Learning the Linear Quadratic Regulator from Nonlinear Observations

Zakaria Mhammedi   Dylan Foster   Max Simchowitz   Dipendra Misra   Wen Sun   Akshay Krishnamurthy   Alexander Rakhlin   John Langford

Tag Line

We develop **efficient algorithms** with provable sample complexity guarantees for **nonlinear control with rich observations**.

Overview

We propose a **new learning-theoretic framework** for rich observation continuous control in which the environment is summarized by a low dimensional continuous latent state, while the agent operates on high-dimensional observations.

We focus our attention on perhaps the simplest instantiation: **the rich observation linear quadratic regulator** (RichLQR), which posits that latent states evolve according to noisy linear equations and that each observation is associated with a latent state by an unknown nonlinear mapping.

We present the first algorithm RichID in this setting with a sample complexity guarantee that is **polynomial in the dimension of the latent space** and **independent of the observation space.**

The RichLQR Setting

RichLQR is a continuous control problem described by the following dynamics:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t, \qquad \mathbf{y}_t \sim q(\cdot \mid \mathbf{x}_t), \tag{1}$$

where $(\mathbf{x}_t) \subseteq \mathbb{R}^{d_\mathbf{x}}$, $(\mathbf{u}_t)$, $(\mathbf{w}_t)$, and $(\mathbf{y}_t) \subseteq \mathbb{R}^{d_\mathbf{y}}$ represent the states, actions, noise, and observations, respectively.

Observations. The learner does not directly observe $\mathbf{x}_t \in \mathbb{R}^{d_\mathbf{x}}$, instead sees observation $\mathbf{y}_t \in \mathbb{R}^{d_\mathbf{y}}$ drawn from an unknown **observation distribution** $q(\cdot \mid \mathbf{x}_t)$; it might be the case that $d_\mathbf{x} \gg d_\mathbf{y}$.

Goal. The aim is to choose a policy $\hat{\pi} = (\hat{\pi}_t)$ which selects $\mathbf{u}_t = \hat{\pi}_t(\mathbf{y}_0, \dots, \mathbf{y}_t)$ based on past and current observations to minimize

$$J_T(\pi) := \mathbb{E}_\pi\left[\frac{1}{T}\sum_{t=1}^T \mathbf{x}_t^\top Q \mathbf{x}_t + \mathbf{u}_t^\top R \mathbf{u}_t\right],$$

where $Q, R \succ 0$ are quadratic cost matrices.

Main Assumptions:

- **Perfect Decodability.** There exists a decoder $f_\star : \mathbb{R}^{d_\mathbf{y}} \to \mathbb{R}^{d_\mathbf{x}}$ such that $f_\star(y) = x$ for all $y \in \operatorname{supp} q(\cdot \mid x)$.
- **Realizability.** The learner's decoder class $\mathcal{F}$ contains the true decoder $f_\star$. (The function class $\mathcal{F}$ is used to decode observations.)
- **Noise Process.** We assume a Gaussian noise process; i.e. $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{\Sigma})$.

Is Perfect Decodability Necessary?

While perfect decodability may seem like a strong assumption, we show that without it, the problem can quickly become intractable. Consider the following variant of the model (1):

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t, \qquad \mathbf{y}_t = f_\star^{-1}(\mathbf{x}_t) + \varepsilon_t, \tag{2}$$

where ε_t is an independent **mean-zero** output noise; the presence of noise can break perfect-decodability in general.

Theorem 1 (informal). *Consider the dynamics (2) with $d_\mathbf{x} = d_\mathbf{y} = d_\mathbf{u} = T = 1$ and unit Gaussian noise. For every $\epsilon > 0$, the exists an $\mathcal{O}(\epsilon^{-1})$-Lipschitz decoder $f_\star$ and realizable function class $\mathcal{F}$ with $|\mathcal{F}| = 2$ s.t. any algorithm requires $\Omega(2^{(1/\epsilon)^{2/3}})$ trajectories to learn an ϵ-optimal decoder.*

Main Contribution

Our main contribution is a **new algorithmic principle**, Rich Iterative Decoding, or RichID, which solves the RichLQR problem with **sample complexity scaling polynomially** in the latent dimension $d_\mathbf{x}$ and $\ln|\mathcal{F}|$. Our main theorem is as follows:

Theorem 2 (Main theorem). *Under appropriate regularity conditions on the system parameters and noise process, RichID learns an ϵ-optimal policy $\hat{\pi} = (\hat{\pi}_t)_{t\in[T]}$ for horizon T using $C \cdot (d_\mathbf{x} + d_\mathbf{u})^1 6 T^4 \ln|\mathcal{F}|/\epsilon^6$ trajectories, where C is a problem-dependent constant.*

Algorithm Overview

Our algorithm RichID consists of three phases.

- In Phase I, we roll in with **Gaussian control inputs** and learn a good decoder $\hat{f}$ under this roll-in distribution by solving a certain regression problem involving our decoder class $\mathcal{F}$.
- In Phase II, we use the decoder $\hat{f}$ from Phase 1, **to learn a model** $(\widehat{A}, \widehat{B})$ for the system dynamics (up to a similarity transform M). Moreover, we can synthesize a controller $\widehat{K}$ so that the controller $\mathbf{u}_t = \widehat{K}\mathbf{x}_t$ is optimal for $(\widehat{A}, \widehat{B})$, and thus near-optimal for (A, B).
- In Phase III, we inductively solve a sequence of regression problems, one for each time $t = 0, \dots, T$, to learn a sequence of state decoders $(\hat{f}_t)$, s.t. for each t, $\hat{f}_t \approx f_\star$. Set $\hat{\pi} = (\hat{\pi}_t)$ with $\hat{\pi}_t \equiv \widehat{K}\hat{f}_t$.

Why Phase III?

Why not just execute the policy $\hat{\pi} \equiv \widehat{K}\hat{f}$ with $\hat{f}$ from Phase I? This decoder is "good" under the state distribution generated by taking **Gaussian random actions** $(\mathbf{u}_t)$, i.e. $\mathbb{E}_{\mathbf{u}_{1k}\sim\mathcal{N}(0,I)^k}\|\hat{f} - f_\star\| \le$ (small). This **does not imply** that $\mathbb{E}_{\hat{\pi}}\|\hat{f} - f_\star\| \le$ (small), which is what we want. So we learn $(\hat{f}_t)$ s.t. $\mathbb{E}_{\hat{\pi}_{1:t-1}}\|\hat{f}_t - f_\star\| \le$ (small), with $\hat{\pi}_t \equiv \widehat{K}\hat{f}_t$.

Phase I Overview

Goal. Learn an off-policy decoder $\hat{f}$.

- **Key idea** Predict random Gaussian actions $\mathbf{u}_k$ from observations. In particular, for an appropriate k, we solve the least squares problem:

$$\hat{f} \in \operatorname{argmin}_{f\in\mathcal{F}'} \sum_{i=1}^n \left\| f(\mathbf{y}_{k+1}^{(i)}) - \mathbf{u}_k^{(i)} \right\|^2,$$

where the superscript $^{(i)}$ denotes the trajectory index, and $\mathcal{F}'$ is an **augmented** class obtained from $\mathcal{F}$ through matrix multiplication.

- **Key result.** $\exists M$ **invertible** such that $\hat{f}(\mathbf{y}_{k+1}) \approx M\mathbf{x}_{k+1}$.

Question. Why is $\hat{f}(\mathbf{y}_{k+1}) \approx M\mathbf{x}_{k+1}$? There are two reasons for this:

1. As $n \to \infty$, we have

$$\frac{1}{n}\sum_{i=1}^n \left\| f(\mathbf{y}_{k+1}^{(i)}) - \mathbf{u}_k^{(i)} \right\|^2 \to \mathbb{E}\left[\|f(\mathbf{y}_{k+1}) - \mathbf{u}_k\|^2\right]$$

2. If $f_\star \in \operatorname{argmin}_{f\in\mathcal{F}'} \mathbb{E}[\|f(\mathbf{y}_{k+1}) - \mathbf{u}_k\|^2]$, then

$$f_\star(\mathbf{y}_{k+1}) \overset{(*)}{=} \mathbb{E}[\mathbf{u}_k|\mathbf{y}_{k+1}] \overset{(**)}{=} \mathbb{E}[\mathbf{u}_{k+1}|\mathbf{x}_{t+1}] \overset{(***)}{=} M\mathbf{x}_{k+1}.$$

(*) follows from the Bayes optimal solution of least squares: If $g_\star \in \operatorname{argmin}_g \mathbb{E}[\|g(Y) - U\|_2^2]$, then $g_\star(Y) = \mathbb{E}[U|Y]$.
(**) follows from the perfect decodability assumption.
(***) follows from properties of Gaussian conditional expectations: If $(U, X) \sim \mathcal{N}(0, \mathbf{\Sigma})$, then $\mathbb{E}[U|X] = \mathbf{\Sigma}_{ux}\mathbf{\Sigma}_{xx}^{-1}X$.

Phase II Overview

The decode $\hat{f}$ from Phase I gives an estimator of the **transformed state** $\tilde{\mathbf{x}} = M\mathbf{x}$, which follows the **modified** linear equations:

$$\tilde{\mathbf{x}}_{t+1} = MAM^\dagger\tilde{\mathbf{x}}_t + MB\mathbf{u}_t + M\mathbf{w}_t. \tag{3}$$

Using the decoder $\hat{f}$, we solve the following least squares problem to obtain estimates $(\widehat{A}, \widehat{B}) \approx (MAM^\dagger, MB)$ (recall that $\hat{f}(\mathbf{y}_k) \approx \tilde{\mathbf{x}}_k$):

$$(\widehat{A}, \widehat{B}) \in \operatorname{argmin}_{A',B'} \sum_{i=1}^n \left\| \hat{f}(\mathbf{y}_{k+1}^{(i)}) - A'\hat{f}(\mathbf{y}_k^{(i)}) - B'\mathbf{u}_k^{(i)} \right\|^2. \tag{4}$$

Phase III Overview

Goal: We will iteratively learn a sequence of decoders $(\hat{f}_t)$ s.t.

$$\mathbb{E}_{\hat{\pi}_{1:t-1}}\left[\left\|\hat{f}_t(\mathbf{y}_{0:t}) - f_\star(\mathbf{y}_t)\right\|^2\right] \le \text{(small)}, \forall t, \tag{5}$$

where $\hat{\pi}_s(\mathbf{y}_{0:s}) = \widehat{K}\hat{f}_s(\mathbf{y}_{0:s})$. This is **enough** by the **performance difference lemma**. See pre-print for more details on Phase III.