Planning with Learned Dynamics: Probabilistic Guarantees on Safety and Reachability via Lipschitz Constants

Devesh

December 19, 2024

1 Introduction

This paper addresses the challenge of planning and control with safety and reachability guarantees for systems with unknown dynamics. Traditional model-based control methods require precise dynamics models, which are often impractical due to unmodeled effects. Machine learning methods, while flexible, typically lack formal guarantees, making them unsuitable for safety-critical applications. The proposed method provides probabilistic guarantees on safety, reachability, and goal invariance by using the Lipschitz constant of the error between true and learned dynamics to determine the trustworthiness of the learned model. By estimating the Lipschitz constant using Extreme Value Theory and validating it with a Kolmogorov-Smirnov test, the method provides strong safety guarantees. It requires estimating three Lipschitz constants, resulting in safety and reachability guarantees with high probability. Additionally, conditions for the existence of a feedback controller that ensures tight tracking of planned trajectories are established. The contributions include a method to bound error using Lipschitz constants, conditions for feasible feedback laws, a planner with probabilistic guarantees, and evaluation on a 7DOF Kuka arm and a 6D quadrotor.

2 Prelims

2.1 Extreme Value Theory

Extreme Value Theory (EVT) is a branch of statistics that deals with the extreme deviations from the median of probability distributions. It is particularly useful for assessing the risk of rare events, such as financial crashes, natural disasters, or, in this context, the estimation of Lipschitz constants for model errors.

2.1.1 Generalized Extreme Value Distribution

The Generalized Extreme Value (GEV) distribution is used to model the distribution of the maximum (or minimum) of a large sample of independent and identically distributed random variables. The GEV distribution combines three types of distributions: Gumbel, Fréchet, and Weibull, and is defined as:

$$G(z;\mu,\sigma,\xi) = \exp\left\{-\left[1+\xi\left(\frac{z-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}$$

where:

- μ: Location parameter.
- σ: Scale parameter.
- ξ: Shape parameter.

2.1.2 Block Maxima Method

To apply EVT, the data is divided into blocks, and the maximum value within each block is recorded. These block maxima are then used to fit the GEV distribution. The steps are:

- Divide the data into non-overlapping blocks.
- Extract the maximum value from each block.
- Fit the GEV distribution to these maxima.

2.1.3 Peaks Over Threshold Method

Another approach is the Peaks Over Threshold (POT) method, which models the distribution of exceedances over a high threshold using the Generalized Pareto Distribution (GPD). The GPD is defined as:

$$G(z;\sigma,\xi) = 1 - \left(1 + \xi\frac{z}{\sigma}\right)^{-1/\xi}$$

where:

- σ: Scale parameter.
- ξ: Shape parameter.

2.1.4 Application to Lipschitz Constant Estimation

In the context of estimating Lipschitz constants, EVT can be used to model the distribution of the maximum errors between the true dynamics f and the learned dynamics g. By fitting a GEV or GPD to the maximum errors observed in the training data, we can estimate the tail behavior and obtain a probabilistic bound on the Lipschitz constant.

- Collect the errors $\|f(x_i, u_i) - g(x_i, u_i)\|$ for the training data.
- Apply the block maxima or POT method to extract extreme errors.
- Fit the GEV or GPD to these extreme errors.
- Use the fitted distribution to estimate a high quantile, providing a probabilistic bound on the Lipschitz constant.

2.1.5 Reverse Weibull Distribution

The Reverse Weibull distribution is one of the three types of distributions that make up the Generalized Extreme Value (GEV) distribution. It is used to model the minimum of a large sample of independent and identically distributed random variables. The Reverse Weibull distribution is defined as:

$$G(z;\mu,\sigma,\xi) = \exp\left\{-\left[1 - \xi\left(\frac{z-\mu}{\sigma}\right)\right]^{1/\xi}\right\}$$

where:

- μ: Location parameter.
- σ: Scale parameter.
- ξ: Shape parameter ($\xi < 0$ for the Reverse Weibull distribution).

The Reverse Weibull distribution is particularly useful for modeling the lower tail of the distribution, capturing the behavior of the smallest values in the dataset.

2.1.6 Fitting Reverse Weibull Distribution using MLE

Maximum Likelihood Estimation (MLE) is a method used to estimate the parameters of a statistical model. For the Reverse Weibull distribution, the parameters μ, σ, and ξ can be estimated using MLE by maximizing the likelihood function.
Given a dataset $\{z_1, z_2, \ldots, z_n\}$ of observed minima, the likelihood function for the Reverse Weibull distribution is:

$$L(\mu,\sigma,\xi) = \prod_{i=1}^{n} \frac{1}{\sigma}\left(1 - \xi\frac{z_i-\mu}{\sigma}\right)^{1/\xi-1} \exp\left\{-\left(1 - \xi\frac{z_i-\mu}{\sigma}\right)^{1/\xi}\right\}$$

The log-likelihood function is:

$$\ell(\mu,\sigma,\xi) = -n\log\sigma + \left(\frac{1}{\xi} - 1\right)\sum_{i=1}^{n}\log\left(1 - \xi\frac{z_i-\mu}{\sigma}\right) - \sum_{i=1}^{n}\left(1 - \xi\frac{z_i-\mu}{\sigma}\right)^{1/\xi}$$

To find the MLE estimates of μ, σ, and ξ, we solve the following optimization problem:

$$(\hat{\mu},\hat{\sigma},\hat{\xi}) = \arg\max_{\mu,\sigma,\xi} \ell(\mu,\sigma,\xi)$$

This optimization can be performed using numerical methods such as the Newton-Raphson method or gradient descent. Many statistical software packages provide built-in functions to perform MLE for various distributions, including the Reverse Weibull distribution.

2.1.7 Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (KS) test is a non-parametric test used to compare a sample distribution with a reference probability distribution or to compare two sample distributions. It quantifies the distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution. The KS statistic is defined as:

$$D = \sup_x |F_n(x) - F(x)|$$

where $F_n(x)$ is the empirical distribution function of the sample, and $F(x)$ is the reference cumulative distribution function. The null hypothesis is rejected if D is greater than a critical value, indicating that the sample does not follow the reference distribution.

2.1.8 P-values and Z-scores

Null Hypothesis: The null hypothesis is a general statement or default position that there is no relationship between two measured phenomena or no association among groups. In the context of the KS test, the null hypothesis states that the sample distribution follows the reference distribution. If the p-value is less than the chosen significance level (e.g., 0.05), we reject the null hypothesis, indicating that the sample distribution significantly differs from the reference distribution.
P-values: A p-value is a measure used in statistical hypothesis testing to determine the significance of the results. It represents the probability of obtaining test results at least as extreme as the observed results, assuming that the null hypothesis is true. A low p-value (typically less than 0.05) indicates that the observed data is unlikely under the null hypothesis, leading to the rejection of the null hypothesis.
Z-scores: A Z-score, also known as a standard score, measures how many standard deviations an element is from the mean of the distribution. It is calculated as:

$$Z = \frac{X - \mu}{\sigma}$$

where X is the value, μ is the mean, and σ is the standard deviation of the distribution. Z-scores are used to determine the position of a value within a distribution and to calculate p-values.
Calculating P-values from Z-scores: To calculate the p-value from a Z-score, we use the cumulative distribution function (CDF) of the standard normal distribution. The p-value for a given Z-score Z is:

$$p = 2\left(1 - \Phi(|Z|)\right)$$

where Φ is the CDF of the standard normal distribution. The factor of 2 accounts for the two-tailed nature of the test.
For example, if $Z = 1.96$, the p-value is:

$$p = 2\left(1 - \Phi(1.96)\right) \approx 0.05$$

indicating that there is a 5

2.2 Method

2.2.1 True and Approximate Dynamics

Let $f : X \times U \to X$ represent the true unknown dynamics, and $g : X \times U \to X$ be an approximation:

$$g(x, u) = g_0(x) + g_1(x)u.$$

The approximate dynamics are modeled with a neural network. Let $S = \{(x_i, u_i, f(x_i, u_i))\}_{i=1}^N$ be the training data, and $\Psi = \{(x_j, u_j, f(x_j, u_j))\}_{j=1}^M$ be samples to estimate the Lipschitz constant.

2.2.2 Lipschitz Constants

A Lipschitz constant L for a function h over a domain Z satisfies:

$$\|h(z_1) - h(z_2)\| \leq L\|z_1 - z_2\| \quad \forall z_1, z_2 \in Z.$$

Define L_{f-g}, L_{g_0}, and L_{g_1} as the smallest Lipschitz constants for $f - g$, g_0, and g_1.

2.2.3 Method Overview

The method has two components: determining a trusted domain $D \subseteq X \times U$ and estimating Lipschitz constants, and planning a safe and reachable path.
Problem 1. Given g, f, Ψ, and S, determine D where $\|f(x, u) - g(x, u)\| \leq \epsilon$ and estimate L_{f-g}, L_{g_0}, and L_{g_1}.
Problem 2. Given g, f, start x_I, goal x_G, tolerance λ, D, L_{f-g}, L_{g_0}, L_{g_1}, and X_{unsafe}, plan a trajectory $(x_0, \ldots, x_K)$, $(u_0, \ldots, u_{K-1})$ such that $x_0 = x_I$, $x_{k+1} = g(x_k, u_k)$, $K < \infty$, and $\|x_K - x_G\| \leq \lambda$. Ensure closed-loop execution under f avoids X_{unsafe}, converges to $B_{\epsilon+\lambda}(x_G)$, and remains in $B_{\epsilon+\lambda}(x_G)$ after reaching x_K.

2.2.4 Trusted Domain

Consider a single training point $(\bar{x}, \bar{u})$ and a novel point (x, u). We derive a bound on the error between the true and estimated dynamics at (x, u) using the triangle inequality and Lipschitz constant of the error:

$$\|f(x,u) - g(x,u)\| = \|f(x,u) - g(x,u) - f(\bar{x},\bar{u}) + g(\bar{x},\bar{u}) + f(\bar{x},\bar{u}) - g(\bar{x},\bar{u})\|$$

$(\bar{x}, \bar{u})$ to the novel point (x, u).

$$\leq L_{f-g}\|(x,u) - (\bar{x},\bar{u})\| + \|f(\bar{x},\bar{u}) - g(\bar{x},\bar{u})\|.$$

$\|f(x,u) - g(x,u) - f(\bar{x},\bar{u}) + g(\bar{x},\bar{u})\|$ measures how the error grows when moving from the training point. $\|f(\bar{x},\bar{u}) - g(\bar{x},\bar{u})\|$ is the model error.

The above relation describes the error at a novel point, but we can also generalize to any domain D. Define b_T to be the dispersion of $S \cap D$ in D and define e_T to be the maximum training error of the learned model. Explicitly,

$$b_T = \sup_{(x,u)\in D} \min_{(\bar{x},\bar{u})\in S\cap D} \|(x,u) - (\bar{x},\bar{u})\|,$$

$$e_T = \max_{(\bar{x},\bar{u})\in S\cap D} \|f(\bar{x},\bar{u}) - g(\bar{x},\bar{u})\|.$$

b_T is the largest possible value of the minimum distance between (x, u) in $\mathcal{D}$, and its closest training point in $(\bar{x}, \bar{u})$ in $S \cap D$. e_T is the maximum error in the training data.

Then, we can uniformly bound the error across the entire set D to yield a simple and exact relation between f and g:

$$\epsilon = L_{f-g} b_T + e_T.$$

$$\forall (x,u) \in D, \quad f(x,u) = g(x,u) + \delta, \quad \|\delta\| \leq \epsilon.$$

For the remainder of the method, we select D to be the union of r-balls about a subset of the training data $S_D \subseteq S$:

$$D = \bigcup_{(\bar{x},\bar{u})\in S_D} B_r(\bar{x},\bar{u}).$$

In the next section, we discuss the selection of S_D, its role in estimating L_{f-g}, and how r is selected.

2.2.5 Estimating the Lipschitz Constants

To estimate the Lipschitz constants L_{f-g}, L_{g_0}, and L_{g_1}, we use Extreme Value Theory (EVT) and the Reverse Weibull distribution. The steps are as follows:

1. **Collect Slopes:** For each pair of points (z_1, z_2) in the dataset Ψ, compute the slope:

$$s_i = \frac{\|h(z_1^i) - h(z_2^i)\|}{\|z_1^i - z_2^i\|}$$

   where h is the function for which we are estimating the Lipschitz constant.

2. **Fit Reverse Weibull Distribution:** Use Maximum Likelihood Estimation (MLE) to fit the Reverse Weibull distribution to the collected slopes. The Cumulative Distribution Function (CDF) for the Reverse Weibull distribution is:

$$F_W(w) = \begin{cases} \exp\left(-\left(\frac{\gamma - w}{\alpha}\right)^\beta\right) & \text{if } w < \gamma \\ 1 & \text{if } w \geq \gamma \end{cases}$$

   where γ is the location parameter (represents the upper bound), α is the scale parameter, and β is the shape parameter.

3. **Validate with KS Test:** Perform the Kolmogorov-Smirnov (KS) test to validate the fit of the Reverse Weibull distribution to the slopes. The KS statistic is:

$$D = \sup_x |F_n(x) - F(x)|$$

   where $F_n(x)$ is the empirical distribution function of the sample, and $F(x)$ is the fitted CDF. A p-value greater than 0.05 indicates a good fit.

4. **Compute Confidence Interval:** Calculate the confidence interval for the location parameter γ to account for statistical uncertainty. The confidence interval c is given by:

$$c = \Phi^{-1}(\rho)\xi$$

where $\Phi^{-1}(\rho)$ is the Z-score for the desired confidence level ρ (e.g., $\rho = 0.95$ corresponds to 1.96), and ξ is the standard error of the γ estimate. The standard error measures the uncertainty in the estimate of γ and is calculated during the MLE process.

5. **Estimate Lipschitz Constant:** The final estimate for the Lipschitz constant is:

$$\hat{L}_h = \hat{\gamma} + c$$

This ensures $\hat{L}_h \geq L_h$ with probability ρ.

Filtering Procedure The goal is to choose D large enough for planning while keeping L_{f-g} small (further away from the training data but with less error). To achieve this, a filtering procedure is used to reduce the impact of outliers in S.
Let μ and σ be the mean and standard deviation of the error over S. Define:

$$S_D = \{(\bar{x}, \bar{u}) \in S \mid \|f(\bar{x}, \bar{u}) - g(\bar{x}, \bar{u})\| \leq \mu + a\sigma\}$$

where a is a user-defined parameter.

2.2.6 Algorithm 2: Selecting r and D

1. **Input:** $\mu, \sigma, a, S_D, \Psi, \alpha > 0$
2. $r \leftarrow \mu + a\sigma$
3. **while** True **do**
   (a) construct D using equation (8)
   (b) estimate L_{f-g} using Algorithm 1 and Ψ
   (c) calculate ϵ using equation (6)
   (d) **if** $L_{f-g} \geq 1$ **then** return failure
   (e) **if** $r > \epsilon$ **then** return r and D
   (f) **else** $r \leftarrow \epsilon + \alpha$ // α is a small constant

Algorithm 2 is run to grow D by proposing values of r, estimating L_{f-g}, and increasing r until $r > \epsilon$ or $L_{f-g} \geq 1$. Finding D with $r > \epsilon$ and $L_{f-g} < 1$ is useful for planning.
In Euclidean spaces, $r \geq b_T$. If no filtering is done, $r = b_T$, since no point in D is further than a distance r from S_D and the furthest any point in D can lie from a point in S_D is r; however, filtering shrinks D and thus decreases the dispersion, making it possible that $r \geq b_T$. The parameter a should be chosen to balance the size of D against the magnitude of L_{f-g}, which we tune heuristically.
This filtering lets us exclude regions where our learned model is less accurate, yielding smaller e_T. Note that filtering does not affect the i.i.d. property of the samples needed for Algorithm 1; it only applies a mask to the domain. We also note that Algorithm 2 returns a minimum value for r, but a larger r can be chosen as long as L_{f-g} is estimated with Algorithm 1. A larger r makes planning easier by expanding the trusted domain.
While we never explicitly address the assumption that the true dynamics are deterministic, the estimated Lipschitz constant may be unbounded in the stochastic case, such as when two samples have the same inputs but different outputs due to noise, causing a division by 0 in line 3 of Algorithm 1. L_{g_0} and L_{g_1} may also be estimated with Algorithm 1, which we employ in the results. Alternatively, other methods can give tight upper bounds on the Lipschitz constant of neural networks, though they may not scale to the networks used in our results. Future work will examine approaches that improve scalability at the cost of looser Lipschitz upper bounds.

3 Questions

- How do you sample around the trained data?
- How do you get the true dynamics?

3.0.1 Planning