Proceedings of Machine Learning Research vol 120:1–12, 2020

Robust Regression for Safe Exploration in Control

Anqi Liu ANQILIU@CALTECH.EDU
Guanya Shi GSHI@CALTECH.EDU
Soon-Jo Chung SJCHUNG@CALTECH.EDU
Anima Anandkumar ANIMA@CALTECH.EDU
Yisong Yue YYUE@CALTECH.EDU
California Institute of Technology

Abstract

We study the problem of safe learning and exploration in sequential control problems. The goal is to safely collect data samples from operating in an environment, in order to learn to achieve a challenging control goal (e.g., an agile maneuver close to a boundary). A central challenge in this setting is how to quantify uncertainty in order to choose provably-safe actions that allow us to collect informative data and reduce uncertainty, thereby achieving both improved controller safety and optimality. To address this challenge, we present a deep robust regression model that is trained to directly predict the uncertainty bounds for safe exploration. We derive generalization bounds for learning, and connect them with safety and stability bounds in control. We demonstrate empirically that our robust regression approach can outperform conventional Gaussian process (GP) based safe exploration in settings where it is difficult to specify a good GP prior.

Keywords: Safe Exploration, Robust Regression, Covariate Shift, Generalization, Stability

1. Introduction

A key challenge in data-driven design for robotic controllers is automatically and safely collecting training data. Consider safely landing a drone at fast landing speeds (e.g., beyond a human expert's piloting abilities). The dynamics are both highly non-linear and poorly modeled as the drone approaches the ground (Cheeseman and Bennett, 1955), but such dynamics can be learnable given the appropriate training data (Shi et al., 2019). To collect such data autonomously, one must guarantee safety while operating in the environment, which is the problem of *safe exploration*. In the drone landing example, collecting informative training data requires the drone to land increasingly faster while not crashing. Figure 1 depicts an example, where the goal is to learn the most aggressive yet safe trajectory (orange), while not being overconfident and execute trajectories that crash (green); the initial nominal controller may only be able to execute very conservative trajectories (blue).

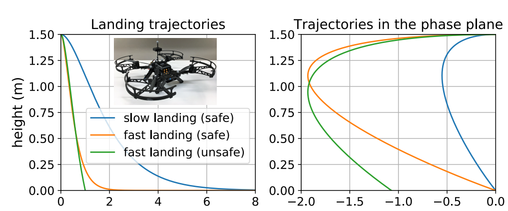

Figure 1: Fast drone landing

In order to safely collect such informative training data, we need to overcome two difficulties. First, we must quantify the learning errors in out-of-sample data. Every step of data collection creates a shift in the training data distribution. More specifically, our setting is an instance of covariate shift, where the underlying true physics stay constant, but the sampling of the state space is biased by the data collection (Chen et al., 2016). In order to leverage modern learning approaches, such as deep learning, we must reason about the impact of covariate shift when predicting on states not well represented by the training set. Second, we must reason about how to guarantee safety and stability when controlling using the current learned model. Our ultimate goal is to control the dynamical system with desired properties but staying safe and stable while data collection. The imperfect dynamical model's error translate to possible control error, which must be quantified and controlled.

Our Contributions. In this paper, we propose a deep robust regression approach for safe exploration in model-based control. We view exploration as a data shift problem, i.e., the "test" data in the proposed exploratory trajectory comes from a shifted distribution compared to the training set. Our approach explicitly learns to quantify uncertainty under such covariate shift, which we use to learn robust dynamics models to quantify uncertainty of entire trajectories for safe exploration.

We analyze learning performance from both generalization and data perturbation perspectives. We use our robust regression analysis to derive stability bounds for control performance when learning robust dynamics models, which is used for safe exploration. We empirically show that our approach outperforms conventional safe exploration approaches with much less tuning effort in two scenarios: (a) inverted pendulum trajectory tracking under wind disturbance; and (b) fast drone landing using an aerodynamics simulation based on real-world flight data (Shi et al., 2019).

2. Problem Setup

At a high level, our problem can be framed as a three-way interaction of: (i) learning the unmodeled, or residual, dynamics from collected data, (ii) determining whether the current learned dynamics model enables tracking a given trajectory within a safety set, and (iii) selecting trajectories for data collection that are both safe and informative, i.e., safe exploration. In the drone landing example in Figure 1, the residual dynamics is the ground effect that perturbs the nominal multi-rotor model, the safety set is not crashing into the ground, and safe exploration pertains to selecting the most aggressive landing trajectory that is provably safe with the current learned dynamics model.

A Mixed Model for Robotic Dynamics. We consider a standard mixed model for continuous robotic dynamics (Shi et al., 2019): $M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) - Bu = \underbrace{d(q,\dot{q})}_{\text{unknown}}$, with generalized coordinates $q \in \mathbb{R}^n$ (and their first & second time derivatives, $\dot{q}$ & $\ddot{q}$), control input $u \in \mathbb{R}^m$, inertia matrix $M(q) \in \mathbb{S}^n_{++}$, centrifugal and Coriolis terms $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$, gravitational forces $G \in \mathbb{R}^n$, actuation matrix $B \in \mathbb{R}^{n\times m}$ and some unknown residual dynamics $d \in \mathbb{R}^n$. Note that the C matrix is chosen to make $\dot{M} - 2C$ skew-symmetric from the relationship between the Riemannian metric $M(q)$ and Christoffel symbols. Here d is general, which potentially captures both parametric and nonparametric unmodeled terms. We aim to learn the unknown, or residual, dynamics $d(q,\dot{q})$ using machine learning models. The intuition behind this hybrid dynamical model is the sample efficiency of learning the residual should be much smaller than learning the whole model directly from data.

Model Based Nonlinear Control. To keep the ancillary design choices simple, we employ a standard nonlinear controller design (Shi et al., 2019). Define the reference trajectory as $\dot{q}_r = \dot{q}_g - \Lambda\tilde{q}$, where $\tilde{q} = q - q_g$, and the composite variable as $s = \dot{q} - \dot{q}_r = \dot{\tilde{q}} + \Lambda\tilde{q}$, where Λ is uniformly positive definite. The control objective is to drive s to 0 or a small error ball in the presence of bounded uncertainty. Assuming we had a good estimate $\hat{d}(q,\dot{q})$ of $d(q,\dot{q})$, then our controller is:

$$u = B^{\dagger}(M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r - Ks + G(q) - \hat{d}(q,\dot{q})), \tag{1}$$

where K is a uniformly positive definite matrix, and † denotes the Moore-Penrose pseudoinverse.

With the control law Eq. 1, we will have the following closed-loop dynamics:

$$\begin{bmatrix} M(q) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{s} \\ \dot{\tilde{q}} \end{bmatrix} + \begin{bmatrix} C(q,\dot{q})+K & 0 \\ -I & \Lambda \end{bmatrix} \begin{bmatrix} s \\ \tilde{q} \end{bmatrix} = \begin{bmatrix} d-\hat{d} \\ 0 \end{bmatrix} = \begin{bmatrix} \epsilon \\ 0 \end{bmatrix}. \tag{2}$$

where $\epsilon = d - \hat{d}$ is the approximation error between d and $\hat{d}$.

Safety Requirements. For any time-varying desired trajectory, $x_g(t) = [q_g(t), \dot{q}_g(t)]$, we must certify safety during trajectory tracking: $x(t) \in \mathfrak{S}, \forall t$, with high probability, where $\mathfrak{S}$ is some safety set. It is obvious that $x_g(t) \in \mathfrak{S}, \forall t$. However, because of unknown dynamics $d(q, \dot{q})$, the tracking error $\tilde{x}(t) \triangleq x(t) - x_g(t)$ may be large such that $\exists t, x(t) \notin \mathfrak{S}$. In the drone landing example in Figure 1, the safety set is that the vertical velocity at the point of landing should not exceed an upper limit (otherwise the drone is considered to have crash landed).

Safe Exploration. The ultimate goal is to identify a model (and accompanying controller) that can safely track trajectories with minimal cost. We assume that the cost function over trajectories is known (e.g., landing as quickly as possible), but certifying safety is difficult. The goal of safe exploration is then to select a trajectory to track that is both provably safe (with the current model) and leads to informative training data for improving safety certification. Our safe exploration procedure is thus to choose the lowest cost safe trajectory, which is a trajectory that lies at the boundary of the current safety set and closest to the overall minimal cost trajectory.

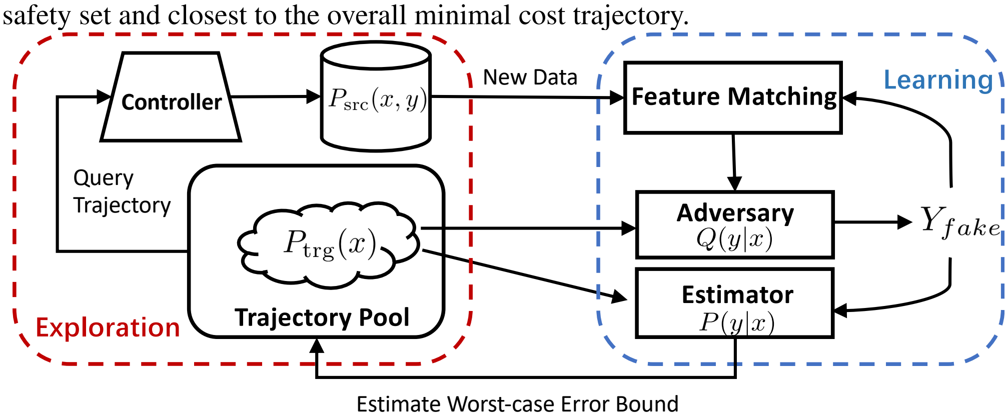

Figure 2: Our overall formulation. In the learning component, our estimator is robust to the worst-case model of the dynamics that is consistent with the observed source data, which we elaborate in Section 3. The learning and tracking error bound is then used for picking a trajectory that is safe if the worst case scenario is safe, whose details are in Section 4. With more source data, the worst-case model is constrained tighter along the exploration. We present the full algorithm in Section 5.

3. Learning Residual Dynamics as Robust Regression under Covariate Shift

Our learning problem is to estimate the residual dynamics $d(q, \dot{q})$ in a way that admits rigorous uncertainty estimates for safety certification. The key challenge is that the training data and test data are not sampled from the same distribution, which can be framed as covariate shift (Shimodaira, 2000). Covariate shift refers to distribution shift caused by the input variables $P(x)$, while keeping $P(y \equiv d(x)|x)$ fixed. In our motivating safe landing example, there is a universal "true" aerodynamics model, but we typically only observe training data from a limited source data distribution

$P_{\text{src}}(x)$. Certifying safety of a proposed trajectory will inevitably cover states that are not well-represented by the training, i.e., data from a target data distribution $P_{\text{trg}}(x)$. In other words, the distribution of states in a proposed trajectory is not the distribution states in the training data.

General intuition. We use robust regression (Chen et al., 2016) to estimate the residual dynamics under covariate shift. Robust regression is derived from a minimax estimation framework (Grünwald et al., 2004), where the estimator $P(y|x)$ tries to minimize a loss function on target data distribution $\mathcal{L}$, and the adversary $Q(y|x)$ tries to maximize the loss under source data constraints Γ:

$$\min_{P(y|x)} \max_{Q(y|x)\in\Gamma} \mathcal{L}. \tag{3}$$

Using the minimax framework, we achieve robustness to the worst-case possible conditional distribution that is "compatible" with finite training data if the estimator reaches the Nash equilibrium by minimizing a loss function defined on target data distribution.

Technical Design Choices. Our derivation hinges on a choice of loss function $\mathcal{L}$ and constraint set for the adversary Γ, from which one can derive a formal objective, a learning algorithm, and an uncertainty bound. See the full version for the complete details.[1] We use a relative loss function defined as the difference in conditional log-loss between an estimator $P(y|x)$ and a baseline conditional distribution $P_0(y|x)$ on the target data distribution $P_{\text{trg}}(x)P(y|x)$: relative loss $\mathcal{L} := \mathbb{E}_{P_{\text{trg}}(x)Q(y|x)}\left[-\log \frac{P(y|x)}{P_0(y|x)}\right]$. To construct the constraint set Γ, we utilize statistical properties of the source data distribution $P_{\text{src}}(x)$:

$$\Gamma := \{Q(y|x)||\mathbb{E}_{P_{\text{src}}(x)Q(y|x)}[\Phi(x,y)] - \mathbf{c}| \leq \lambda, \}$$

where $\phi(x,y)$ correspond to the sufficient statistics of the estimation task, and $\mathbf{c} = \frac{1}{n}\sum_{i=n}^{n}\Phi(x_i,y_i)$ is a vector of sample mean of statistics in the source data. This constraint means the adversary cannot choose a distribution whose sufficient statistics deviate too far from the collected training data.

The consequence of the above choices is that the solution has a parametric form: $P(y|x) \propto P_0(y|x)e^{\frac{P_{\text{src}}(x)}{P_{\text{trg}}(x)}\theta^T\Phi(x,y)}$. This form has two useful properties. First, it is straightforward to compute gradients on θ using only the training data. One can also train deep neural networks by treating $\Phi(x,y)$ as the last hidden layer, i.e, we learn a representation of the sufficient statistics. Second, this form yields a concrete uncertainty bound (see Section 4) that can be used to certify safety. For specific choices of P_0 and Φ, the uncertainty is Gaussian distributed, which can be useful for many stochastic control approaches that assume Gaussian uncertainty. [2].

4. From Learning Guarantees to Tracking Guarantees

We demonstrate that we bound the learning errors on possible target data and further bound the tracking error. We then apply the bound to certify safety. The proofs are in the full version.

Learning Guarantees. The learning performance of robust regression approach can be analyzed from two perspectives: generalization error under covaraite shift and perturbation error based on Lipschitz continuity. The generalization error reflects the expected error on a target distribution given certain function class, bounded distribution discrepancy, and base distribution. The perturbation error reflects the maximum error if target data deviates from training but stays in a Lipschitz ball. These error bounds are compatible with deep neural networks whose Rademacher complexity and Lipschitz constant can be controlled and measured (e.g., spectral-normalized neutral networks).

1. Full version: https://arxiv.org/abs/1906.05819.
2. Our method generalizes naturally to multidimensional output setting, where we predict a multidimensional Gaussian.

Theorem 1 *Assume S is a training set with i.i.d. data $x_i, ..., x_n$ sampled from $P_{src}(x)$, $\mathcal{F}$ is a regression function class satisfying $\sup_{x\in\mathcal{X},f,f'\in\mathcal{F}} |f(x) - f'(x)| \leq M$, $\hat{\mathfrak{R}}_S(\mathcal{F})$ is the Rademacher complexity on S, W is the upper bound of true density ratio $\sup_{x\sim P_{src}(x)} \frac{P_{trg}(x)}{P_{src}(x)} \leq W$, $\theta_y > 0$ is lower bounded by B, the weight estimation for the prediction $\hat{r}(x) = \frac{\hat{P}_{src}(x)}{\hat{P}_{trg}(x)}$ is lower bounded: $\inf_{x\in S} \hat{r}(x) \geq R$, base distribution variance is σ_0^2, and λ is the upperbound of all λ_i among the dimensions of $\phi(x)$. When learning a $\hat{f} \in \mathcal{F}$ on $P_{trg}(x, y)$, the following generalization error bound holds with probability at least $1 - \delta$,*

$$\mathbb{E}_{P_{trg}(x,y)}[(y - \hat{f}(x))^2] \leq W \left[(2RB + \sigma_0^{-2})^{-1} + \lambda + 4M\hat{\mathfrak{R}}_S(\mathcal{F}) + 3M^2\sqrt{\frac{\log \frac{2}{\delta}}{2n}} \right].$$

If we assume that target data samples x's stay in a ball $\mathbb{B}(\epsilon)$ with diameter ϵ from the source data S, $\mathbb{B}(\epsilon) = \{x| \sup_{x'\in S} \| x - x'\| \leq \epsilon\}$, the true function $f(x)$ is Lipschitz continuous with constant L, and the robust regression mean estimator $\hat{f}$ is also Lipschitz continuous with constant $\hat{L}$, then,

$$\sup_{x\in\mathbb{B}(\epsilon), y\sim f(x)} [(y - \hat{f}(x))^2] \leq ((2RB + \sigma_0^{-2})^{-1/2} + \sqrt{\lambda} + \left(L + \hat{L}\right) \|\epsilon\|)^2. \tag{4}$$

The density ratio W can be controlled by choosing the target distribution carefully in the safe exploration algorithm (Alg. 1). In other words, we can design the desired trajectories to be close enough to the training set so that the resulting tracking bounds are tight enough to guarantee safety.

Tracking Guarantees. We set $\|\epsilon\|^2 = \|y - \hat{f}(x)\|^2$ to correspond with the learning bounds. The target data is set to a single proposed trajectory $x_{\text{trg}}(t)$, which means W can be bounded. The second option is to use a perturbation bound, where $x_{\text{trg}}(t) \in \mathbb{B}(\epsilon)$. We emphasize that $\|\epsilon\|$ is upper bounded with $\|\epsilon\| \leq \sup_{x\in x_{\text{trg}}(t)} \|\epsilon(x)\|$ when we define target data in a specific set and use robust regression for learning dynamics. We show $\|x(t) - x_g(t)\| \triangleq \|\tilde{x}(t)\|$ (Euclidean distance between the desired trajectory and the real trajectory) is bounded when the error of the dynamics estimation is bounded. Again, recall that $x = [q, \dot{q}]$ is our state, and $x_g(t)$ is the desired trajectory.

Theorem 2 *Suppose x is in some compact set $\mathcal{X}$, and $\epsilon_m = \sup_{x\in\mathcal{X}} \|\epsilon\|$. Then $\tilde{x}$ will exponentially converge to the following ball: $\lim_{t\to\infty} \|\tilde{x}(t)\| = \gamma \cdot \epsilon_m$, where*

$$\gamma = \frac{\lambda_{\max}(M)}{\lambda_{\min}(K)\lambda_{\min}(M)} \sqrt{(\frac{1}{\lambda_{\min}(\Lambda)})^2 + (1 + \frac{\lambda_{\max}(\Lambda)}{\lambda_{\min}(\Lambda)})^2}, \tag{5}$$

where $\lambda_{\max}$ denotes the maximum eigenvalue and $\lambda_{\min}$ denotes the minimum eigenvalue.

Integration to safe exploration. We can integrate the bounds on learning error and tracking error into safe exploration. Specially, if we can design a compact set $\mathcal{X}$ and find the corresponding maximum error bound $\epsilon_m = \sup_{x\in\mathcal{X}} \|\epsilon\|$ on it, we can use it to decide whether a trajectory in this set is safe or not by checking whether its worst-case possible tracking trajectory is in the safety set $\mathfrak{S}$. Then we only pick the safe trajectories with the minimum cost in data collection.

5. Safe Exploration Algorithm

For simplicity, we maintain a finite set of candidate trajectories to select from for safe exploration; future work includes integration with continuous trajectory optimization (Nakka and Chung, 2019).

The worst-case tracking trajectories can be computed by generating a "tube" using euclidean distance in Theorem 2. We then eliminate unsafe ones and choose the most "aggressive" one in terms of our cost function for the next iteration. Instead of evaluating the actual upper bound, we use $\beta \cdot \max_x \sigma(x)$ for measuring ϵ_m as an approximation, since it is guaranteed that the error is within $\beta \cdot \max_x \sigma(x)$ with high probability as long as the prediction is a Gaussian distribution, if the true function is drawn from the same distribution. Here $\sigma(x)$ is the standard deviation of the Gaussian distribution predicted by our robust regression algorithm. Algorithm 1 describes this procedure.

6. Experiments

We conduct simulation experiments on the inverted pendulum and drone landing. We use kernel density estimation to estimate the density ratios. We demonstrate that our approach can reliably and safely converge to optimal behavior. We also compare with a Gaussian process (GP) version of Algorithm 1. In general, we find it is difficult to tune the GP kernel parameters, especially in the multidimensional output cases.

Algorithm 1 Safe Exploration for Control using Robust Dynamics Estimation

Input: Pool of desired trajectories $x_g^k(t), k = 1, 2, ..., K$; cost function J; robust regression model of dynamics f; controller U; safety set $\mathfrak{S}$; base distribution $\mathcal{N}(\mu_0, \sigma_0^2)$; parameter β
Initialize dynamics model $f_0 = \mathcal{N}(\mu_0, \sigma_0^2)$
Initialize training set = $\emptyset$, $f = f_0$
While $e = 1, ..., \mathcal{E}$
  Safe trajectory list $L = \emptyset$
  For $k = 1, ..., K$
    Predict $(\mu, \sigma^2) = f(x_g^k(t))$
    $\sigma_m = \max \sigma(x_g^k(t))$; $\epsilon_m = \beta \cdot \sigma_m$
    If worst-case trajectory in $\mathfrak{S}$
      Add $x_g^k(t)$ to L
  Track $x_g^*(t) = \arg\min_{x_g(t) \in L} J(x_g(t))$ to collect data $x'(t)$ using controller U
  Add data $x'(t)$ to Training set
  Train dynamics model f', $f = f'$
Output: dynamics model f, last desired trajectory $x_{\mathcal{E}}(t)$ and actual trajectory $x'_{\mathcal{E}}(t)$

Example 1 (inverted pendulum with external wind). Unlike the classical pendulum model, we consider unknown external wind. Dynamics can be described as $ml^2\ddot{q} - mlg \sin q - u = d(q, \dot{q})$, where $d(q, \dot{q})$ is external torque generated by the unknown wind. Our control goal is to track $q_g(t) = \sin(t)$, and the safety set is $\mathfrak{S} = \{(q, \dot{q}) : |q| < 1.5\}$.

We design a desired trajectory pool using $\mathcal{P}(C) = \{q_g(t) = C \cdot \sin(t), 0 < C \leq 1\}$. The ground truth of wind comes from quadratic air drag model. We use the angle upper bound in trajectory as the reward function for choosing "most aggressive" trajectories. We use base distribution $\mathcal{N}(0, 0.5)$ to start with and $\beta = 0.5$.

Example 2 (drone landing with ground effect) We consider drone landing with unknown ground effect. Dynamics is $m\ddot{q} + mg - c_T u^2 = d(q, \dot{q})$, where c_T is the thrust coefficient. The control goal is smooth and quick landing, i.e., quickly driving $q \to 0$. The safety set is $\mathfrak{S} = \{(q, \dot{q}) :$ when $q = 0, \dot{q} > -1\}$, i.e., the drone cannot hit the ground with high velocity. Our desired trajectory pool is $\mathcal{P}(C, h_g) = \{q_g(t) = e^{-Ct}(1+Ct)(1.5-h_g)+h_g, 0 < C, 0 \leq h_g < 1.5\}$, which means the drone smoothly moves from $z(0) = 1.5$ to the desired height h_d. If $h_d = 0$, the drone lands successfully. Greater C means faster landing. We use landing time to determine the next "aggressive" trajectory. The ground truth of aerodynamics comes from a dynamics simulator that is trained in (Shi et al., 2019), where $d(q, \dot{q})$ is a four-layer ReLU neural network trained by real flying data. We use base distribution $\mathcal{N}(0, 1)$ for robust regression and $\beta = 1$.

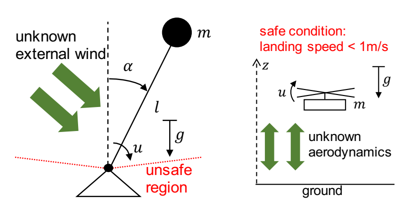

Figure 3: Illustration of two examples

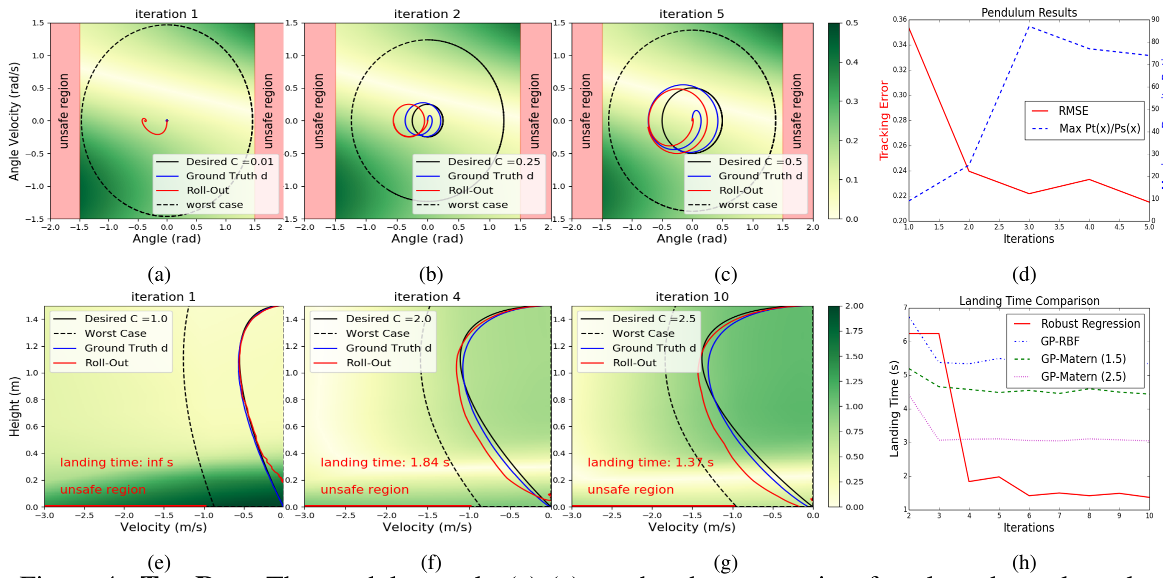

Figure 4: **Top Row.** The pendulum task: (a)-(c) are the phase portraits of angle and angular velocity; Blue curve is tracking the desired trajectory with ground-truth disturbance; the worst-case possible trajectory is calculated according to Theorem 2; heatmap is the difference between predicted dynamics (the wind) and the ground truth; and (d) is the tracking error and the maximum density ratio. **Bottom Row.** The drone landing task: (e)-(g) are the phase portraits with height and velocity; heatmap is difference between the predicted ground effect) and the ground truth; (h) is the comparison with GPs in landing time.

Result Analysis Figure 4(a) to (c) and (e) to (g) demonstrate the exploration process with selected desired trajectories, worst-case tracking trajectory under current dynamics model, tracking trajectories with the ground truth unknown dynamics model, and actual tracking trajectories. Note that for landing we learn three-dimensional ground effect where $d(q, \dot{q})$ corresponds to the z-component, while the trajectory design and error bound computation depend on z-component. In both tasks, the algorithm selects small C to guarantee safety at the beginning, and gradually is able to select larger C values and track it while staying safe. We also demonstrate the decaying tracking error in each iteration for the pendulum task in Figure 4 (d). We validate that our density ratio is always bounded along the exploration. The growing density ratio indicate that we are exploring more aggressively with more confident dynamics model. We examine the drone landing time for the landing task in Figure 4(h). We compare against multitask GP models (Bonilla et al., 2008) with both RBF kernel and Matern kernel. Our approach outperforms all GP models. Modeling the ground effect is notoriously challenging (Shi et al., 2019), and the GP suffers from model misspecification, especially in the multidimensional setting (Owhadi et al., 2015). Besides, GP models are also more computationally expensive than our method in making predictions. In contrast, our approach can fit general non-linear function estimators such as deep neural networks adaptively to the available data, which leads to more flexible inductive bias and better fitting of the data and uncertainty quantification.

7. Related Work

Safe Exploration. Most approaches for safe exploration use Gaussian processes (GPs) to quantify uncertainty (Sui et al., 2015, 2018; Kirschner et al., 2019; Akametalu et al., 2014; Berkenkamp et al., 2016; Turchetta et al., 2016; Wachi et al., 2018; Berkenkamp et al., 2017; Fisac et al., 2018; Khalil and Grizzle, 2002). These methods are related to bandit algorithms (Bubeck et al., 2012) and typically employ upper confidence bounds (Auer, 2002) to balance exploration versus exploitation (Srinivas et al., 2010). However, GPs are sensitive to model (i.e., the kernel) selection, and thus are often not suitable for tasks that aim to gradually reach boundaries of safety sets in a highly non-linear environment. In the high-dimensional case and under finite information, GPs suffer from bad priors even more severely (Owhadi et al., 2015). In the drone landing example, the fastest landing trajectory is one that is just barely safe. One could blend GP-based modeling with general function approximations (such as deep learning) (Berkenkamp et al., 2017; Cheng et al., 2019a), but the resulting optimization-based control problem can be challenging to solve. Other approaches either require having a safety model pre-specified upfront (Alshiekh et al., 2018), are restricted to relatively simple models (Moldovan and Abbeel, 2012), have no convergence guarantees during learning (Taylor et al., 2019), or have no safety guarantees (Garcia and Fernández, 2012).

Distribution Shift. The study of learning under distribution shift has seen increasing interest, owing to the widespread practical issue that test distributions rarely match the training distribution. Our work is stylistically similar to (Liu et al., 2015; Chen et al., 2016; Liu and Ziebart, 2014, 2017), which also frame uncertainty quantification through the lens of covariate shift, although ours is the first to extend to deep neural networks with rigorous guarantees. More broadly, dealing with domain shift is a fundamental challenge in deep learning, as highlighted by their vulnerability to adversarial inputs (Goodfellow et al., 2014), and the implied lack of robustness. Beyond robust estimation, the typical approaches are to either regularize (Srivastava et al., 2014; Wager et al., 2013; Le et al., 2016; Bartlett et al., 2017; Miyato et al., 2018; Shi et al., 2019; Benjamin et al., 2019; Cheng et al., 2019b) or synthesize an augmented dataset that anticipates the domain shift (Prest et al., 2012; Zheng et al., 2016; Stewart and Ermon, 2017). We also utilize the former approach by employing spectral normalization (Bartlett et al., 2017; Shi et al., 2019) in conjunction with robust estimation.

Robust and Adaptive Control. Robust control (Zhou and Doyle, 1998) and adaptive control (Slotine et al., 1991) are two classical frameworks to handle uncertainties in the dynamics. GPs have been combined with nonlinear MPC for online adaptation and uncertainty estimation (Ostafew et al., 2016). However, robust control suffers from large uncertainty set and it is hard to analyse convergence and quantify uncertainty in adaptive control. Ours is the first to explicitly consider covariate shift in learning. We pick the region to estimate uncertainty carefully and adapt the controller to track safe proposed trajectory in data collection.

8. Conclusion

In this paper, we propose an algorithmic framework for safe exploration in model-based control. To quantify uncertainty, we develop a robust deep regression method for dynamics estimation. Using robust regression, we explicitly deal with data shifts during episodic learning, and in particular can quantify uncertainty over entire trajectories. We prove the generalization and perturbation bounds for robust regression, and show how to integrate with control to derive safety bounds in terms of stability. These bounds explicitly translates the error in dynamics learning to the tracking error in control. From this, we design a safe exploration algorithm based on a finite pool of desired trajectories. We empirically show that our method achieves superior performance than GP-based methods in control of an inverted pendulum and drone landing examples

Acknowledgments

Anqi Liu is supported by PIMCO Postdoctoral Fellowship at Caltech. Prof. Anandkumar is supported by Bren endowed Chair, faculty awards from Microsoft, Google, and Adobe, DARPA PAI and LwLL grants. This work is also funded in part by Caltech's CAST and the Raytheon Company.

References

Anayo K Akametalu, Jaime F Fisac, Jeremy H Gillula, Shahab Kaynama, Melanie N Zeilinger, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control*, pages 1424–1431. IEEE, 2014.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6240–6249, 2017.

Ari S Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. In *International Conference on Learning Representations (ICLR)*, 2019.

Felix Berkenkamp, Angela P Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 491–496. IEEE, 2016.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, pages 908–918, 2017.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2008.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

IC Cheeseman and WE Bennett. The effect of ground on a helicopter rotor in forward flight. 1955.

Xiangli Chen, Mathew Monfort, Anqi Liu, and Brian D Ziebart. Robust covariate shift regression. In *Artificial Intelligence and Statistics*, pages 1270–1279, 2016.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Conference on Artificial Intelligence (AAAI)*, 2019a.

Richard Cheng, Abhinav Verma, Gabor Orosz, Swarat Chaudhuri, Yisong Yue, and Joel Burdick. Control regularization for reduced variance reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2019b.

Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 2018.

Javier Garcia and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Peter D Grünwald, A Philip Dawid, et al. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *the Annals of Statistics*, 32(4):1367–1433, 2004.

Hassan Khalil and Jessy Grizzle. Nonlinear systems. *Prentice hall*, 2002.

Johannes Kirschner, Mojmír Mutnỳ, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. In *International Conference on Machine Learning (ICML)*, 2019.

Hoang M. Le, Andrew Kang, Yisong Yue, and Peter Carr. Smooth imitation learning for online sequence prediction. In *International Conference on Machine Learning (ICML)*, 2016.

Anqi Liu and Brian Ziebart. Robust classification under sample selection bias. In *Advances in neural information processing systems*, pages 37–45, 2014.

Anqi Liu and Brian D Ziebart. Robust covariate shift prediction with general losses and feature views. *arXiv preprint arXiv:1712.10043*, 2017.

Anqi Liu, Lev Reyzin, and Brian D Ziebart. Shift-pessimistic active learning using robust bias-aware prediction. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. In *International Conference on Machine Learning (ICML)*, 2012.

Yashwanth Kumar Nakka and Soon-Jo Chung. Trajectory optimization for chance-constrained nonlinear stochastic systems. In *Conference on Decision and Control (CDC)*, 2019.

Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Robust constrained learning-based nmpc enabling reliable mobile robot path tracking. *The International Journal of Robotics Research*, 35(13):1547–1563, 2016.

Houman Owhadi, Clint Scovel, and Tim Sullivan. On the brittleness of bayesian inference. *SIAM Review*, 57(4):566–582, 2015.

Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3282–3289. IEEE, 2012.

Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. *International Conference on Robotics and Automation (ICRA)*, 2019.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with gaussian processes. In *International Conference on Machine Learning*, pages 997–1005, 2015.

Yanan Sui, Vincent Zhuang, Joel W Burdick, and Yisong Yue. Stagewise safe bayesian optimization with gaussian processes. In *International Conference on Machine Learning (ICML)*, 2018.

Andrew J Taylor, Victor D Dorobantu, Hoang M Le, Yisong Yue, and Aaron D Ames. Episodic learning with control lyapunov functions for uncertain robotic systems. *arXiv preprint arXiv:1903.01577*, 2019.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4312–4320, 2016.

Akifumi Wachi, Yanan Sui, Yisong Yue, and Masahiro Ono. Safe exploration and optimization of constrained mdps using gaussian processes. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in neural information processing systems*, pages 351–359, 2013.

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488, 2016.

Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.