

Dynamic Regret Convergence Analysis and an Adaptive Regularization Algorithm for On-Policy Robot Imitation Learning

Jonathan N. Lee^{*1}, Michael Laskey², Ajay Kumar Tanwani¹,
Anil Aswani¹ and Ken Goldberg¹

¹University of California, Berkeley

²Toyota Research Institute

August 12, 2019

Abstract

On-policy imitation learning algorithms such as DAgger evolve a robot control policy by executing it, measuring performance (loss), obtaining corrective feedback from a supervisor, and generating the next policy. As the loss between iterations can vary unpredictably, a fundamental question is under what conditions this process will eventually achieve a converged policy. If one assumes the underlying trajectory distribution is static (stationary), it is possible to prove convergence for DAgger. However, in more realistic models for robotics, the underlying trajectory distribution is dynamic because it is a function of the policy. Recent results show it is possible to prove convergence of DAgger when a regularity condition on the rate of change of the trajectory distributions is satisfied. In this article, we reframe this result using dynamic regret theory from the field of online optimization and show that dynamic regret can be applied to any on-policy algorithm to analyze its convergence and optimality. These results inspire a new algorithm, Adaptive On-Policy Regularization (AOR), that ensures the conditions for convergence. We present simulation results with cart-pole balancing and locomotion benchmarks that suggest AOR can significantly decrease dynamic regret and chattering as the robot learns. To our knowledge, this the first application of dynamic regret theory to imitation learning.

1 Introduction

There has been great interest in learning-based methods to enable robots to accomplish difficult tasks autonomously. One of the foremost methods is learning by demonstration, also known as imitation learning. In imitation learning, a robot observes states and control labels from a supervisor and estimates a mapping from states to controls. A fundamental problem in imitation learning is covariate shift (Bagnell, 2015), where the distribution of trajectories experienced by the robot at run time differs from the distributions experienced during training time. For example, consider an autonomous vehicle trained to drive on a road from demonstrations of humans driving safely on the center of the same road. If the vehicle makes slight errors when it is deployed, it may drift

^{*}jonathan.lee@berkeley.edu

towards the sides of the road where it had not previously experienced data from human supervisors, resulting in situations from which it cannot recover.

On-policy Imitation learning algorithms such as DAgger (Ross et al., 2011), AggreVaTeD (Sun et al., 2017), LOKI (Cheng et al., 2018), and SIMILE (Le et al., 2016) have been proposed to mitigate this issue. As opposed to learning only from supervisor demonstrations, these algorithms roll out the robot’s current policy at each iteration, allowing it to make errors and observe new states. The supervisor then provides corrective control labels for these new states retroactively. For this reason, these algorithms are often referred to as on-policy imitation learning algorithms because the robot iteratively learns from its current policy (Osa et al., 2018) much like an on-policy reinforcement learning algorithm. This is in contrast to off-policy algorithms where the robot learns from passively observing the supervisor’s demonstrations.

On-policy algorithms for imitation learning have been extremely successful in robot learning applications (Duan et al., 2017; Duvallet et al., 2013; Pan et al., 2018; Ross et al., 2013; Zhang and Cho, 2017). However, the theoretical understanding of these algorithms is limited. For a given algorithm, we might ask the following fundamental questions: Does it return an “optimal” solution? Does it provably converge to a policy? If so, what are the properties of this converged policy? The original work of Ross et al. (2011) on DAgger sought to bound the worst-case average loss of a policy returned from the algorithm. However, the aforementioned questions remained open. Recently, the work of Cheng and Boots (2018) proved convergence for the case of DAgger under regularity conditions on the dynamics. In this work, we aim to extend these prior results by analyzing a general formulation of on-policy algorithms in an effort to answer the above questions directly. We examine these algorithms through the lens of modern results in dynamic regret theory from the field of online optimization (Zinkevich, 2003).

On-policy algorithms can be viewed as derivatives of algorithms from online optimization (Hazan, 2016; Ross et al., 2011), a rich theoretical field often used to analyze problems such as portfolio management and network routing analysis where environments change over time (Hazan and Seshadhri, 2007). The general formulation is as follows: at iteration n , a learner makes a decision θ_n from a set of decisions Θ and then the environment presents some loss function $f_n : \Theta \mapsto \mathbb{R}$. The learner aims to minimize the incurred losses $f_n(\theta_n)$ over time. In on-policy imitation learning, θ_n would correspond to the parameters of the robot’s policy. The loss f_n would be a supervised learning loss function obtained from rolling out θ_n and observing corrective labels from the supervisor. At each iteration the supervised learning loss function is different because the distribution of trajectories induced by the robot changes as the policy is updated. The goal is to continually try to find the optimal θ_n at each iteration based on the sequence of past loss functions. A common choice in online optimization to measure the performance of an algorithm is static regret over N iterations, defined as

$$R_S(\theta_1, \dots, \theta_N) \stackrel{\text{def.}}{=} \sum_{n=1}^N f_n(\theta_n) - \min_{\theta} \sum_{n=1}^N f_n(\theta). \quad (1)$$

In imitation learning, this would mean the robot is compared against the best it could have done on the average of its trajectory distributions seen in the past.

In this article, we focus on determining when on-policy algorithms can and cannot converge and specifically when the policy is performing optimally on its own distribution. We draw a connection between this very natural objective and an alternative metric known as *dynamic regret*, which has recently gained significant traction in the online optimization community (Hall and Willett, 2015;

Mokhtari et al., 2016; Yang et al., 2016). As opposed to the well known static regret, dynamic regret measures performance of a policy at each instantaneous iteration:

$$R_D(\theta_1, \dots, \theta_N) \stackrel{\text{def.}}{=} \sum_{n=1}^N f_n(\theta_n) - \sum_{n=1}^N \min_{\theta} f_n(\theta). \quad (2)$$

The difference between static and dynamic regret is that, for dynamic regret, the minimum resides inside the summation, meaning that the regret is an instantaneous difference at each iteration. Proving that static regret is low implies that the policies are at least as good as a single fixed policy that does well on the average of the distributions seen during training. This was precisely the original result of Ross et al. (2011). In contrast, proving that dynamic regret is low implies that the policies are optimal on their own distribution. Consequently, dynamic regret can be more relevant as a theoretical metric in robotics where the trajectory distributions are changing, and understanding the dynamic regret of an algorithm positions us to answer the aforementioned questions about on-policy imitation learning.

As an example of the difference in practice, consider again the autonomous driving scenario using an on-policy algorithm where the car will encounter both challenging safety-critical situations, such as avoiding collisions, and relatively easy situations, such as lane-following. Low static regret implies that the car will do well in many of these situations on average. However, this could mean the policy is good at lane-following but highly suboptimal at avoiding collisions even if it has to encounter both situations at run time. However, low dynamic regret offers a much safer guarantee by ensuring that the car will be optimal (among its policy class) with respect to the situations it actually encounters at run time.

However, showing an algorithm achieves low dynamic regret is inherently harder than achieving low static regret because $R_S \leq R_D$. Furthermore, it is well known that it is not possible to guarantee low dynamic regret in general due to the possibility of adversarial loss functions (Yang et al., 2016). The key insight in imitation learning, however, is that the loss functions at each iteration represent the trajectory distributions as a function of the policy parameters. Therefore, we can leverage information known about how the trajectory distribution changes in response to changing policy parameters to obtain interpretable dynamic regret rates. In particular, we rely on the continuity of the trajectories as a function of the policy (Cheng and Boots, 2018).

This paper makes four contributions:

1. Introduces a novel dynamic regret analysis to evaluate the convergence of on-policy imitation learning algorithms.
2. Presents average dynamic regret rates and conditions for convergence for DAgger, Imitation Gradient, and Multiple Imitation Gradient.
3. Introduces Adaptive On-Policy Regularization (AOR), a novel algorithm that adaptively regularizes on-policy algorithms to improve dynamic regret and induce convergence.
4. Presents empirical evidence of non-convergent on-policy algorithms and shows that AOR can ensure convergence in a cart-pole balancing task and locomotion tasks.

This article is a significantly revised and extended version of our conference publication at the Workshop on the Algorithmic Foundations of Robotics (Lee et al., 2018a,b). In particular, this article (1) presents new theoretical results, greatly extending the formalization of dynamic

regret as a metric in imitation learning; (2) provides detailed examples and analysis of well known systems that satisfy the continuity condition required in the theory; (3) explores connections with our subsequent work in Continuous Online Learning (Cheng et al., 2019a) and the variational inequality problem; (4) presents new experimental results.

2 Related Work

The challenge of covariate shift in imitation learning by supervised learning is the subject of significant research in robotics. It is especially prevalent when the robot’s policy cannot fully represent the supervisor (Laskey et al., 2017). In robotics, many algorithms have been proposed to mitigate covariate shift for imitation learning. Ross et al. (2011) introduced DAgger, an on-policy algorithm that allows the robot to make mistakes and then observe corrective labels from the supervisor in new states that might not be seen from ideal supervisor demonstrations alone.

Gradient-based on-policy methods for imitation learning have gained interest due to their similarity to policy gradient algorithms and their computational efficiency. These are also known as Imitation Gradient methods. AggreVaTeD (Sun et al., 2017) was proposed for fast policy updates designed for deep neural network policies. LOKI (Cheng et al., 2018) uses a mirror descent algorithm on an imitation learning loss to bootstrap reinforcement learning.

Ross et al. (2011) first introduced a static regret-style analysis for DAgger, showing that with strongly convex losses, running DAgger results in low static regret in all cases. This means that a DAgger policy is on average at least as good as one policy that does well on the average of trajectory distributions seen during training. However, the performance on the average of trajectory distributions is not always informative, as shown by Laskey et al. (2017), because the average may contain irrelevant distributions as a result of rolling out suboptimal policies. Cheng and Boots (2018) recently expanded the DAgger analysis proving that, despite guaranteed convergence in static regret, the algorithm may not always converge to a low loss policy. Furthermore, they identified regularity conditions sufficient for convergence to local optima. This work extends the results of Cheng and Boots (2018) by drawing a connection with dynamic regret theory to identify conditions for convergence for DAgger and other on-policy algorithms and as a basis for a new algorithm.

In Cheng et al. (2019a) (subsequent to Lee et al. (2018a)), we presented a notion of “continuous online learning” where we studied the dynamic regret of online optimization problems with smoothly changing losses. In that work, we generalized the structure observed in the imitation learning problem to prove low dynamic regret more generally. We discuss connections to the continuous online learning problem in Section 6.5.

3 Preliminaries

In this section, we review mathematical background and notation. We then introduce the problem statement for imitation learning by supervised learning.

3.1 Mathematical Background

Let $\Theta \subset \mathbb{R}^d$ be a convex and compact set with l_2 -norm $\|\cdot\|$. A differentiable function $h : \Theta \mapsto \mathbb{R}$ is said to be convex if $\forall \theta, \theta' \in \Theta$,

$$h(\theta) \geq h(\theta') + \langle \nabla h(\theta'), \theta - \theta' \rangle.$$

It is strictly convex if the inequality is strict for all $\theta \neq \theta'$. It is α -strongly convex if there exists $\alpha > 0$ such that

$$h(\theta) \geq h(\theta') + \langle \nabla h(\theta'), \theta - \theta' \rangle + \frac{\alpha}{2} \|\theta - \theta'\|^2.$$

Strongly convex functions are strictly convex, and strictly convex functions are convex. The function h is said to be γ -smooth if its gradients are γ -Lipschitz continuous:

$$\|\nabla h(\theta) - \nabla h(\theta')\| \leq \gamma \|\theta - \theta'\|.$$

An equivalent definition of γ -smoothness is that for all $\theta, \theta' \in \Theta$,

$$h(\theta) \leq h(\theta') + \langle \nabla h(\theta'), \theta - \theta' \rangle + \frac{\gamma}{2} \|\theta - \theta'\|^2,$$

which parallels the definition of strong convexity. The Bregman divergence $\mathcal{D}_h : \Theta \times \Theta \mapsto \mathbb{R}_{\geq 0}$ with respect to a strictly convex function h is defined as

$$\mathcal{D}_h(\theta, \theta') = h(\theta) - h(\theta') - \langle \nabla h(\theta'), \theta - \theta' \rangle.$$

3.2 Imitation Learning

Setup. We consider systems represented by a Markov decision process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{U}, p, T \rangle$, where \mathcal{S} is a bounded set of states, \mathcal{U} is a bounded set of allowable controls, $p : \mathcal{S} \times \mathcal{U} \mapsto \mathbb{P}(\mathcal{S})$ is transition function mapping states and actions to a density over states, and T is the time horizon. We do not assume access to a reward or cost function intrinsic to the MDP. In this setting \mathcal{S} and \mathcal{U} can be discrete or continuous.

A policy $\pi : \mathcal{S} \mapsto \mathbb{P}(\mathcal{U})$ from the set of allowable policies Π is a mapping from states to a density over the set of controls. When a policy is deployed on the MDP, such as by rolling out a robot in the real world, it generates a trajectory τ , which is a tuple of states and actions: $\tau = (s_1, u_1, \dots, s_{T-1}, u_{T-1}, s_T)$. Given an initial state distribution μ over \mathcal{S} , the probability density of a trajectory under policy π is given by

$$p(\tau; \pi) = \mu(s_1) \prod_{t=1}^{T-1} \pi(u_t | s_t) p(s_{t+1} | s_t, u_t),$$

When π is deterministic, it is taken to be the Dirac (or Kronecker in the discrete case) delta function. In this paper, we consider learning parametric policies, i.e., there is a convex and compact subset of parameters $\Theta \subset \mathbb{R}^d$ equipped with the l_2 -norm $\|\cdot\|$ that parameterizes policies π_θ for $\theta \in \Theta$. Its diameter is given by $D \stackrel{\text{def.}}{=} \max_{\theta_1, \theta_2 \in \Theta} \|\theta_1 - \theta_2\|$.

Objective. Rather than assuming access to a cost or reward function, we assume that the robot has access to a supervisor policy π^* that is capable of providing control labels to states. Our goal is to find a parameterized policy π_θ that minimizes some loss with respect to the supervisor π^* . The learned policy in general is not able to match π^* . The loss of a policy π_θ along a trajectory τ is a non-negative function such that

$$J_\tau(\pi_\theta) \stackrel{\text{def.}}{=} \frac{1}{T-1} \sum_{t=1}^{T-1} \ell_t(s_t, \pi_\theta),$$

where $\ell_t : \mathcal{S} \times \Theta \mapsto \mathbb{R}_{\geq 0}$ is a differentiable (in θ) per-time step surrogate loss function. The general optimization problem of imitation learning can be written as

$$\min_{\theta} \mathbb{E}_{p(\tau; \pi_\theta)} J_\tau(\pi_\theta) \quad \text{s.t.} \quad \theta \in \Theta$$

where

$$\mathbb{E}_{p(\tau; \pi_\theta)} J_\tau(\pi_\theta) = \int p(\tau; \pi_\theta) J_\tau(\pi_\theta) d\tau$$

The expectation is taken over the distribution of trajectories that π_θ induces. We can interpret this objective in the following way: trajectories are sampled from the distribution induced by π_θ and then the performance of π_θ is evaluated along those trajectories. We wish to find the parameter θ in Θ that gives rise to trajectories that ensure that $J_\tau(\theta)$ is low. This problem reflects the goal of having the policy do well on its own induced distribution. However, this is challenging and cannot be solved efficiently (Sun et al., 2017) since the distribution of trajectories is unknown. It also cannot be exactly solved with regular supervised learning by sampling supervisor trajectories because the sampling distribution is a function of the policy (Bagnell, 2015).

Relaxations. Existing algorithms relax the problem by fixing the trajectory distribution and then optimizing over the evaluation parameter. This decouples the sampling from the supervised learning problem. For example, in standard behavior cloning, one sets the trajectory distribution to the supervisor’s trajectory distribution and finds the policy that minimizes loss on that distribution. Formally, we consider the average loss of a parameter $\theta \in \Theta$ over the distribution of trajectories generated by a possibly different policy parameter $\theta' \in \Theta$:

$$f_{\theta'}(\theta) \stackrel{\text{def.}}{=} \mathbb{E}_{p(\tau; \pi_{\theta'})} J_\tau(\pi_\theta). \tag{3}$$

Here, θ' controls the trajectory distribution and θ controls the predictions used to compute the loss on that distribution. For this reason, we refer to the θ' as the distribution-generating parameter and θ as the evaluation parameter. We can view $f_{\theta'}(\theta)$ as a bifunction (Cheng et al., 2019a).

Optimization problems in this article will be of the form $\min_{\theta \in \Theta} f_{\theta'}(\theta)$ for some fixed and known $\theta' \in \Theta$. Because the trajectory distribution no longer depends on the variable θ , the supervised learning problem can now be feasibly solved by sampling from $p(\tau; \pi_{\theta'})$ which corresponds to rolling out trajectories under the fixed policy $\pi_{\theta'}$. Specifically in this article, we will consider on-policy algorithms which operate over $N \in \mathbb{N}$ iterations. At any iteration n for $1 \leq n \leq N$, the policy parameter θ_n is rolled out as the distribution-generating parameter and the loss $f_{\theta_n}(\theta) = \mathbb{E}_{p(\tau; \pi_{\theta_n})} J_\tau(\pi_\theta)$ is observed, where θ is the free variable. For convenience, we write $f_n(\theta) \stackrel{\text{def.}}{=} f_{\theta_n}(\theta)$. These loss functions form the sequence of losses used in the regret metrics R_S and R_D .

3.3 Assumptions

Finally, we briefly describe the main assumptions of this article. The assumptions are stated formally in Section 6. As in prior work in both imitation learning and online optimization (Hazan, 2016; Ross et al., 2011), we assume strong convexity and smoothness of the loss function in the evaluation parameter $J_\tau(\pi_\theta)$. Intuitively, strong convexity ensures the loss is curved at least quadratically while smoothness guarantees it is not too curved. In practice these assumptions are satisfied when $J_\tau(\pi_\theta)$ is a ridge regression loss, for example.

As in the work of Cheng and Boots (2018), we also assume a regularity condition on the f_n sequence, which bounds the sensitivity of the trajectory distribution in response to changes in the distribution-generating parameter. That is, for a fixed θ , we assume that $\nabla f_{\theta'}(\theta)$ is Lipschitz continuous in θ' . This condition captures the basic continuity and structure of the imitation learning problem. We discuss its motivation in Section 5 and examples in Section 8.

In practice, many works have observed good performance of both off-policy and on-policy algorithms using non-convex losses such as neural networks even though these algorithms were often proposed for convex losses. (Hussein et al., 2018; Laskey et al., 2017; Pan et al., 2018; Pomerleau, 1989; Zhang et al., 2018).

4 On-Policy Algorithms

We now review three on-policy algorithms that will be the focus of the main theoretical and empirical results of the article.

4.1 DAgger

DAgger is a derivative of the follow-the-leader algorithm from Online Optimization. For detailed discussion of implementation, we refer the reader to (Hazan, 2016; Ross et al., 2011). DAgger proceeds by rolling out the current policy and observing a loss based on the induced trajectory distribution. The next policy parameter is computed by aggregating all observed losses and minimizing over them. An example of this for the l_2 -regularized linear regression problem would be $\min_\theta \sum_{m=1}^n f_m(\theta)$, where

$$\sum_{m=1}^n f_m(\theta) = \sum_{m=1}^n \mathbb{E} \|S_m \theta - U_m\|^2 + \frac{\alpha_0}{2} \|\theta\|^2. \tag{4}$$

Here, S_m is the matrix of state vectors observed from rolling out π_m and U_m is the matrix of control labels from the supervisor at the m th iteration. The expectation is taken with respect to the randomness in S_m and U_m . In the original DAgger algorithm, a user-defined stochastic mixing term was included. We omit this term because setting it to zero only improves the original bound by Ross et al. (2011).

4.2 Imitation Gradient

Recently there has been interest in “Imitation Gradient” algorithms. Algorithms such as AggreVaTeD and LOKI fall in this family. The online gradient descent algorithm from online optimization underlies such algorithms and their variants. The aforementioned methods have explored more

Algorithm 1 DAgger (Ross et al., 2011)

Input: Initial policy parameter θ_1 ,
Max iterations N .
for $n = 1$ **to** $N - 1$ **do**
Roll out θ_n and collect τ_n .
Form loss $f_n(\theta) = f_{\theta_n}(\theta)$ from feedback on τ_n .
 $\theta_{n+1} \leftarrow \arg \min_{\theta \in \Theta} \sum_{m=1}^n f_m(\theta)$.
end for

Algorithm 2 Imitation Gradient (Cheng et al., 2018; Sun et al., 2017)

Input: Initial policy parameter θ_1 ,
Max iterations N ,
Stepsize η .
for $n = 1$ **to** $N - 1$ **do**
Roll out θ_n and collect τ_n .
Form loss $f_n(\theta) = f_{\theta_n}(\theta)$ from τ_n .
 $\theta_{n+1} \leftarrow P_{\Theta}(\theta_n - \eta \nabla f_n(\theta_n))$.
end for

Algorithm 3 Multiple Imitation Gradient (Zhang et al., 2017)

Input: Initial policy parameter θ_1 ,
Max iterations N ,
Updates per iteration K ,
Stepsize η .
for $n = 1$ **to** $N - 1$ **do**
Roll out θ_n and collect τ_n .
Form loss $f_n(\theta) = f_{\theta_n}(\theta)$ from τ_n .
 $\theta_n^1 \leftarrow \theta_n$
for $k = 1$ **to** K **do**
 $\theta_n^{k+1} \leftarrow P_{\Theta}(\theta_n^k - \eta \nabla f_n(\theta_n^k))$.
end for
 $\theta_{n+1} \leftarrow \theta_n^{K+1}$.
end for

Top: DAgger minimizes over all observed loss functions which are represented by a supervised learning loss over all observed data. Middle: Imitation Gradient computes a gradient on data collected from only the most recent rollout and takes a single gradient step. P_{Θ} is a projection operation, projecting the resulting parameter vector back onto Θ in the event the gradient updates leaves it outside. Bottom: Multiple Imitation Gradient is a generalization of Imitation Gradient where $K \geq 1$ gradient steps are taken with each rollout.

general gradient-based algorithms such as those inspired by the natural gradient and mirror descent. The analysis in this article will focus on the basic online gradient descent algorithm. Online gradient descent proceeds by observing f_n at each iteration and taking a weighted gradient step: $\theta_n - \eta \nabla f_n(\theta_n)$. In the event that the resulting parameter lies outside of Θ , it is projected back on the space with the projection $P_\Theta(\theta) = \arg \min_{\theta' \in \Theta} \|\theta' - \theta\|$. To address the mirror descent case, we would instead define the updates as

$$\theta_{n+1} = \arg \min_{\theta \in \Theta} \eta \langle \nabla f_n(\theta_n), \theta \rangle + \mathcal{D}_h(\theta, \theta_n),$$

where again \mathcal{D}_h is the Bregman divergence with respect a strictly convex function h . Taking $h(\theta) = \frac{1}{2} \|\theta\|^2$ recovers the basic imitation gradient algorithm.

Again, using the l_2 -regularized regression example, the update would take the form

$$\theta_{n+1} = P_\Theta \left(\theta_n - 2\eta \left(\mathbb{E} \left[S_n^\top (S_n - U_n) \right] + \frac{\alpha_0}{2} \theta_n \right) \right).$$

In online optimization (and general convex optimization), the step size is sometimes taken to be variable in order to ensure convergence under different assumptions.

4.3 Multiple Imitation Gradient

We also consider a related algorithm termed Multiple Imitation Gradient, based on the online multiple gradient descent algorithm. This algorithm is identical to the Imitation Gradient, but at each iteration it updates the policy parameters K times, recomputing the gradient each time. Imitation Gradient is a special case of Multiple Imitation Gradient where $K = 1$. It was shown by [Zhang et al. \(2017\)](#) that, for dynamic regret, multiple imitation gradient is asymptotically at least as good as imitation gradient and in some cases it achieves improved rates.

5 Towards Dynamic Regret in Imitation Learning

The imitation learning algorithms presented in the preceding section are derivatives of standard online optimization algorithms. At first glance, the underlying objectives in both fields also seem aligned: find a sequence of policies (parameters) $\{\theta_n\}_{n=1}^N$ such that the cumulative loss $\sum_{n=1}^N f_n(\theta_n)$ is low in some regard. It is only natural then that they are analyzed in a familiar online optimization framework. An algorithm is regarded as performing well if its regret (e.g. R_S or R_D) is *sublinear* in N , written as $o(N)$ ([Hazan, 2016](#)). That is, it performs at least as well as a static or dynamic comparator on average. We note that some authors use the term “no-regret” ([Ross et al., 2011](#)), which means the same as sublinear regret. No-regret refers to the *average* regret $\frac{1}{N} R_S$ vanishing, which occurs if and only if it is sublinear.

However, we must be careful when applying off-the-shelf results. In this section, we discuss the limitations of the standard static regret analysis from the perspective of imitation learning and argue that dynamic regret provides a more informative measure of performance.

5.1 Limitations of Static Regret

The more well known static regret, reproduced here as

$$R_S = \sum_{n=1}^N f_n(\theta_n) - \min_{\theta} \sum_{n=1}^N f_n(\theta),$$

has a rich history of analysis in online optimization. Indeed, this was the first regret definition introduced by [Zinkevich \(2003\)](#). Algorithms have been proposed and analyzed under varying notions of convexity and smoothness to yield sublinear static regret rates ([Hazan et al., 2007](#); [Shalev-Shwartz and Kakade, 2009](#)). Furthermore, the metric has also been used for analysis of stochastic descent methods in convex optimization ([Duchi et al., 2011](#)).

Sublinear static regret implies that the parameters played are at least as good as a single fixed parameter, ensuring that

$$f_N(\theta_N) - \min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^N f_n(\theta)$$

is vanishing (or even negative) on average. Achieving sublinear static regret is important for problems that are online versions of batch learning problems because an algorithm that achieves sublinear static regret is essentially as good as learning if all data was given at once. This validates the use of online learning algorithms for standard statistical learning tasks ([Rakhlin and Sridharan, 2014](#)) and mirrors the types of problems one encounters in classical statistics ([Bubeck, 2011](#)). A canonical example is an email spam filter where a classifier is learned over time as samples of spam and true emails are revealed from a static distribution.

In the context of imitation learning, the static regret metric compares an algorithm’s sequence of policies to the minimizer over the average of losses on all trajectory distributions seen during training. Efficient algorithms exist to achieve static regret for convex and strongly convex losses regardless of any external conditions such as the robot or system dynamics or distributions induced ([Ross et al., 2011](#)). However, the ability to have low regret compared to the average of the past has little bearing at run time. The most we can say about an on-policy algorithm with sublinear dynamic regret is that there exists a policy parameter θ generated by the algorithm such that,

$$f_{\theta}(\theta) \leq \frac{o(N)}{N} + \min_{\theta' \in \Theta} \frac{1}{N} \sum_{n=1}^N f_n(\theta'),$$

where the term $o(N)/N$ is vanishing in N . A common custom in static regret analyses for imitation learning is to define $\epsilon_N = \min_{\theta' \in \Theta} \frac{1}{N} \sum_{n=1}^N f_n(\theta')$ and assume that ϵ_N is small if the policy class is sufficiently expressive ([Ke et al., 2019](#)). In practice, this assumption can actually be very strict as pointed out by [Laskey et al. \(2017\)](#) and [Cheng and Boots \(2018\)](#). It suggests that a policy exists that is simultaneously good on *any* arbitrary collection of N trajectory distributions. That is, for static regret to guarantee a low loss policy, we would need to assume there is nearly zero approximation error (bias) *everywhere* in the state space. Even if such a policy did exist, it was argued by [Laskey \(2018\)](#) that a standard off-policy behavior cloning approach can be more efficient in practice due to increased variance.

Thus, the gap $f_{\theta}(\theta) - \min_{\theta' \in \Theta} f_{\theta}(\theta')$ could still be very large. The intuitive reason is that some distributions observed during training can be irrelevant to the task due to bad initialization or

extremely sensitive dynamics, but they are still included in the average for static regret. As a result, static regret can be sublinear regardless of actual policy performance at run time or whether the algorithm leads to convergent policies. Prior work has shown in experiments and theoretical examples that on-policy algorithms can indeed fail on “hard” problems (Cheng and Boots, 2018; Laskey et al., 2017), but this is not obvious in static regret theory. Thus, it appears that static regret can be overly optimistic and ambiguous as a theoretical metric for imitation learning unless one assumes that ϵ_N is arbitrarily small. In order to amend these deficiencies, we turn to dynamic regret.

5.2 Dynamic Regret

In online optimization literature, dynamic regret has become increasingly popular to analyze online learning problems where the objective is constantly shifting (Hall and Willett, 2015; Hazan and Seshadhri, 2007; Jadbabaie et al., 2015; Mokhtari et al., 2016; Yang et al., 2016; Zhang et al., 2017; Zinkevich, 2003). In contrast to static regret, sublinear dynamic regret implies that

$$f_N(\theta_N) - \min_{\theta \in \Theta} f_N(\theta)$$

is vanishing on average with N . Dynamic regret targets the problem statement where it is important to achieve low loss on each function observed instantaneously such as portfolio optimization (Hazan and Seshadhri, 2007). In this setting, one aims to achieve high returns (low loss) at each point in time as the market changes, rather than perform well compared to a fixed benchmark on average. If the market is shifting over time, i.e. prices are changing, we are interested in playing the best strategy for the given state of the market. A related metric, known as adaptive regret, has also been studied. It observes a window of regret rather than instantaneous regret (Adamskiy et al., 2012; Hazan and Seshadhri, 2007).

To achieve sublinear dynamic regret is difficult, in fact, impossible without further assumptions or knowledge of the underlying environment. For example, suppose the environment plays

$$f_n(\theta) = (\theta - X_n)^2 \quad \text{where} \quad X_n \sim \text{Unif}[-10, 10].$$

There is no hope that any learner could overcome this volatility to achieve sublinear dynamic regret by simply using past information from observed losses (Cheng et al., 2019a; Yang et al., 2016).

Fundamentally, this difficulty of achieving sublinear dynamic regret is dependent on the change in the loss functions over iterations, often expressed in terms of quantities called variations in the literature. If the loss functions change in an unpredictable or adversarial manner, we can expect large variation terms leading to large regret and suboptimal policies. This is the reason that sublinear dynamic regret bounds cannot be obtained in general using only the assumptions commonly used for static regret (Yang et al., 2016). In this article we consider the commonly used path variation and a squared variant of it introduced by Zhang et al. (2017).

Definition 1 (Path Variation and Squared Path Variation). *For a sequence of optimal parameters from m to n given by $\theta_{m:n}^* := \{\theta_i^*\}_{i=m}^n$, the path variation is defined as*

$$V(\theta_{m:n}^*) \stackrel{\text{def.}}{=} \sum_{i=m}^{n-1} \|\theta_i^* - \theta_{i+1}^*\|,$$

and the squared path variation is defined as

$$S(\theta_{m:n}^*) \stackrel{\text{def.}}{=} \sum_{i=m}^{n-1} \|\theta_i^* - \theta_{i+1}^*\|^2.$$

These measures reflect the variation in the loss functions by computing the distances or the square distances between consecutive optimal parameters. Thus the rate in N of a measure of variation generally influences the rate of the resulting dynamic regret bound and so we can often think of the variation as describing the difficulty of the problem.

Many algorithms have been proposed and analyzed in this regret framework in terms of variation measures of the loss functions. For example, [Zinkevich \(2003\)](#) proved a dynamic regret rate of $O(\sqrt{N}(1 + V(\theta_{1:N}^*)))$ for online gradient descent with convex losses. Here, the regret rate is dependent on the rate of the path variation, which might also be a function of N . Therein lies the difficulty of achieving low dynamic regret: no matter the algorithm, rates depend on the variation, which can be large for arbitrary sequences of loss functions.

In the context of imitation learning, dynamic regret compares the n th policy to the instantaneous best policy on the n th distribution, which means it is a stricter metric than static regret. The advantage of the dynamic regret metric is that a policy’s performance at any iteration is always evaluated with respect to the most relevant trajectory distribution: the current one. Proving dynamic regret is low implies that each policy on average is as good as the instantaneous best on its own distribution. Furthermore, we can examine convergence properties and solution characteristics of an algorithm by proving that dynamic regret is low. We formalize this statement in the next section.

We emphasize that the difficulty of achieving sublinear dynamic regret is not a drawback of the dynamic regret metric but rather an important advantage because it reveals whether an imitation learning problem is likely to be solvable by an on-policy algorithm. Achieving low regret implies the algorithm might be successful and find a (near) optimal solution. However, in cases where high dynamic regret is unavoidable, we would be inclined to seek other options.

5.3 Continuity in the Distribution-Generating Parameter

Dynamic regret analyses offer a promising framework for theoretical analysis of on-policy imitation learning algorithms but only as long as the variation can be characterized. In imitation learning, there is considerable structure available that is not possible in general-purpose dynamic regret analyses. This structure prevents pathological cases such as the ones discussed previously. We discuss this notion at a high level to provide intuition.

Recall the imitation learning objective function:

$$f_{\theta'}(\theta) = \mathbb{E}_{\tau \sim p(\tau; \theta')} J_{\tau}(\theta).$$

Here we abuse notation slightly, writing $p(\tau; \theta) = p(\tau; \pi_{\theta})$ and $J_{\tau}(\theta) = J_{\tau}(\pi_{\theta})$. We first consider an impractical but illustrative example where an on-policy algorithm does not even update the robot’s policy. That is,

$$\theta_1 = \theta_2 = \dots = \theta_N.$$

This is a poor strategy, but interestingly the loss functions are identical at each round because the distribution-generating parameter is always the same. This is due to the bifunction form of the objective:

$$f_{\theta_1}(\theta) = \dots = f_{\theta_N}(\theta) \quad \forall \theta \in \Theta.$$

Thus, the variation is always zero. Still the regret could be large if θ_1 is not already optimal on its own distribution, but this gives us hope that the variation can be characterized for certain algorithms exploiting this structure.

Now suppose in a simplified example that we perturb θ_1 by a small amount to generate θ_2 . We expect the robot to apply similar but different controls than before, but in the worst case these different controls could lead to entirely new trajectories if the dynamics are highly irregular. That is $p(\tau; \theta_1)$ could be very different from $p(\tau; \theta_2)$. The cost $J_\tau(\theta)$ would then be evaluated on vastly different trajectories. In this case, due to the irregularity of the dynamics, the costs at each round of an imitation learning algorithm could be just as difficult as the adversarial case in general-purpose dynamic regret analyses, thus leading to unavoidably high dynamic regret.

Alternatively, the system may be sufficiently well-behaved that similar controls give rise to similar trajectories. This notion resembles continuity of the trajectory distribution in the distribution-generating parameter: to ensure $\mathbb{E}_{p(\tau; \theta_1)} J(\theta)$ is necessarily close to $\mathbb{E}_{p(\tau; \theta_2)} J_\theta(\theta)$ we should choose θ_2 to be sufficiently close to θ_1 . This intuition suggests we should restrict our attention to well-behaved systems that exhibit this continuity to avoid the pathological scenarios discussed previously. Indeed, these well-behaved systems are exactly what one might expect to see in many robotics applications. Such a condition is not unlike differentiability assumptions in reinforcement learning literature (Schulman et al., 2015) or continuity assumptions on dynamics studied in nonlinear systems (Khalil and Grizzle, 2002). In Section 8, we explore precedents and examples in detail.

Formally, we describe this notion via Lipschitz continuity. We consider systems that satisfy the following condition for some $\beta \geq 0$:

$$\|\nabla f_{\theta_1}(\theta) - \nabla f_{\theta_2}(\theta)\| \leq \beta \|\theta_1 - \theta_2\| \quad \forall \theta \in \Theta \tag{5}$$

We refer to this condition as distribution continuity. This condition was first introduced for the imitation learning problem by Cheng and Boots (2018). If $\beta = 0$ and $J_\tau(\theta) \neq 0$, this would correspond to a dynamics model where controls have no influence. We emphasize that the difference between the terms on the left-hand side lies entirely in the effect of $p(\tau; \theta_1)$ and $p(\tau; \theta_2)$ being different in the objective. The constant β can be interpreted as measuring the sensitivity of the loss to changes in the distribution-generating parameter. Continuity in the distribution-generating parameter might also be expressed by directly assuming Lipschitz continuity of $p(\tau; \theta)$ but it turns out that this is as actually a stronger assumption because it implies (5), assuming that $\sup_\tau \|\nabla J_\tau(\theta)\|$ is finite for all $\theta \in \Theta$. We discuss this in detail in Section 8.

Continuity in the distribution-generating parameter is a characteristic of the system, which the user does not get to control explicitly. In contrast, the other assumptions made in this paper are largely up to the control of the user by imposing different choices of scalings and loss functions.

6 Main Results

This section presents the primary theoretical results of the paper for the infinite sample or deterministic case as in (Cheng and Boots, 2018; Ross et al., 2011). We aim to achieve the following:

1. To make rigorous the statements in the preceding section, showing that an on-policy algorithm's dynamic regret reveals important theoretical properties regarding convergence.
2. To show that convergence and sublinear dynamic regret can be guaranteed for Algorithms 1-3 under certain conditions on the sensitivity of the trajectory distribution.

In particular, we show that it is possible for all algorithms to achieve $\frac{1}{N}R_D = O(N^{-1})$ average dynamic regret (i.e. sublinear). Furthermore, because dynamic regret upper bounds static regret, these results suggest that we also improve static regret rates of Ross et al. (2011) by a logarithmic factor.

We first formally state the assumptions introduced in Section 3. We begin with common convex optimization assumptions on the loss in the evaluation parameter. Note that all gradients of f in this section are taken with respect to the evaluation parameter, i.e. $\nabla f_{\theta'}(\theta)$ is the gradient with respect to θ , not θ' .

Assumption 1 (Strong Convexity). *For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \alpha > 0$ such that*

$$f_{\theta}(\theta_2) \geq f_{\theta}(\theta_1) + \langle \nabla f_{\theta}(\theta_1), \theta_2 - \theta_1 \rangle + \frac{\alpha}{2} \|\theta_1 - \theta_2\|^2.$$

Assumption 2 (Smoothness and Bounded Gradient). *For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \gamma > 0$ such that*

$$\|\nabla f_{\theta}(\theta_1) - \nabla f_{\theta}(\theta_2)\| \leq \gamma \|\theta_1 - \theta_2\|$$

and $\exists G > 0$ such that $\|\nabla f_{\theta}(\theta_1)\| \leq G$.

Assumption 3 (Stationary Optimum). *For all $\theta' \in \Theta$, θ^* is in the relative interior of Θ where $\theta^* = \arg \min_{\theta \in \Theta} f_{\theta'}(\theta)$. That is, $\nabla f_{\theta'}(\theta^*) = 0$.*

In practice the above conditions are not difficult to satisfy. For example, running DAgger with l_2 -regularized linear regression, i.e. ridge regression, would simultaneously satisfy all three. Finally, we restate the regularity condition on the loss as a function of the distribution-generating parameter discussed in the previous section, which we can interpret as a measure of sensitivity of the loss.

Assumption 4 (Distribution Continuity). *For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \beta > 0$ such that*

$$\|\nabla f_{\theta_1}(\theta) - \nabla f_{\theta_2}(\theta)\| \leq \beta \|\theta_1 - \theta_2\|.$$

This assumption is a Lipschitz regularity on the gradients of the loss, but it is distinct from Assumption 2 in that the Lipschitz continuity is in the distribution-generating parameter, not the evaluation parameter. This is distinct from Assumption 2, where continuity is in the evaluation parameter. The first three assumptions describe conditions imposed on the actual loss function $J_{\tau}(\theta)$; Assumption 4 describes a constraint on the system dynamics by virtue of the trajectory distribution $p(\tau; \theta)$.

6.1 Solution Analysis under Dynamic Regret

Let $\theta_n^* = \arg \min_{\theta \in \Theta} f_n(\theta)$ be the optimal parameter at iteration n . This minimizer is unique because f_n is strongly convex, but it is important to distinguish that this is not the same as the solution to the true objective $\min_{\theta \in \Theta} f_{\theta}(\theta)$. We begin with a result concerning a stability constant $\lambda := \frac{\beta}{\alpha}$. λ represents the ratio of the distribution continuity and the strong convexity.

Proposition 1. Let $\theta, \hat{\theta} \in \Theta$ be two policy parameters inducing loss functions f_θ and $f_{\hat{\theta}}$. Let θ^* and $\hat{\theta}^*$ be the corresponding optimal policies on f_θ and $f_{\hat{\theta}}$. The following inequality holds:

$$\|\theta^* - \hat{\theta}^*\| \leq \lambda \|\theta - \hat{\theta}\|.$$

Proof. Since f_θ is differentiable in the evaluation parameter, the first-order necessary optimality condition implies

$$\langle \nabla f_\theta(\theta^*), \hat{\theta}^* - \theta^* \rangle \geq 0,$$

and the same is true with θ and $\hat{\theta}$ switched. This, combined with strong convexity of f_θ implies

$$\frac{\alpha}{2} \|\theta^* - \hat{\theta}^*\|^2 \leq f_\theta(\hat{\theta}^*) - f_\theta(\theta^*).$$

Applying strong convexity of f_θ a second time and the first-order optimality condition of $f_{\hat{\theta}}$ gives:

$$\begin{aligned} \alpha \|\theta^* - \hat{\theta}^*\|^2 &\leq \langle \nabla f_\theta(\hat{\theta}^*), \hat{\theta} - \theta \rangle \\ &\leq \langle \nabla f_\theta(\hat{\theta}^*) - \nabla f_{\hat{\theta}}(\hat{\theta}^*), \hat{\theta} - \theta \rangle. \end{aligned}$$

Bounding the right-hand side from above with the Cauchy-Schwarz inequality and rearranging terms, we have

$$\|\theta^* - \hat{\theta}^*\| \leq \frac{1}{\alpha} \|\nabla f_\theta(\hat{\theta}^*) - \nabla f_{\hat{\theta}}(\hat{\theta}^*)\| \leq \frac{\beta}{\alpha} \|\theta - \hat{\theta}\|,$$

where the last inequality follows from Assumption 4. \square

There are a couple telling conclusions of this proposition. The first becomes clear when we take the parameters to be consecutive iterates. Let $\theta = \theta_n$ and $\hat{\theta} = \theta_{n+1}$ where $\{\theta_n\}_{n=1}^N$ is a sequence of iterates from an arbitrary on-policy algorithms. The proposition suggests that when $\lambda < 1$, we know with certainty that $\|\theta_{n+1}^* - \theta_n^*\| < \|\theta_{n+1} - \theta_n\|$. In other words, the optimal parameters cannot run away faster than the algorithm's iterates. We can even view this problem as a pursuit-evasion game, where, given Proposition 1, there is hope that we can design an algorithm that catches the optimal parameter. This intuition is also consistent with the findings of prior work (Cheng and Boots, 2018), which shows that convergence of the N th policy can be guaranteed when $\lambda < 1$ for DAGger.

Second, we can consider the self-mapping function $F : \Theta \mapsto \Theta$ defined by $F(\theta) = \arg \min_{\hat{\theta} \in \Theta} f_\theta(\hat{\theta})$. If $\lambda < 1$, by Proposition 1, F is a contraction mapping on Θ . Furthermore since Θ is compact, F has a unique stationary point which is attainable by fixed-point iteration (Banach, 1922). This conclusion naturally leads to the following result.

Corollary 1. If $\lambda < 1$, then there exists a unique stationary point $\theta^* \in \Theta$ such that the policy π_{θ^*} satisfies

$$\mathbb{E}_{p(\tau; \pi_{\theta^*})} J_\tau(\pi_{\theta^*}) = \min_{\theta \in \Theta} \mathbb{E}_{p(\tau; \pi_\theta)} J_\tau(\pi_\theta).$$

Proof. By the Banach fixed-point theorem for the contraction mapping $F(\theta)$, there is a unique stationary point $\theta^* \in \Theta$ satisfying $\theta^* = F(\theta^*)$. By the definition of F , θ^* is the minimizer of $\{\mathbb{E}_{p(\tau; \pi_\theta)} J_\tau(\pi_\theta) : \theta \in \Theta\}$. \square

This result is somewhat surprising. It implies that we cannot get stuck in other stationary points that may be worse than some globally optimal stationary point if $\lambda < 1$. Rather there is a single stationary point in Θ that performs optimally on its own distribution. We can also see that popular on-policy algorithms will preserve this stationary point.

Proposition 2. *Suppose $\lambda < 1$. Let θ^* be a stationary point in Θ . For n between 1 and N , if $\theta_n = \theta^*$, then $\theta_m = \theta^*$ for all $n \leq m \leq N$ for Algorithms 1-3.*

Proof. For DAgger (Algorithm 1), if $\theta_n = \theta^*$, then

$$\theta^* = \arg \min_{\theta \in \Theta} \sum_{m=1}^{n-1} f_m(\theta).$$

By definition $\theta^* = \arg \min_{\theta \in \Theta} f_{\theta^*}(\theta)$. Therefore $\theta^* = \arg \min_{\theta \in \Theta} \sum_{m=1}^n f_m(\theta)$ and so θ^* is a stationary point for DAgger.

For Imitation Gradient (Algorithm 2), if $\theta_n = \theta^*$, then

$$\theta_{n+1} = \arg \min_{\theta \in \Theta} \langle \nabla f_{\theta^*}(\theta^*), \theta - \theta^* \rangle + \frac{1}{2} \|\theta - \theta^*\|^2.$$

Because θ^* minimizes f_{θ^*} , we have $\langle \nabla f_{\theta^*}(\theta^*), \theta - \theta^* \rangle \geq 0$ by the first order optimality condition. Thus, the minimum, which is zero, is achieved at $\theta_{n+1} = \theta^*$ and so θ^* is stationary for this class of gradient-based algorithms. The same holds for Multiple Imitation Gradient (Algorithm 3). Note that this is also true for mirror descent algorithms by substituting in the general Bregman divergence. \square

We emphasize that this notion of stationary points is an inherent property of on-policy algorithms and the imitation learning problem. It is not dependent on any modifications to the algorithms from what was introduced in the original works. However, as discussed in the previous section, convergence of static regret fails to reveal these properties. We now illustrate the power of the dynamic regret analysis, showing that by achieving sublinear dynamic, we can converge to the stationary point and vice-versa.

Theorem 1. *Let $\{\theta_n\}_{n=1}^N$ be a sequence of policy parameters generated by an arbitrary on-policy algorithm. Suppose that $\lambda < 1$. If $R_D(\theta_1, \dots, \theta_N)$ is sublinear, there exists a subsequence of $\{\theta_n\}_{n=1}^N$ that converges to the stationary point θ^* . Conversely, if $\{\theta_n\}_{n=1}^N$ converges to θ^* , then $R_D(\theta_1, \dots, \theta_N)$ is sublinear.*

Proof. Because $\lambda < 1$, we know the stationary point exists and is unique. Define $v(\theta) \stackrel{\text{def.}}{=} f_\theta(\theta) - \min_{\theta'} f_\theta(\theta')$ as a candidate potential function. Note that $v(\theta) \geq 0$ for all $\theta \in \Theta$ and $v(\theta) = 0$ if and only if $\theta = \theta^*$.

We first prove the latter direction. Suppose that $\theta_N \xrightarrow{N} \theta^*$. Note that $v(\theta)$ is continuous at θ^* . Therefore, because $\lim_{N \rightarrow \infty} \theta_N = \theta^*$, we know $\lim_{N \rightarrow \infty} v(\theta_N) = v(\theta^*) = 0$. This implies the desired result:

$$\lim_{N \rightarrow \infty} R_D(\theta_1, \dots, \theta_N) = \lim_{N \rightarrow \infty} \sum_{n=1}^N v(\theta_n) = o(N).$$

To prove the other direction, we are given that the sequence $\{v(\theta_n)\}_{n=1}^N$ is sublinear in dynamic regret. This implies that there is a subsequence $\{v(\theta_{n_k})\}_{k=1}^K$ such that $\lim_{k \rightarrow \infty} v(\theta_{n_k}) = 0$ (e.g. take any monotonically decreasing subsequence). Define the ball of radius ϵ centered at θ^* as

$$B_\epsilon(\theta^*) = \{\theta \in \Theta : \|\theta - \theta^*\| \leq \epsilon\}$$

Because v is continuous at θ^* , for every $\epsilon > 0$, we can choose δ and the corresponding compact level set

$$\Omega_\delta \stackrel{\text{def.}}{=} \{\theta \in \Theta : v(\theta) \leq \delta\},$$

which satisfies $\Omega_\delta \subset B_\epsilon(\theta^*)$. The level set can be made invariant by taking K large enough such that $\forall k \geq K, v(\theta_{n_k}) < \delta$. This is possible because $v(\theta_{n_k})$ converges to zero. Since ϵ was arbitrary, we have that the sequence $\{\theta_{n_k}\}$ converges to θ^* . \square

Remark 1. *If it is known that R_D is sublinear, we can acquire a convergent sequence by taking*

$$\hat{\theta}_N = \arg \min_{\theta \in \{\theta_n\}_{n=1}^N} v(\theta).$$

The function $v(\theta)$ is reminiscent of a Lyapunov function. Indeed, the observation that sublinear dynamic regret implies convergence essentially mirrors the proof of Lyapunov's direct method for asymptotic stability (Khalil and Grizzle, 2002; Sastry, 1999). Generalizations of such functions also appear in literature on variational inequalities as we noted in Cheng et al. (2019a). In this field, they are referred to as merit functions, acting as proxies between the variational inequality problem and the other optimization problems with known solutions (Fukushima, 1996). In our case, we use $v(\theta)$ as a merit function between between the dynamic regret problem and the fixed-point problem, which is reformulation of certain variational inequalities (Facchinei and Pang, 2007). We leave extended discussion on this topic for Section 6.5.

So far, we have shown that achieving sublinear dynamic regret reveals not only that policies are performing optimally on their own distributions, but also that we can guarantee convergence to a unique fixed point when $\lambda < 1$. We now turn our attention to popular on-policy algorithms, showing that it is indeed possible for these algorithms to achieve sublinear regret under conditions on the loss function properties α , β , and γ .

6.2 DAgger

We now introduce a dynamic regret corollary to Theorem 2 of Cheng and Boots (2018). For convenience, we restate their result.

Theorem 2 ((Cheng and Boots (2018), Theorem 2)). *For $\lambda = \frac{\beta}{\alpha}$ and $n \geq 1$, it holds that*

$$f_n(\theta_n) - f_n(\theta_n^*) \leq \frac{(\lambda e^{1-\lambda} G)^2}{2\alpha n^{2(1-\lambda)}},$$

and, if $\lambda < 1$, then $\{\theta_n\}_{n=1}^\infty$ is convergent.

Corollary 2. *For DAgger (Algorithm 1), if $\lambda < 1$, then the average dynamic regret $\frac{1}{N}R_D$ tends towards zero in N with rate $O(\max(N^{-1}, N^{2\lambda-2}))$.*

Proof. The proof is immediate from the result of Theorem 2 of [Cheng and Boots \(2018\)](#). We have

$$f_n(\theta_n) - f_n(\theta_n^*) \leq \frac{(\lambda e^{1-\lambda} G)^2}{2\alpha n^{2(1-\lambda)}}.$$

Summing from 1 to N , we get

$$\begin{aligned} \sum_{n=1}^N f_n(\theta_n) - \sum_{n=1}^N f_n(\theta_n^*) &\leq \sum_{n=1}^N \frac{(\lambda e^{1-\lambda} G)^2}{2\alpha n^{2(1-\lambda)}} \\ &= O(\max(1, N^{2\lambda-1})). \end{aligned}$$

Then the average dynamic regret is

$$\frac{1}{N} R_D = O(\max(N^{-1}, N^{2\lambda-2})),$$

which approaches zero in N . □

The corollary reveals that the convergence result for DAgger proved by Cheng and Boots can be reframed as a dynamic regret analysis. The rate is dependent on the stability constant $\lambda < 1$. The dynamic regret shows that policies generated from DAgger on average achieve local optimality and that for a sufficiently small λ the regret grows no more than a finite amount.

6.3 Imitation Gradient

For the analysis of dynamic regret bounds for the Imitation Gradient algorithm, we require a slightly stronger condition that $\alpha^2 > 2\gamma\beta$. Written another way, the condition is $2\lambda < \psi$ where λ is the stability constant and $\psi := \frac{\alpha}{\gamma}$ is the condition number of f_n . So we require that the problem is both stable and well-conditioned. The proof of this theorem, which can be found in the next section, makes use of the path variation.

Theorem 3. *For Imitation Gradient (Algorithm 2), if $\lambda < 1$, $2\lambda < \psi$ and $\eta = \frac{\alpha(\alpha^2 - 2\gamma\beta)}{2\gamma^2(\alpha^2 - \beta^2)}$, then the average dynamic regret $\frac{1}{N} R_D$ tends towards zero in N with rate $O(N^{-1})$. Furthermore, $f_n(\theta_n) - f_n(\theta_n^*)$ converges to zero and the sequence of policies $\{\theta_n\}_{n=1}^\infty$ is convergent.*

Intuitively, the theorem states that if the conditions are met, the dynamic regret grows no more than a constant value. If we again interpret the variation as describing the difficulty of a problem, this theorem suggests that under the appropriate conditions, solving an imitation learning problem with Imitation Gradient algorithms is as easy as solving a general dynamic regret problem with path variation $V(\theta_{1:N}^*) = O(1)$. In other words, the equivalent dynamic regret problem is stationary in the limit: the optimal parameters cumulatively move no more than a finite distance. The reason is that the change in the loss functions is so closely tied to the policy parameters in imitation learning.

6.4 Multiple Imitation Gradient

We now present a similar dynamic regret rate for the Multiple Imitation Gradient algorithm. This theorem will make use of the squared path variation as opposed to the path variation. The squared path variation is especially amenable for a conversion from the standard dynamic regret bounds to interpretable rates for imitation learning, as we will see in the proof of the theorem.

Theorem 4. *For Multiple Imitation Gradient (Algorithm 3), if $\lambda < 1$, $\lambda \log 4 < \psi^{3/2}$, $\eta < \min \left\{ 1/\gamma, \frac{\alpha^{5/2} - \gamma^{3/2} \beta \log 4}{2\gamma^{3/2} \alpha \beta \log 4} \right\}$, and $K = \lceil \frac{1/\eta + \alpha}{2\alpha} \log 4 \rceil$ then the average dynamic regret $\frac{1}{N} R_D$ tends towards zero in N with rate $O(N^{-1})$. Furthermore, $f_n(\theta_n) - f_n(\theta_n^*)$ converges to zero and the sequence of policies $\{\theta_n\}_{n=1}^\infty$ is convergent.*

The above theorem demonstrates that again a constant upper bound on the dynamic regret can be obtained with this algorithm. Interestingly, the conditions sufficient to guarantee convergence are different. Instead of requiring $2\lambda < \psi$, this theorem requires $\lambda \log 4 < \psi^{3/2}$. While $\log 4 < 2$, we also have that $\psi^{3/2} \leq \psi$ because the condition number, which is the ratio of α and γ , is at most 1.

The implication of this observation is that there is a trade-off. For the Multiple Imitation Gradient algorithm, we can guarantee $O(N^{-1})$ average regret for higher values of λ , i.e. more sensitive systems, but we must then require that our loss functions are better conditioned. Conversely, the Imitation Gradient algorithm achieves a similar guarantee using losses with lower condition numbers, but λ must be smaller.

6.5 Connections to Continuous Online Learning and the Variational Inequality Problem

The results in this section are closely related to the problem of continuous online learning, which we introduced in [Cheng et al. \(2019a\)](#). Indeed, the assumptions that we impose on the bifunction $f_{\theta'}(\theta)$ fall within this framework. One of the key features of continuous online learning is that achieving dynamic regret in these settings is equivalent to solving certain variational inequalities, fixed-point problems, and equilibrium problems, paralleling our results in the previous section. Here, we review the variational inequality problem and its connections to on-policy imitation learning. A detailed overview can be found in [Facchinei and Pang \(2007\)](#).

The variational inequality aims to find solutions to the following problem: for a map $F : \Theta \mapsto \mathbb{R}^d$, find $\theta \in \Theta$ such that

$$\langle F(\theta), \theta' - \theta \rangle \geq 0 \quad \forall \theta' \in \Theta.$$

The problem is written as $\text{VI}(\Theta, F)$ and the set of solutions is written as $\text{SOL}(\Theta, F)$. The map F is said to be monotone if

$$\langle F(\theta) - F(\theta'), \theta - \theta' \rangle \geq 0 \quad \forall \theta, \theta' \in \Theta.$$

It is said to be ζ -strongly monotone if

$$\langle F(\theta) - F(\theta'), \theta - \theta' \rangle \geq \zeta \|\theta - \theta'\|^2 \quad \forall \theta, \theta' \in \Theta.$$

In literature on variational inequalities, strongly monotone functions conduce to efficient algorithms for finding solutions.

In the context of continuous online learning, it can be seen by first order optimality conditions that $f_\theta(\theta) - \min_{\theta' \in \Theta} f_\theta(\theta') = 0$ if and only if θ is a solution to the variational inequality with $F(\theta) \stackrel{\text{def.}}{=} \nabla f_\theta(\theta)$. In [Cheng et al. \(2019a\)](#), we proved that if $\alpha > \beta$, $f_\theta(\cdot)$ is α -strongly convex, and $\nabla f_\theta(\cdot)$ is β -Lipschitz, then $\nabla f_\theta(\theta)$ is a $(\alpha - \beta)$ -strongly monotone map. There are two important implications that are direct results of standard variational inequalities. First, because $\nabla f_\theta(\theta)$ is

strongly monotone and Θ is compact, there is a unique solution to $\text{VI}(\Theta, \nabla f)$. Second, there exist efficient algorithms to find this solutions such as the Basic Projection Algorithm (Facchinei and Pang, 2007)

$$\theta_{n+1} = P_{\Theta}(\theta_n - \eta F(\theta_n)),$$

which exactly corresponds to online gradient descent. There are a number of other properties of variational inequalities, equilibrium problems, and fixed-point problems that are potentially of interest for future work in the on-policy imitation learning problem.

7 Complete Proofs of Algorithm Guarantees

We now present the proofs of the theorems for both gradient-based methods. Both will make heavy use of the following well-known result on the strong convexity of a function (Hazan and Kale, 2014).

Lemma 1. *The following holds for all $\theta \in \Theta$ and $\theta^* = \arg \min_{\theta'} f(\theta')$:*

$$f(\theta) - f(\theta^*) \geq \frac{\alpha}{2} \|\theta - \theta^*\|^2.$$

7.1 Proof of Theorem 3

Before directly proving this theorem, we establish several supporting results based on the path variation.

Lemma 2. *For a sequence of predictions made by Imitation Gradient $\{\theta_n\}_{n=1}^N$ and a sequence of optimal parameters $\{\theta_n^*\}_{n=1}^N$, the following inequality holds on the path variation:*

$$V(\theta_{1:N}^*) \leq \eta \frac{\beta\gamma}{\alpha} \sum_{n=1}^N \|\theta_n - \theta_n^*\|.$$

Proof. From Proposition 1, we have

$$\begin{aligned} \|\theta_{n+1}^* - \theta_n^*\| &\leq \frac{\beta}{\alpha} \|\theta_{n+1} - \theta_n\| \\ &\leq \frac{\beta}{\alpha} \|\eta \nabla f_n(\theta_n)\| \\ &= \eta \frac{\beta}{\alpha} \|\nabla f_n(\theta_n) - \nabla f_n(\theta_n^*)\| \\ &\leq \eta \frac{\beta\gamma}{\alpha} \|\theta_n - \theta_n^*\|, \end{aligned}$$

where the final equality uses Assumption 3 and the final inequality uses Assumption 2. The result follows immediately by summing. \square

Lemma 3. *Let $\rho = (1 - \alpha\eta + \gamma^2\eta^2)^{1/2} > 0$. The following inequality holds for Imitation Gradient*

$$\sum_{n=1}^N \|\theta_n - \theta_n^*\| \leq \|\theta_1 - \theta_1^*\| + \sum_{n=1}^N \rho \|\theta_n - \theta_n^*\| + V(\theta_{1:N}^*).$$

Proof. First note that ρ is always non-negative because, for any positive choice of η , we have $\gamma \geq \alpha$ by definition. By strong convexity we have the following: $0 \leq 2(f_n(\theta_n) - f_n(\theta_n^*)) \leq 2\langle \nabla f_n(\theta_n), \theta_n - \theta_n^* \rangle - \alpha \|\theta_n^* - \theta_n\|^2$. By the update rule given in the algorithm:

$$\begin{aligned} \|\theta_{n+1} - \theta_n^*\|^2 &\leq \|\theta_n - \eta \nabla f_n(\theta_n) - \theta_n^*\|^2 \\ &= \|\eta \nabla f_n(\theta_n)\|^2 + \|\theta_n - \theta_n^*\|^2 \\ &\quad - 2\eta \langle \nabla f_n(\theta_n), \theta_n - \theta_n^* \rangle. \end{aligned} \quad (6)$$

Rearranging the terms in (6) and combining with the very first inequality, we arrive at the following:

$$\|\theta_{n+1} - \theta_n^*\|^2 \leq (1 - \alpha\eta) \|\theta_n - \theta_n^*\|^2 + \|\eta \nabla f_n(\theta_n)\|^2.$$

Using Assumption 2, the smoothness of f_n , and the fact that $\nabla f_n(\theta_n^*) = 0$:

$$\begin{aligned} \|\theta_{n+1} - \theta_n^*\|^2 &\leq \|\theta_n - \theta_n^*\|^2 - \alpha\eta \|\theta_n - \theta_n^*\|^2 \\ &\quad + \eta^2 \|\nabla f_n(\theta_n) - \nabla f_n(\theta_n^*)\|^2 \\ &\leq (1 - \alpha\eta + \gamma^2\eta^2) \|\theta_n - \theta_n^*\|^2. \end{aligned} \quad (7)$$

Let $\rho = (1 - \alpha\eta + \gamma^2\eta^2)^{1/2}$. Following from Mokhtari et al. (2016), we consider the series

$$\begin{aligned} \sum_{n=1}^N \|\theta_n - \theta_n^*\| &\leq \|\theta_1 - \theta_1^*\| + \sum_{n=2}^N \|\theta_n - \theta_{n-1}^*\| + V(\theta_{1:N}^*) \\ &\leq \|\theta_1 - \theta_1^*\| + \sum_{n=1}^N \rho \|\theta_n - \theta_n^*\| + V(\theta_{1:N}^*), \end{aligned}$$

where the first line uses the triangle inequality and the definition of the path variation and the third line uses (7). \square

Proof of Theorem 3. We begin by bounding the result from Lemma 3 above by Lemma 2:

$$\sum_{n=1}^N \|\theta_n - \theta_n^*\| \leq \|\theta_1 - \theta_1^*\| + \left(\rho + \eta \frac{\beta\gamma}{\alpha} \right) \sum_{n=1}^N \|\theta_n - \theta_n^*\|.$$

By rearranging the terms and bounding by the diameter of Θ :

$$\sum_{n=1}^N \|\theta_n - \theta_n^*\| \leq \frac{D}{1 - \rho - \eta \frac{\beta\gamma}{\alpha}}. \quad (8)$$

Choosing $\eta = \frac{\alpha(\alpha^2 - 2\beta\gamma)}{2\gamma^2(\alpha^2 - \beta^2)}$ ensures that $(1 - \rho - \eta \frac{\beta\gamma}{\alpha})$ is positive. By the G -Lipschitz continuity of f_n , we have

$$\sum_{n=1}^N f_n(\theta_n) - \sum_{n=1}^N f_n(\theta_n^*) \leq \frac{GD}{1 - \rho - \eta \frac{\beta\gamma}{\alpha}},$$

and so $R_D(\theta_1, \dots, \theta_N) = O(1)$. Therefore, $\frac{1}{N} R_D = O(1/N)$ which approaches zero.

Interpreting the dynamic regret as a monotone partial sum, we have bounded $\sum_{n=1}^{\infty} f_n(\theta_n) - f_n(\theta_n^*)$ by a constant, meaning that the series converges by the monotone convergence theorem. Because the elements of the partial sum, $f_n(\theta_n) - f_n(\theta_n^*)$, are non-negative, they must then converge to zero. The proof of convergence of the parameters takes a similar approach. From the proof of Lemma 2, we have

$$\frac{1}{\eta\gamma} \|\theta_{n+1} - \theta_n\| \leq \|\theta_n - \theta_n^*\|,$$

The series $\sum_{n=1}^{\infty} \|\theta_n - \theta_n^*\|$ converges due to (8). Thus the series $\sum_{n=1}^{\infty} \|\theta_{n+1} - \theta_n\|$ converges. Since the series of distances between consecutive elements converges, we have that the sequence $\{\theta_n\}_{n=1}^{\infty}$ converges in the limit. \square

7.2 Proof of Theorem 4

For this proof, we again require two technical lemmas modified from Zhang et al. (2017) which are proved in the appendix for completeness.

Lemma 4. *For online multiple gradient descent, let θ' be the current parameter played by the algorithm at any iteration, $\hat{\theta} \stackrel{\text{def.}}{=} P_{\Theta}(\theta' - \eta \nabla f_n(\theta'))$. It holds that*

$$\|\hat{\theta} - \theta_n^*\|^2 \leq \left(1 - \frac{2\alpha}{1/\eta + \alpha}\right) \|\theta' - \theta_n^*\|^2.$$

Lemma 5 ((Zhang et al. (2017), Theorem 3)). *If $\eta < 1/\gamma$ and $K = \lceil \frac{1/\eta + \alpha}{2\alpha} \log 4 \rceil$, then the following is true for online multiple gradient descent:*

$$R_D(\theta_1, \dots, \theta_N) \leq 2\gamma S(\theta_{1:N}^*) + \gamma \|\theta_1 - \theta_1^*\|^2.$$

Proof of Theorem 4. We begin by upper bounding the squared path variation using Proposition 1 and Assumption 4:

$$\begin{aligned} \|\theta_{n+1}^* - \theta_n^*\| &\leq \frac{\beta}{\alpha} \|\theta_{n+1} - \theta_n\| \\ &\leq \frac{\beta\eta}{\alpha} \|\nabla f_n(\theta_n^1) + \dots + \nabla f_n(\theta_n^K)\| \\ &\leq \frac{\beta\eta}{\alpha} \sum_{j=1}^K \|\nabla f_n(\theta_n^j)\| \\ &= \frac{\beta\eta}{\alpha} \sum_{j=1}^K \|\nabla f_n(\theta_n^j) - \nabla f_n(\theta_n^*)\| \end{aligned}$$

Next we apply Lemma 4:

$$\begin{aligned}
\|\theta_{n+1}^* - \theta_n^*\| &\leq \frac{\beta\eta\gamma}{\alpha} \sum_{j=1}^K \|\theta_n^j - \theta_n^*\| \\
&\leq \frac{\beta\eta\gamma}{\alpha} \|\theta_n - \theta_n^*\| \sum_{j=1}^K \left(1 - \frac{2\alpha}{1/\eta + \alpha}\right)^j \\
&\leq \frac{\beta\eta\gamma K}{\alpha} \|\theta_n - \theta_n^*\|
\end{aligned}$$

Summing over all n , we have

$$S(\theta_{1:N}^*) \leq \left(\frac{\beta\eta\gamma K}{\alpha}\right)^2 \sum_{n=1}^N \|\theta_n - \theta_n^*\|^2$$

Then by substituting into Lemma 5 and bounding the quantity from above using the strong convexity of f_n , we have

$$\begin{aligned}
R_D &\leq \gamma \|\theta_1 - \theta_1^*\|^2 + \frac{4\beta^2\eta^2\gamma^3 K^2}{\alpha^3} \sum_{n=1}^N f_n(\theta_n) - f_n(\theta_n^*) \\
&\leq \frac{\gamma \|\theta_1 - \theta_1^*\|^2}{1 - \frac{4\beta^2\eta^2\gamma^3 K^2}{\alpha^3}} \leq \frac{\gamma D^2}{1 - \frac{4\beta^2\eta^2\gamma^3 K^2}{\alpha^3}}.
\end{aligned}$$

For $\eta < \min\left\{1/\gamma, \frac{\alpha^{5/2} - \gamma^{3/2}\beta \log 4}{2\gamma^{3/2}\alpha\beta \log 4}\right\}$ and $K > \frac{1/\eta + \alpha}{2\alpha} \log 4$, the denominator is positive. So the regret is again $O(1)$.

The proof of convergence of $f_n(\theta_n) - f_n(\theta_n^*)$ to zero is identical to that of Theorem 3. The proof of convergence of $\{\theta_n\}_{n=1}^\infty$ is similar. An intermediate result of Lemma 5 yields

$$\|\theta_{n+1} - \theta_n^*\| \leq \frac{1}{2} \|\theta_n - \theta_n^*\|. \quad (9)$$

Then, as before, we bound the sum of the differences between policy parameters and optima, but without the square:

$$\sum_{n=1}^N \|\theta_n - \theta_n^*\| \leq 2\|\theta_1 - \theta_1^*\| + 2V(\theta_{1:N}^*)$$

By bounding the path variation from above using Lemma 4 and again rearranging terms, a constant upper bound on $\sum_{n=1}^N \|\theta_n - \theta_n^*\|$ is established in the same way as Theorem 3 except we require $1 > \frac{2\beta\eta\gamma K}{\alpha}$. This is satisfied with the same condition on η . The rest of proof proceeds exactly in the same way as Theorem 3. Therefore with Multiple Imitation Gradient, $\{\theta_n\}_{n=1}^N$ converges in the limit. \square

8 Systems with Distribution Continuity

Fundamentally, the preceding results for on-policy imitation learning and core idea behind continuous online learning rely on the supposition that there is Lipschitz continuity in $\nabla f_{\theta'}(\theta)$ with

respect to θ' , the distribution-generating parameter. However, it may not be immediately clear when this condition is satisfied in scenarios of interest. In this section, we discuss several examples of well-studied MDPs that satisfy this condition.

8.1 Linear Dynamical Systems

We can show that the ridge regression loss on a linear dynamical system satisfies Assumption 4. We consider the following system:

$$s_{t+1} = As_t + Bu_t,$$

where $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{m \times 1}$ and the fixed initial state is $s_1 \in \mathbb{R}^m$. The policies give linear feedback with $u_t = \theta^\top s_t$ with $\theta \in \mathbb{R}^m$, so the closed loop dynamics are

$$s_{t+1} = (A + B\theta^\top)s_t.$$

We consider a supervisor policy that is affine in the state $Ks_t + k$. Over a full trajectory, the loss function with distribution-generating parameter θ' and evaluation parameter θ is

$$f_{\theta'}(\theta) = \left\| \begin{bmatrix} s_1^\top \\ \vdots \\ s_{T-1}^\top \end{bmatrix} \theta - \begin{bmatrix} (Ks_1 + k)^\top \\ \vdots \\ (Ks_{T-1} + k)^\top \end{bmatrix} \right\|^2 + \frac{\alpha}{2} \|\theta\|^2$$

where $s_t = (A + B(\theta')^\top)^{t-1}s_1$. The derivative is then

$$\nabla f_{\theta'}(\theta) = \begin{bmatrix} s_1 s_1^\top \\ \vdots \\ s_{T-1} s_{T-1}^\top \end{bmatrix} \theta - \begin{bmatrix} s_1 (Ks_1 + k)^\top \\ \vdots \\ s_{T-1} (Ks_{T-1} + k)^\top \end{bmatrix} + \alpha \theta$$

Because of the definition of s_t , we can see that $\nabla f_{\theta'}(\theta)$ is a polynomial map in θ' . Therefore, on the compact set Θ , its derivative is continuous and bounded, so $\nabla f_{\theta'}(\theta)$ is Lipschitz continuous.

The supervisor policy can be polynomial in the state and the result is the same due to composition of polynomials. We can also extend this to multiple input systems but, notationally, this requires redefining Θ as a subset of matrices.

8.2 Lipschitz Markov Decision Processes

Another problem class that satisfies the conditions of Assumption 4 is ‘‘Lipschitz MDPs’’ (Asadi et al., 2018; Bertsekas, 1975; Hinderer, 2005; Pirotta et al., 2015). This category of Markov decision processes studied in reinforcement learning theory is restricted to exhibit smoothness in the transition dynamics and the reward. Before discussing in detail the formal Lipschitz MDP setting, we first return to an intuitive example that illustrates this smoothness of dynamics from Section 5.

The left-hand side of the inequality in Assumption 4, can be expanded as

$$\begin{aligned} \|\nabla f_{\theta_1}(\theta) - \nabla f_{\theta_2}(\theta)\| &= \left\| \int \nabla J_\tau(\theta) (p(\tau; \pi_{\theta_1}) - p(\tau; \pi_{\theta_2})) d\tau \right\| \\ &\leq \sup_\tau \|\nabla J_\tau(\theta)\| \int |p(\tau; \pi_{\theta_1}) - p(\tau; \pi_{\theta_2})| d\tau \\ &= \sup_\tau 2\|\nabla J_\tau(\theta)\| d_{TV}(p(\tau; \pi_{\theta_1}), p(\tau; \pi_{\theta_2})) \end{aligned}$$

where $d_{TV}(p, q)$ denotes the total variation between distributions p and q . If $\sup_{\tau} \|\nabla J_{\tau}(\theta)\|$ is finite, then Assumption 4 depends only on the condition that small changes in the policy parameters induce small changes in the trajectory distributions with respect to total variation distance. That is, if $p(\tau|\pi_{\theta_2})$ is L_{TV} -Lipschitz

$$d_{TV}(p(\tau|\pi_{\theta_1}), p(\tau|\pi_{\theta_2})) \leq L_{TV} \|\theta_1 - \theta_2\|,$$

then Assumption 4 is satisfied with

$$\beta = \sup_{\tau} 2 \|\nabla J_{\tau}(\theta)\| L_{TV}.$$

This condition is similar to differentiability assumptions in reinforcement learning (Schulman et al., 2015). While it conveys intuition, the Lipschitz continuity of the trajectory distribution in total variation distance is a higher level assumption than what is typically considered in Lipschitz MDPs. However, it can be derived from lower level assumptions. In our discussion of this setting, we will follow the assumptions of Pirotta et al. (2015), who consider an infinite horizon ($T \rightarrow \infty$) and γ -discounted version of the MDP \mathcal{M} for $\gamma \in (0, 1)$.

The Lipschitz MDP has regularity in the transition dynamics $p(\cdot|s, u)$ and the reward function. In the imitation learning setting, we do not assume access to a reward function but we can view the loss $\ell(s, \theta)$ in its place. Let $d_{\mathcal{S}, \mathcal{U}}((s, u), (s', u')) = d_{\mathcal{S}}(s, s') + d_{\mathcal{U}}(u, u')$ denote the sum of distance metrics on the state set \mathcal{S} and control set \mathcal{U} . Lipschitz MDPs assume there exists $L_p \geq 0$ such that

$$\mathcal{K}(p(\cdot|s, u), p(\cdot|s', u')) \leq L_p d_{\mathcal{S}, \mathcal{U}}((s, u), (s', u')).$$

where \mathcal{K} is the Kantorovich distance, defined for probability measures p and q as

$$\mathcal{K}(p, q) \stackrel{\text{def.}}{=} \sup \left\{ \int f(x) (dp - dq) : f \text{ is 1-Lipschitz cts.} \right\}$$

Due to its generality, the Kantorovich distance (a specific case of the Wasserstein distance) is used extensively in literature on Lipschitz MDPs in place of the total variation, but the choice of distribution distance does not ultimately affect Assumption 4. In addition, we assume that we have chosen a loss function ℓ such that the partial derivatives with respect to θ are L_{ℓ} -Lipschitz:

$$\left| \frac{\partial \ell(s, \theta)}{\partial \theta_i} - \frac{\partial \ell(s, \theta')}{\partial \theta'_i} \right| \leq L_{\ell} \|\theta - \theta'\|$$

We consider an alternative, but equivalent form of the imitation learning objective. Rather than defining a trajectory distribution, we define an average state distribution δ_{θ} under π_{θ} such that $\mathbb{E}_{\tau \sim p(\tau|\theta)} J_{\tau}(\theta) = \mathbb{E}_{s \sim \delta_{\theta}} \ell(s, \theta)$. This is a standard reformulation (Ross et al., 2011). Lipschitz continuity of δ_{θ} in the Kantorovich distance was shown by Pirotta et al. (2015).

Lemma 6 ((Pirotta et al., 2015)). *If $\pi_{\theta}(\cdot|s)$ is $L_{\mathcal{S}}$ -Lipschitz in $s \in \mathcal{S}$ and Lipschitz in $\theta \in \Theta$ and $\gamma L_p(1 + L_{\mathcal{S}}) < 1$, then there exists a finite $L_{\delta} > 0$ such that*

$$\mathcal{K}(\delta_{\theta_1}, \delta_{\theta_2}) \leq L_{\delta} \|\theta_1 - \theta_2\|.$$

Algorithm 4 Adaptive On-Policy Regularization (AOR)

Input: Initial parameters θ_1 ,
Max iterations N ,
Initial regularization $\hat{\alpha}_1$.

for $n = 1$ **to** $N - 1$ **do**
Roll out θ_n and collect trajectory τ_n .
Observe loss $f_n(\theta) = f_{\theta_n}(\theta)$ from feedback on τ_n .
 $\theta_{n+1} \leftarrow \text{UPDATE}(\theta_n, \hat{\alpha}_n)$.
 $\hat{\lambda}_n \leftarrow \text{ESTIMATE}(f_1, \dots, f_n)$.
 $\hat{\alpha}_{n+1} \leftarrow \text{TUNE}(\hat{\alpha}_n, \hat{\lambda}_n)$.

end for

Adaptive On-Policy Regularization adaptively increases α , the regularization parameter for strong convexity, to stabilize any regularized on-policy algorithm.

Then we can show that Assumption 4 is satisfied when the conditions of the previous lemma are satisfied:

$$\begin{aligned} \|\nabla f_{\theta_1}(\theta) - \nabla f_{\theta_2}(\theta)\| &= \left\| \int \nabla \ell(s, \theta)(d\delta_{\theta_1} - d\delta_{\theta_2}) \right\| \\ &\leq \sum_{i=1}^d \left| \int \frac{\partial \ell}{\partial \theta_i}(d\delta_{\theta_1} - d\delta_{\theta_2}) \right| \\ &\leq dL_\ell \mathcal{K}(\delta_{\theta_1}, \delta_{\theta_2}) \\ &\leq dL_\ell L_\delta \|\theta_1 - \theta_2\|. \end{aligned}$$

In this case, we observe that $\beta = dL_\ell L_\delta$. This upper bound is fairly loose and we conjecture that the additional dimension dependence is more pessimistic than necessary.

9 Adaptive On-Policy Regularization

We now apply these theoretical results to motivate an adaptive regularization algorithm to help ensure convergence. As noted by Cheng and Boots (2018), regularization can lead to convergent policies for DAgger.

In all the theoretical results, a key sufficient condition is that $\lambda < 1$, meaning that the strong convexity constant α must be greater than the regularity constant β . While β is a fixed property of the dynamics, α is largely controllable by the user and robot. α can be increased by increasing the regularization of the supervised learning loss. By Proposition 1, a lower bound on λ may be estimated by finding the ratio of the distance between optimal parameters and the distance between policy parameters:

$$\hat{\lambda} = \frac{\|\theta_{n+1}^* - \theta_n^*\|}{\|\theta_{n+1} - \theta_n\|}.$$

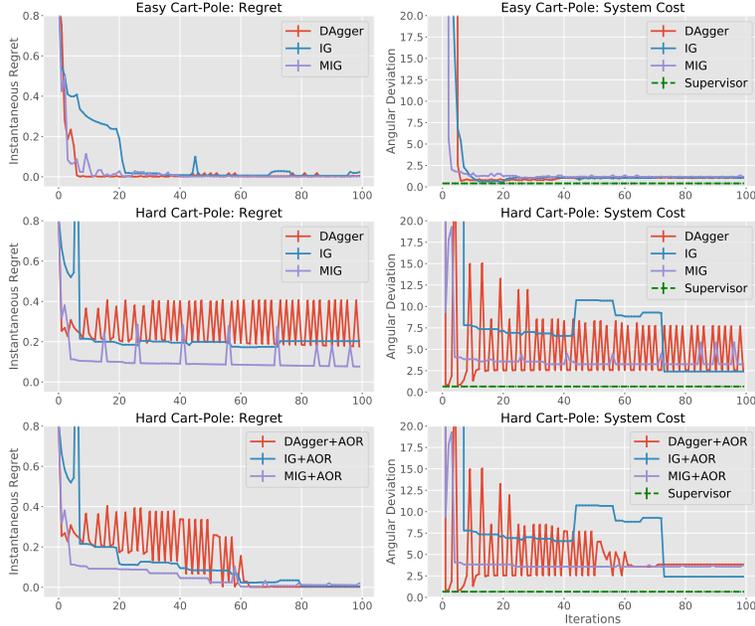


Figure 1: For the cart-pole balancing task, regret is shown in the right column while true system cost, measured as angular deviation, is shown in the left column. All three on-policy algorithms without adaptive regularization are shown on both the easy (top row) and hard (middle, bottom rows) settings of cart-pole. The result shows existence of a system is difficult enough to induce unstable learning curves. The bottom row shows the algorithms again on the hard setting but using AOR. The dynamic regret tends towards zero and the chattering in DAgger and MIG is reduced.

Therefore, we can propose an adaptive algorithm to compute a new regularization term at each iteration n . One caveat of adaptively updating α is that we do not want it to be too large. While the regret will converge, the policy performance can suffer as the regularization term will dominate the loss function and thus the convergent solution will simply be the solution that minimizes the regularizer. This subtlety and the theoretical motivation in the previous sections are the basis for Algorithm 4, which we call Adaptive On-Policy Regularization, an algorithm for making conservative updates to α that can be applied to any regularized on-policy algorithm. At each iteration, the policy is updated according to a given on-policy algorithm such as DAgger using subroutine $\text{UPDATE}(\theta_n, \hat{\alpha}_n)$ which depends on the current regularization. Then $\hat{\lambda}_n$ is computed with subroutine ESTIMATE . We use a mean of observed λ values over iterations. Finally, the $\hat{\alpha}_{n+1}$ is tuned based on $\hat{\lambda}_n$. We use a linear weighting update rule: $\hat{\alpha}_{n+1} = t\hat{\lambda}_n\hat{\alpha}_n + (1-t)\hat{\alpha}_n$ for $t \in (0, 1)$, where $\hat{\lambda}_n\hat{\alpha}_n$ is a conservative estimate of β .

10 Empirical Evaluation

We study the empirical properties of DAgger, Imitation Gradient (IG) and Multiple Imitation Gradient (MIG), showing that even in low dimensional and convex settings, the implications of the convergence of policies become apparent. We intentionally sought out cases and chose parameters such that these on-policy algorithms do not achieve convergence in order to better understand their properties. We consider deterministic domains for the sake of accurately measuring the true dynamic regret. In this evaluation, we attempt to address the following questions: (1) How are policy performance and dynamic regret affected by changing system parameters? (2) Can

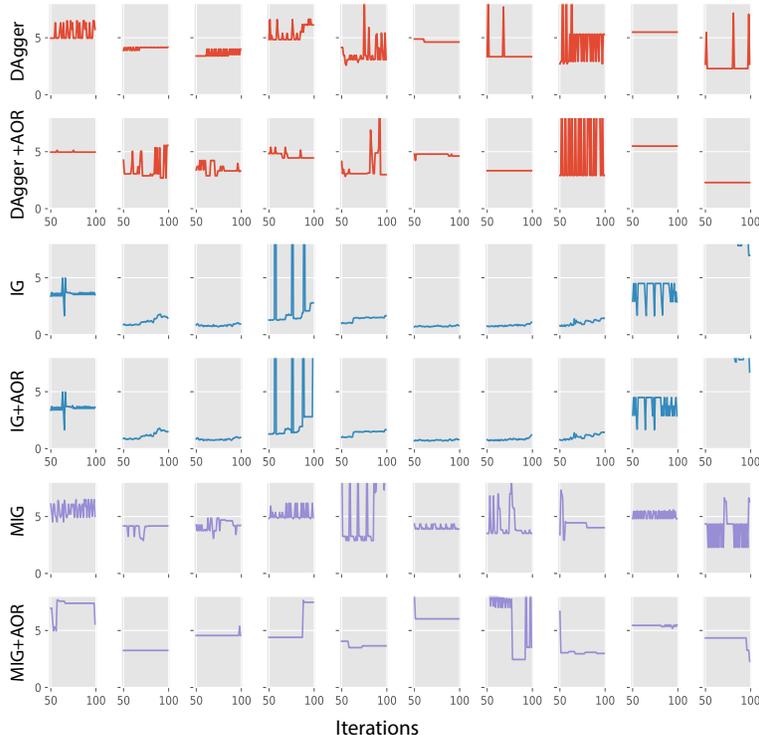


Figure 2: The cart-pole experiment was repeated over 10 different initial pole angle conditions for each of the algorithms with and without AOR. The cost curves, measured in angular deviation, are shown. Columns correspond to different trials with different initial conditions. Each color group represents a different base algorithm. The top row of each color group is the algorithm without adaptive regularization. The bottom rows utilize adaptive regularization. AOR is generally able to stabilize the cost curves while maintaining low cost.

Adaptive On-Policy Regularization improve convergence of the average dynamic regret and policy performance? The code for all experiments can be found at <https://github.com/jon--lee/aor>.

10.1 Cart-Pole Balancing

First we consider a task where the robot learns to push a cart left or right with a fixed force magnitude in order to balance a pole upright over 100 iterations. The control space is discrete $\{\text{left}, \text{right}\}$ and the state space consists of cart location and velocity and pole angle and angular velocity. We measure the absolute performance of a policy as the angular deviation from the upright position. We obtained a nonlinear algorithmic supervisor via reinforcement learning. The robot’s policy was learned using l_2 -regularized linear regression as in (4). In this setting, we vary the difficulty of the problem, i.e. controlling β , by setting the force magnitude to either low or high values corresponding to easy and hard settings, respectively. For all algorithms, the regularization was initially set to $\hat{\alpha}_1 = 0.1$. Stepsizes η for IG and MIG were set to 0.0001 and 0.01, respectively.

To generate the easy and hard versions of cart-pole balancing, we varied the parameter controlling the force magnitude applied with each left or right control. A low force magnitude of 2.0 was used for the easy setting and a higher force magnitude of 10.0 was used for the hard setting. The value 2.0 was the smallest integer before the force was too low to control the cart. The value 10.0 was one of the highest before we noticed the average λ values began to decrease as a function of

the force magnitude. As mentioned, the parameters η and α_1 were intentionally chosen so that the task would exhibit unstable or suboptimal results. Trajectories were 200 time steps long.

In order to estimate $\hat{\lambda}_n$ between each iteration, we compute $\|\theta_n - \theta_{n-1}\|$ directly since both quantities are known. Because each f_n are strongly convex supervised learning problems, full information is known f_n and so the minimum $\theta_n^* = \arg \min f_n(\theta)$ can be solved. In this case, f_n corresponded to a l_2 -regularized ridge regression problem which has a closed form solution. In the case of stochastic problems, it may be necessary to obtain a sample estimate of f_n first by collecting several trajectories per iteration and then estimating θ_n^* from the sample average. Our estimate of $\hat{\lambda}_n$ was simply the ratio of these normed differences averaged over iterations. Note that λ as defined in Assumption 4, is a global constant, but in practice only local regions may be relevant for the problem at hand, which is why we estimate λ at each iteration.

The instantaneous regrets for all three imitation learning algorithms, measured as $f_n(\theta_n) - \min_{\theta} f_n(\theta)$, are shown in the left column of the top and middle rows of Figure 1 for both the easy and hard settings, respectively. In the right column are the actual angular deviations of the pole. In the easy setting, the algorithms converge to costs similar to the supervisor. In the hard setting, there is chattering during the learning process. We observe that regret does not always converge indicating a discrepancy between the supervisor and the robot, which is consistent with the theoretical results that suggest difficulty influences convergence to the best policy.

The bottom row of Figure 1 shows that Adaptive On-Policy Regularization (AOR) can be used to improve dynamic regret. In our implementation of AOR, we set $t = 0.01$ and updated $\hat{\lambda}_n$ and $\hat{\alpha}_n$ every ten iterations. For IG and MIG with AOR, we adjusted the stepsize η as a function of $\hat{\alpha}_n$ as motivated by the conditions in the theorems. In practice this counteracts potentially large gradients caused by large $\hat{\alpha}$ values. DAgger without AOR exhibits severe chattering. With AOR, the performance is stabilized, leading to a converged policy. A similar result is observed for MIG. We note that IG did not initially exhibit chattering and the learning curve was unaffected. We attribute this to the discrete nature of the control space; even if the policy parameters have different regret rates, the resulting trajectories could be the same.

We repeated the cart-pole experiment over 10 different initial pole angle conditions in the hard setting in Figure 2. The cost curves are shown, measured as angular deviation from the upright position. The top row of each pair shows without AOR and the bottom row shows with AOR. Each column corresponds to a different set of initial conditions. To conserve space, only the last 50 iterations are shown, which is when the curves are typically stabilized by adaptive regularization. DAgger and MIG see a reduction of chattering in most cases when using AOR, while we observe no difference for IG. Interestingly, IG is fairly stable by itself.

10.2 Walker Locomotion

Next, we consider a 2-dimensional walker from the OpenAI Gym, where the objective is to move the farthest distance. Again we induced difficulty and suboptimal policies by increasing the force magnitude of controls. Regularized regression was again used for the robot’s policy. Here, we set $\alpha_1 = 1.0$ and $t = 0.1$ for AOR. We used the same initial η values for IG and MIG and adjusted the stepsize as a function of $\hat{\alpha}_n$ when using AOR. The results are shown in Figure 3. Without adaptive regularization, the average dynamic regret fails to converge and all distance curves exhibit severe chattering with no stable learning. With AOR, average dynamic regret converges to zero and all distance curves are stabilized.

To induce high regret policies in the walker domain, we simply increased the force of controls

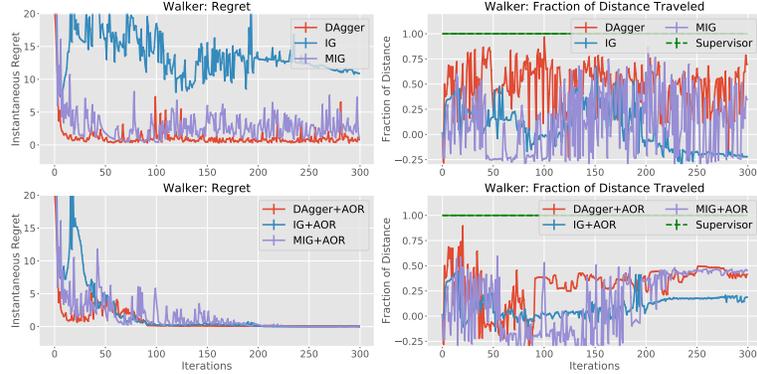


Figure 3: In the 2-dimensional Walker locomotion domain, the instantaneous regrets fail to converge to zero without AOR (top row) and the distance curves exhibit chattering. With AOR (bottom row), the average dynamic regret converges and chattering is reduced after 200 iterations

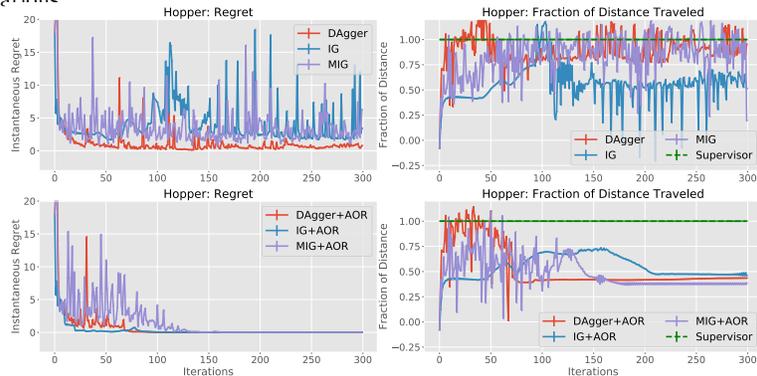


Figure 4: The algorithms are evaluated on OpenAI Gym hopper on the same conditions as the walker experiment. As observed in prior experiments, on-policy algorithms without proper regularization lead to unstable learning. With adaptive regularization the learning is stabilized; however, in this case, performance on the system is reduced on average.

five-fold. As in cart-pole, trajectories consisted of 200 time steps and one trajectory was collected and evaluated at each of the 300 iterations. Again, there was no stochasticity in the environment for the sake of computing the instantaneous regret.

10.3 Hopper Locomotion

We also evaluated the effect of AOR on a different OpenAI gym task, hopper, with exactly the same hyperparameter settings. The results are shown in Figure 4. We note that although chattering is reduced, the distance traveled on average suffers. This is one disadvantage of adaptive regularization that was noted earlier. Increased regularization, although leading to convergence of average dynamic regret, can cause poor policy performance if it is excessive. This example reaffirms that from a practical perspective in imitation learning, it is important to monitor both the surrogate loss and whatever qualitative or quantitative metrics are actually desired to ensure that they agree.

11 Discussion and Future Work

Dynamic regret theory offers a promising framework for theoretical analysis in imitation learning. The question of whether an on-policy algorithm leads to converged or stable policies is inherently captured in dynamic regret, in contrast to static regret which captures policy performance on the average of the distributions. The theoretical analyses suggest new conditions to guarantee convergence. The simulation results suggest that the questions of convergence and optimality must be addressed when designing imitation learning robotics systems because stable performance is not always guaranteed. Indeed, even if static regret is found to be low, the distributions induced by the robot during training may be far too unpredictable for stable policy performance to actually be achieved.

Our results suggest that if the problem is sufficiently well-conditioned, we can guarantee convergence to a unique solution for these on-policy algorithms. This is surprising because intuitively initialization would seem to be an important factor, but does not ultimately affect the solution in the limit. We also draw connections between the on-policy imitation learning problem and variational inequalities and fixed-point problems by virtue of the continuous online learning framework (Cheng et al., 2019a). Further exploration of these connections may reveal additional solution characteristics and efficient algorithms for solving imitation learning problems.

By modeling the problem with Assumption 4, we constrained the dynamics to aid the analysis. If other properties were known about the dynamics, then additional information could improve regret rates and conditions for optimality. For example, similar problem statements were studied in Hall and Willett (2015) and Rakhlin and Sridharan (2013) in general online optimization. It was recently shown that augmenting the mostly model-free analyses with model-based learning can improve static regret bounds and performance (Cheng et al., 2019b). We hypothesize that dynamic regret rates can also be improved.

Furthermore, Assumption 4 excludes certain difficult systems with discontinuous dynamics such as hybrid systems. While the purpose of this article is to provide foundational theoretical results in the continuous setting, it is still not known whether these results can be applied to systems where the loss may be discontinuous on the parameter space. One possible direction is to show there are local regions that are sufficiently continuous to apply the current results and also invariant under the given on-policy algorithm. The problem then reduces to reaching these invariant sets. We leave exploration of this fascinating direction of discontinuities for future work.

References

- Dmitry Adamskiy, Wouter M Koolen, Alexey Chernov, and Vladimir Vovk. A closer look at adaptive regret. In *International Conference on Algorithmic Learning Theory*, pages 290–304. Springer, 2012.
- Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273, 2018.
- J Andrew Bagnell. An invitation to imitation. Technical report, Carnegie Mellon Univ Pittsburgh PA Robotics Inst, 2015.
- Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. math*, 3(1):133–181, 1922.

- D Bertsekas. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(3):415–419, 1975.
- Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, pages 1–86, 2011.
- Ching-An Cheng and Byron Boots. Convergence of value aggregation for imitation learning. *International Conference on Artificial Intelligence and Statistics*, 2018.
- Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning through imitation and reinforcement. In *Conference on Uncertainty in Artificial Intelligence*, 2018.
- Ching-An Cheng, Jonathan Lee, Ken Goldberg, and Byron Boots. Online learning with continuous variations: Dynamic regret and reductions. *arXiv preprint arXiv:1902.07286*, 2019a.
- Ching-An Cheng, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Accelerating imitation learning with predictive models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019b.
- Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in neural information processing systems*, pages 1087–1098, 2017.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.
- Felix Duvallet, Thomas Kollar, and Anthony Stentz. Imitation learning for natural language direction following through unknown environments. In *ICRA*, pages 1047–1053. IEEE, 2013.
- Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.
- Masao Fukushima. Merit functions for variational inequality and complementarity problems. In *Nonlinear Optimization and Applications*, pages 155–170. Springer, 1996.
- Eric C Hall and Rebecca M Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.
- Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.
- Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1): 2489–2512, 2014.
- Elad Hazan and Comandur Seshadhri. Adaptive algorithms for online decision problems. *Electronic colloquium on computational complexity (ECCC)*, 2007.
- Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- Karl Hinderer. Lipschitz continuity of value functions in markovian decision processes. *Mathematical Methods of Operations Research*, 62(1):3–22, 2005.

- Ahmed Hussein, Eyad Elyan, Mohamed Medhat Gaber, and Chrisina Jayne. Deep imitation learning for 3d navigation tasks. *Neural Computing and Applications*, 29(7):389–404, 2018.
- Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406, 2015.
- Liyiming Ke, Matt Barnes, Wen Sun, Gilwoo Lee, Sanjiban Choudhury, and Siddhartha Srinivasa. Imitation learning as f -divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019.
- Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.
- Michael Laskey, Caleb Chuck, Jonathan Lee, Jeffrey Mahler, Sanjay Krishnan, Kevin Jamieson, Anca Dragan, and Ken Goldberg. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA), 2017*, 2017.
- Michael David Laskey. *On and Off-Policy Deep Imitation Learning for Robotics*. PhD thesis, UC Berkeley, 2018.
- Hoang Le, Andrew Kang, Yisong Yue, and Peter Carr. Smooth imitation learning for online sequence prediction. In *International Conference on Machine Learning*, pages 680–688, 2016.
- Jonathan Lee, Michael Laskey, Ajay Kumar Tanwani, Anil Aswani, and Ken Goldberg. A dynamic regret analysis and adaptive regularization algorithm for on-policy robot imitation learning. *Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2018a.
- Jonathan Lee, Michael Laskey, Ajay Kumar Tanwani, and Ken Goldberg. Stability analysis of on-policy imitation learning algorithms using dynamic regret. In *RSS Workshop on Imitation and Causality*, 2018b.
- Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 7195–7201. IEEE, 2016.
- Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018.
- Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile off-road autonomous driving using end-to-end deep imitation learning. In *Robotics: Science and Systems*, 2018.
- Matteo Pirodda, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2-3):255–283, 2015.
- Dean A Pomerleau. *Alvinn: An autonomous land vehicle in a neural network*. Technical report, Carnegie-Mellon University, 1989.
- A Rakhlin and K Sridharan. Statistical learning and sequential prediction. *Book Draft*, 2014.

- Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.
- Stéphane Ross, Geoffrey J Gordon, and J Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *International Conference on Artificial Intelligence and Statistics*, 2011.
- Stéphane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J Andrew Bagnell, and Martial Hebert. Learning monocular reactive uav control in cluttered natural environments. In *2013 IEEE international conference on robotics and automation*, pages 1765–1772. IEEE, 2013.
- Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer-Verlag New York, 1999.
- John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, 2015.
- Shai Shalev-Shwartz and Sham M Kakade. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *Advances in Neural Information Processing Systems*, pages 1457–1464, 2009.
- Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply ag-grevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*, 2017.
- Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, 2016.
- Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end simulated driving. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- Lijun Zhang, Tianbao Yang, Jinfeng Yi, Jing Rong, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems*, 2017.
- Tianhao Zhang, Zoe McCarthy, Owen Jowl, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.
- Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

A Omitted Proofs

For convenience, the main assumptions of the paper are reproduced here:

- For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \alpha > 0$ such that

$$f_\theta(\theta_2) \geq f_\theta(\theta_1) + \langle \nabla f_\theta(\theta_1), \theta_2 - \theta_1 \rangle + \frac{\alpha}{2} \|\theta_1 - \theta_2\|^2.$$

- For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \gamma > 0$ such that

$$\|\nabla f_\theta(\theta_1) - \nabla f_\theta(\theta_2)\| \leq \gamma \|\theta_1 - \theta_2\|$$

and $\exists G > 0$ such that $\|\nabla f_\theta(\theta_1)\| \leq G$.

- For all $\theta' \in \Theta$, θ^* is in the relative interior of Θ where $\theta^* = \arg \min_{\theta \in \Theta} f_{\theta'}(\theta)$. That is, $\nabla f_{\theta'}(\theta^*) = 0$.

- For all $\theta_1, \theta_2, \theta \in \Theta$, $\exists \beta > 0$ such that

$$\|\nabla f_{\theta_1}(\theta) - \nabla f_{\theta_2}(\theta)\| \leq \beta \|\theta_1 - \theta_2\|.$$

A.1 Proof of Lemma 4

The result of Lemma 4 is based on a technical lemma from [Zhang et al. \(2017\)](#).

Proof. By the update rule:

$$\begin{aligned} \hat{\theta} &= P_\Theta(\theta' - \eta \nabla f_n(\theta')) \\ &= \arg \min_{\theta \in \Theta} \|\theta' - \eta \nabla f_n(\theta') - \theta\|^2 \\ &= \arg \min_{\theta \in \Theta} \{2\langle \eta \nabla f_n(\theta'), \theta - \theta' \rangle + \|\theta' - \theta\|^2 + \|\eta^2 \nabla f_n(\theta')\|^2\} \\ &= \arg \min_{\theta \in \Theta} \left\{ f_n(\theta') + \langle \nabla f_n(\theta'), \theta - \theta' \rangle + \frac{1}{2\eta} \|\theta' - \theta\|^2 \right\} \\ &= \arg \min_{\theta \in \Theta} h_n(\theta) \end{aligned}$$

where we define $h_n(\theta) := f_n(\theta') + \langle \nabla f_n(\theta'), \theta - \theta' \rangle + \frac{1}{2\eta} \|\theta' - \theta\|^2$. Note that $h(\theta)$ is $\frac{1}{\eta}$ -strongly convex. So by applying Lemma 1 to h_n and by the fact that $\hat{\theta}$ is the minimizer of h , we have

$$h_n(\hat{\theta}) \leq h_n(\theta_n^*) - \frac{1}{2\eta} \|\hat{\theta} - \theta_n^*\|^2$$

By the strong convexity of f_n it holds that $f_n(\theta') + \langle \nabla f_n(\theta'), \theta_n^* - \theta' \rangle \leq f_n(\theta_n^*) - \frac{\alpha}{2} \|\theta' - \theta_n^*\|^2$. By smoothness and the fact that $\eta < 1/\gamma$, we also have

$$f_n(\hat{\theta}) \leq f_n(\theta') + \langle \nabla f_n(\theta'), \hat{\theta} - \theta' \rangle + \frac{\gamma}{2} \|\theta' - \hat{\theta}\|^2 \leq h_n(\hat{\theta})$$

Combining these inequalities gives

$$f_n(\hat{\theta}) \leq f_n(\theta_n^*) - \frac{\alpha}{2} \|\theta' - \theta_n^*\|^2 + \frac{1}{2\eta} \|\theta' - \theta_n^*\|^2 - \frac{1}{2\eta} \|\hat{\theta} - \theta_n^*\|^2$$

By applying Lemma 1 again we have:

$$\frac{\alpha}{2} \|\hat{\theta} - \theta_n^*\|^2 \leq -\frac{\alpha}{2} \|\theta' - \theta_n^*\|^2 + \frac{1}{2\eta} \|\theta' - \theta_n^*\|^2 - \frac{1}{2\eta} \|\hat{\theta} - \theta_n^*\|^2$$

The result can be obtained by rearranging and aggregating the terms and then simplifying. \square

A.2 Proof of Lemma 5

The result of Lemma 5 is a slight modification from Theorem 3 from Zhang et al. (2017), the proof of which is reproduced in full here for completeness. Again before proving this theorem, we establish a crucial lemma. As in the proof of Theorem 3, we show a bound on the improvement from a single gradient step.

Proof. According to the update rule of multiple gradients we apply the result established in Lemma 4 K times which gives

$$\begin{aligned} \|\theta_{n+1} - \theta_n^*\|^2 &\leq \left(1 - \frac{2\alpha}{1/\eta + \alpha}\right)^K \|\theta_n - \theta_n^*\|^2 \\ &\leq \exp\left(-\frac{2\alpha K}{1/\eta + \alpha}\right) \|\theta_n - \theta_n^*\|^2 \\ &\leq \frac{1}{4} \|\theta_n - \theta_n^*\|^2 \end{aligned}$$

Then, as in Theorem 3, we bound the distances between the optimal parameters and the algorithm's parameters.

$$\begin{aligned} \sum_{n=1}^N \|\theta_n - \theta_n^*\|^2 &= \|\theta_1 - \theta_1^*\|^2 + \sum_{n=2}^N \|\theta_n - \theta_{n-1}^* + \theta_{n-1}^* - \theta_n^*\|^2 \\ &\leq \|\theta_1 - \theta_1^*\|^2 + 2 \sum_{n=2}^N \|\theta_n - \theta_{n-1}^*\|^2 + \|\theta_{n-1}^* - \theta_n^*\|^2 \\ &\leq \|\theta_1 - \theta_1^*\|^2 + \sum_{n=2}^N \frac{1}{2} \|\theta_{n-1} - \theta_{n-1}^*\|^2 + 2 \|\theta_{n-1}^* - \theta_n^*\|^2 \\ &\leq 2 \|\theta_1 - \theta_1^*\|^2 + 4 \sum_{n=2}^N \|\theta_{n-1}^* - \theta_n^*\|^2 \\ &\leq 2 \|\theta_1 - \theta_1^*\|^2 + 4S(\theta_{1:N}^*) \end{aligned}$$

Finally, by the smoothness of all f_n we have

$$\begin{aligned} \sum_{n=1}^N f_n(\theta_n) - f_n(\theta_n^*) &\leq \sum_{n=1}^N \frac{\gamma}{2} \|\theta_n - \theta_n^*\|^2 \\ &\leq 2\gamma S(\theta_{1:N}^*) + \gamma \|\theta_1 - \theta_1^*\|^2. \end{aligned}$$

\square