That is, if we define $\mathcal{L}(\theta;\mu) = L(\theta, m(\theta,\mu))$, then we apply the update

$$\theta_{t+1} = \theta_t - \eta \nabla_1 \mathcal{L}(\theta_t; \mu_t).$$

We will see a toy example in the next section where this method provably fails to converge to the long-term optimum. Intuitively, this is because naive PerfGD only takes into account the short-term impact of θ. It does not account for the impact that θ_t will have on the loss moving from time $t+1$ to $t+2$ through μ_t.

Black-box derivative-free optimization (DFO) Black-box DFO seeks to optimize a function given only a function value oracle and no direct access to e.g. gradients or higher-order derivatives of the function to be optimized (Flaxman et al., 2005). The non-stateful performative prediction setting is indeed a special case of this general problem, and black-box DFO algorithms can obtain reasonable results for non-stateful performative prediction in some cases (Miller et al., 2021). However, in the stateful performative setting, we no longer even have an exact function value oracle for the long-term performative loss, so we expect black-box DFO methods to have degraded performance (if they work at all).

3. Our algorithm: Stateful PerfGD

Our algorithm takes the same approach as the original PerfGD of (Izzo et al., 2021). That is, we estimate the (long-term) performative gradient and then use this estimate to do approximate gradient descent. As we no longer have direct sample access to the distribution we care about (the long-term distribution), the steps needed to estimate the long-term performative gradient are different from (Izzo et al., 2021), and the error analysis is more involved. Below we give the precise steps for computing our estimate.

3.1. Algorithm description

At time t, we propose the estimate

$$\widehat{\nabla \mathcal{L}_t^*} = \int \nabla_\theta \ell(z;\theta_t)\, p(z;\mu_t)\, dz + \int \ell(z;\theta_t)\, \widehat{\frac{dm^k}{d\theta}}^\top \nabla_\mu p(z;\mu_t)\, dz, \tag{1}$$

$$\widehat{\frac{dm^k}{d\theta}} = \widehat{\partial_1 m} \cdot \frac{1-(\widehat{\partial_2 m})^k}{1-\widehat{\partial_2 m}} \tag{2}$$

where $p(\cdot;\mu)$ is the density for a Gaussian random variable with mean μ and variance Σ, and $\widehat{\partial_i m}$ are estimates for $\partial_i m(\theta_t, \mu_t)$ obtained via finite difference approximations (Algorithm 1). See Appendix A for derivations of these quantities and of Algorithm 1.

4. Approximation guarantees

In the simple case where $m(\theta,\mu) = \delta\mu^*(\theta) + (1-\delta)\mu$, we can bound the error of our approximation for $d\mu^*/d\theta$.

Algorithm 1 Estimating $\partial_i m$

Require: Estimation horizon H
$\varphi_s \leftarrow [\theta_s^\top,\ \mu_{s-1}^\top]^\top$ for all $s \le t$
$\Delta\mu \leftarrow [\mu_{t-1} - \mu_t,\ \ldots,\ \mu_{t-H} - \mu_t]$
$\Delta\varphi \leftarrow [\varphi_{t-1} - \varphi_t,\ \ldots,\ \varphi_{t-H} - \varphi_t]$
$\widehat{\nabla m}^\top \leftarrow (\Delta\mu)(\Delta\varphi)^\dagger$
return $\widehat{\nabla m} = [\widehat{\partial_1 m},\ \widehat{\partial_2 m}]$

In this case the true $\partial_i m$ do not depend on μ, so we have $\widehat{\frac{dm^k}{d\theta}} = \frac{dm^k}{d\theta}$ and by taking k arbitrarily large, we get an arbitrarily good approximation for $dm^k/d\theta$ with oracles for $\partial_i m$. Any errors in estimating $\widehat{\partial_i m}$ due to finite difference approximations do not change this result.

Proposition 1. *Suppose that we can estimate the derivatives $\partial_i m$ to error ε: $|\widehat{\partial_i m} - \partial_i m| \le \varepsilon$. Further suppose that μ^* has bounded derivative $|d\mu^*/d\theta| \le M$. Let $\widehat{\frac{dm^k}{d\theta}}$ be given by Eq. (2). Then error $E_k = \left|\widehat{\frac{dm^k}{d\theta}} - \frac{d\mu^*}{d\theta}\right|$ is bounded by*

$$E_k = \mathcal{O}\left(\left[\frac{M}{\delta^2} - \left\{\frac{k(1-\delta)^{k-1}}{\delta}\right\}\right]\varepsilon + M(1-\delta)^k\right).$$

In particular, for $0 < \delta < 1$, as $k \to \infty$, we have $E_k \lesssim M\varepsilon/\delta^2 \to 0$ as our estimates $\widehat{\partial_i m}$ become exact ($\varepsilon \to 0$).

5. Examples and experiments

In this section, we analytically classify the shortcomings of the naive PerfGD method for the stateful setting. We follow up with experiments for all of the relevant methods, showing stateful PerfGD's improvements over existing algorithms. Throughout, the point loss is $\ell(z;\theta) = -z\theta$, so $L(\theta,\mu) = -\mu\theta$. We take the parameter space to be $\Theta = [-5,5]$. For details on the specific constants and hyperparameters used in the experiments, refer to Appendix D.

5.1. Example 1: Linear m

First, we take the mean update function to be

$$m(\theta,\mu) = \delta\mu^*(\theta) + (1-\delta)\mu, \qquad \mu^*(\theta) = a_0 + a_1\theta$$

for some fixed $\delta \in (0,1)$. We can exactly classify the long-term optimal θ as well as the point to which PerfGD converges. In the following two propositions, we assume that the stated parameter values lie in Θ.

Proposition 2. *When $a_1 < 0$, the long-term optimal point is $\theta_{\mathrm{OPT}} = -a_0/2a_1$.*

Proposition 3. *If naive PerfGD converges, then it converges to $\theta = -a_0/(1+\delta)a_1$.*

Note that when $a_0 \neq 0$, the two values in the above propositions are equal iff $\delta = 1$, which is indeed the stateless

performative case. The upshot is that naive PerfGD is too myopic. It optimizes a "short-term" performative loss, and fails to reach the long-term optimum as a result.

Figure 1 below compares the performance of gradient descent with approximation (1) (labeled Stateful PerfGD in the plot) with several other algorithms. RGD refers to the algorithm of (Perdomo et al., 2020) for finding performatively stable points. FLX refers to the black-box DFO algorithm of (Flaxman et al., 2005); "internal" refers to the internal estimate of the algorithm, rather than the actual (perturbed) query points. Naive PerfGD refers to the procedure discussed in Section 2.1.

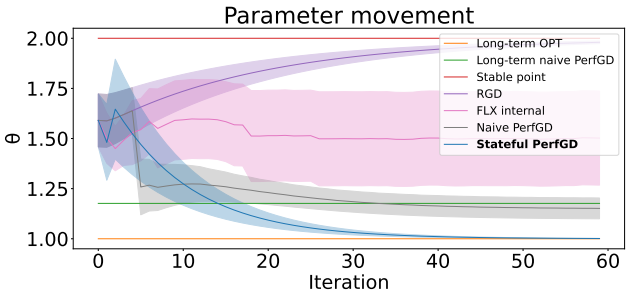

Figure 1. Performance of various algorithms with linear m. The shaded region shows standard error of the mean over 50 trials. The goal is to converge to the long-term optimum θ_{OPT} (orange line). By explicitly taking into account stateful performative effects, our algorithm is the only method which reaches θ_{OPT}.

5.2. Example 2: Non-linear m

To test the method on a more challenging problem and see the effects of the approximation $\widehat{\frac{dm^k}{d\theta}}$, we alter the first toy example so that the rate of convergence to the long-term mean depends on the current mean. In particular, we take

$$m(\theta, \mu) = \delta^{|\mu|}\mu^*(\theta) + (1 - \delta^{|\mu|})\mu,$$

with $\mu^*(\theta) = a_1\theta + a_0$ as before. In this case, the long-term performative loss and optimal point are the same as before, since we still have $m^k(\theta, \mu) \to \mu^*(\theta)$. However, $\partial_i m$ are no longer constants. We have

$$\partial_1 m(\theta, \mu) = \delta^{|\mu|}a_1,$$
$$\partial_2 m(\theta, \mu) = \mathrm{sgn}(\mu)(\ln \delta)\delta^{|\mu|}[\mu^*(\theta) - \mu] + (1 - \delta^{|\mu|}).$$

The approximation of $dm^k/d\theta$ from (2) is no longer exact even if we had oracles for $\partial_i m$. Nevertheless, we see that the approximation (1) is good enough to reach θ_{OPT}.

5.3. Discussion

The results in both the simple linear and more complicated nonlinear cases are similar. RGD converges to a performatively stable point as expected, which in this case is far from

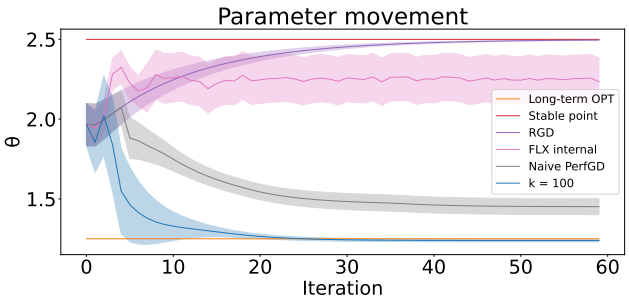

Figure 2. Performance of various algorithms with nonlinear m. The shaded region shows standard error of the mean over 50 trials. Even in the more challenging case with non-constant $\partial_i m$, stateful PerfGD is able to converge to θ_{OPT}.

θ_{OPT}. Flaxman et al.'s algorithm also performs poorly as it assumes access to an exact function value oracle rather than the "biased" oracle we have in the stateful case. Naive PerfGD gets somewhat closer to θ_{OPT} but cannot converge completely due to its myopic nature. Because it explicitly takes into account stateful effects, the approximation (1) is able to converge to θ_{OPT}.

6. Conclusion

We considered the stateful performative setting and showed how to optimize the long-term performative risk under parametric assumptions on the data. We give a theoretical bound on the error of our approximation for relevant derivatives, and we verify empirically that our method is able to overcome the more complicated stateful performative dynamics and find θ_{OPT}, whereas existing methods not tailored to this situation fail.

There are a number of interesting directions for future work. The most obvious is extending the guarantees of our algorithm to more general classes of parametric dynamics. A related goal is to modify our method to work when the stateful performative distribution is a mixture of the long-term and current *distribution*, rather than just a mixture of means. Simply applying the current meta-algorithm on mixtures of parametric distributions is a possibility, but empirically we have found it difficult to estimate the required parameters (e.g. individual means and mixture weights) to sufficient accuracy for this to be effective. In addition, while minimizing the long-term performative risk is a sensible goal, other goals can also be considered—for instance, we can attempt to minimize the regret over the whole time horizon. Lastly, our current method works in the batch setting where we have enough samples to accurately estimate population-level quantities. Developing methods that can work in a stochastic/limited sample regime is also of interest.

References

Brown, G., Hod, S., and Kalemaj, I. Performative prediction in a stateful world, 2020. ISSN 23318422.

Dong, J., Roth, A., Schutzman, Z., Waggoner, B., and Wu, Z. S. Strategic classification from revealed preferences, 2017.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: Gradient descent without a gradient. *SODA*, 2005.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. Strategic classification, 2015.

Izzo, Z., Ying, L., and Zou, J. How to Learn when Data Reacts to Your Model: Performative Gradient Descent. Technical report, 2021.

Miller, J., Perdomo, J. C., and Zrnic, T. Outside the Echo Chamber: Optimizing the Performative Risk. Technical report, 2021.

Perdomo, J. C., Zrnic, T., Mendler-Dünner, C., and Hardt, M. Performative Prediction. *arXiv:2002.06673 [cs, stat]*, apr 2020. ISSN 23318422. URL http://arxiv.org/abs/2002.06673.

A. Derivation of stateful PerfGD

The long-term performative loss is given by

$$\mathcal{L}^*(\theta_t) = \int \ell(z;\theta_t)p(z;\mu^*(\theta_t))\,dz.$$

Its gradient is therefore given by

$$\nabla\mathcal{L}^*(\theta_t) = \int \nabla_\theta \ell(z;\theta_t)\, p(z;\mu^*(\theta_t))\,dz + \int \ell(z;\theta_t)\,\frac{d\mu^*}{d\theta}^\top \nabla_\mu p(z;\mu^*(\theta_t))\,dz.$$

The general form of our gradient estimate (1) arises by substituting μ_t for $\mu^*(\theta_t)$ and $\widehat{\frac{dm^k}{d\theta}}$ for $\frac{d\mu^*}{d\theta}$.

The derivation for Algorithm 1 is as follows. For each time t, let $\varphi_t = [\theta_t^\top,\ \mu_{t-1}^\top]^\top$, and define $m(\varphi_t) = m(\theta_t, \mu_{t-1}) = \mu_t$. By Taylor's theorem, we have

$$m(\varphi_t) - m(\varphi_s) \approx \nabla m(\varphi_s)^\top(\varphi_t - \varphi_s). \tag{3}$$

then we can vectorize equation (3) and obtain

$$\Delta\mu \approx \nabla m(\varphi_t)^\top \Delta\varphi \quad\Longrightarrow\quad \nabla m(\varphi_t)^\top \approx (\Delta\mu)(\Delta\varphi)^\dagger.$$

The expression (2) for $\widehat{\frac{dm^k}{d\theta}}$ arises as follows. Observe that

$$\begin{aligned}\frac{d}{d\theta}m^k(\theta,\mu) &= \frac{d}{d\theta}[m(\theta, m^{k-1}(\theta,\mu))] \\ &= \partial_1 m(\theta, m^{k-1}(\theta,\mu)) \\ &\quad + \partial_2 m(\theta, m^{k-1}(\theta,\mu)) \cdot \frac{d}{d\theta}m^{k-1}(\theta,\mu).\end{aligned} \tag{4}$$

Here if $m(\cdot,\cdot)$, $\mu \in \mathbb{R}^d$ and $\theta \in \mathbb{R}^p$, then $dm^k/d\theta$, $\partial_1 m \in \mathbb{R}^{d\times p}$ and $\partial_2 m \in \mathbb{R}^{d\times d}$. Note that if we expand this formula, $dm^k/d\theta$ depends only on the first-order derivatives $\partial_1 m$ and $\partial_2 m$ evaluated at θ and $m^i(\theta,\mu)$ for $i < k$. In general, we cannot know $m^i(\theta,\mu)$ for $i > 1$ except by deploying θ for several steps. If the mean changes slowly, however, then we can just substitute μ for each of these quantities. That is, we assume that $\frac{dm^k}{d\theta} \approx \partial_1 m(\theta,\mu) + \partial_2 m(\theta,\mu)\cdot\frac{dm^{k-1}}{d\theta}$.

Because we do not change the points at which we are evaluating $\partial_i m$ during each iteration of our approximation, the final output $\widehat{\frac{dm^k}{d\theta}}$ actually has the closed form (2). This formula can easily be shown via induction.

We now find ourselves in one of two scenarios. If the mean does indeed adapt slowly, then the approximation $m^i(\theta,\mu) \approx \mu$ will be good for small values of i, and we can thus get a good estimate of $\frac{dm^k}{d\theta}$. On the other hand, if the mean adapts quickly, then we can actually deploy θ for a few steps to get a good estimate for $\mu^*(\theta)$ and then proceed with the usual PerfGD.

B. Proofs for §4

Proof of Proposition 1. We start by decomposing

$$E_k \le \underbrace{\left|\widehat{\frac{dm^k}{d\theta}} - \frac{dm^k}{d\theta}\right|}_{E_k^1} + \underbrace{\left|\frac{dm^k}{d\theta} - \frac{d\mu^*}{d\theta}\right|}_{E_k^2}$$

and proceed by bounding each E_k^i separately.

We first bound E_k^2. For notational convenience, let $\bar\delta = 1-\delta$. It can be shown via induction that $m^k(\theta,\mu) = (1-\bar\delta^k)\mu^*(\theta) + \bar\delta^k\mu$. It follows that

$$E_k^2 = \left|(1-\bar\delta^k)\frac{d\mu^*}{d\theta} - \frac{d\mu^*}{d\theta}\right| \le M\bar\delta^k.$$

To bound E_k^1, we first observe that if $\widetilde{\frac{dm^k}{d\theta}}$ is given by (2) with the *true* partials $\partial_i m$ used, then $\widehat{\frac{dm^k}{d\theta}} = \frac{dm^k}{d\theta}$. (This is due to the fact that m is linear in μ and will not be true in general.) We therefore have $E_k^1 = \left|\widehat{\frac{dm^k}{d\theta}} - \widetilde{\frac{dm^k}{d\theta}}\right|$. Some

simple algebra using the recursive formula (4) for $\widehat{\frac{dm^k}{d\theta}}$ and $\widetilde{\frac{dm^k}{d\theta}}$ yields

$$E_k^1 = \mathcal{O}\left(\left[\frac{M}{\delta^2} + \frac{1}{\delta} - \left\{\frac{k(1-\delta)^{k-1} + (1-\delta)^k}{\delta}\right\}\right]\varepsilon\right).$$

Combining this with the bound on E_k^2 yields the desired result. □

C. Proofs for §5

Proof of Proposition 2. It is easy to check that in this case the corresponding long-term mean is $\mu^*(\theta) = a_0 + a_1\theta$, which implies that $\mathcal{L}^*(\theta) = -\theta(a_0 + a_1\theta)$. Assuming that $a_1 < 0$, we have $\theta_{\mathrm{OPT}} = -a_0/2a_1$. Note that the long-term mean in this case is $\mu_{\mathrm{OPT}} = a_0/2$. □

Proof of Proposition 3. First, since $\mathcal{L}(\theta;\mu) = -\theta(\delta a_1\theta + \delta a_0 + (1-\delta)\mu)$, we have

$$\nabla_1\mathcal{L}(\theta;\mu) = -(2\delta a_1\theta + \delta a_0 + (1-\delta)\mu).$$

If naive PerfGD converges, then we have the system of equations

$$\nabla_1\mathcal{L}(\theta;\mu) = -(2\delta a_1\theta + \delta a_0 + (1-\delta)\mu) = 0$$

$$m(\theta,\mu) = \delta(a_0 + a_1\theta) + (1-\delta)\mu = \mu.$$

Solving for μ and θ yields $\mu = \delta a_0/(1+\delta)$ and $\theta = -a_0/(1+\delta)a_1$, as desired. We remark that this is equal to θ_{OPT} if $\delta = 1$, in which case we are back in the original "non-stateful" performative setting. □

D. Experiment details

We use the population quantity $m(\theta,\mu)$ in our algorithms rather than estimating it as a sample mean. We can always make our sample large enough so that this is not a problem, and we leave analysis of sample complexity for future work.

In both experiments, all of the algorithms used a learning rate of 0.1 for $T = 60$ steps. We set $k = 100$ in the approximation (1) used for stateful PerfGD. Both stateful and naive PerfGD used the entire history to estimate $\partial_i m$ via finite differences. The query perturbation size for FLX was 0.1.

For the linear m experiment (§5.1), we set $a_0 = 1$, $a_1 = -0.5$, and $\delta = 0.7$.

For the nonlinear m experiment (§5.2), we set $a_0 = 2$, $a_1 = -0.8$, and $\delta = 0.7$.