
Lipschitz Continuity in Model-based Reinforcement Learning

Kavosh Asadi^{*1} Dipendra Misra^{*2} Michael L. Littman¹

Abstract

We examine the impact of learning Lipschitz continuous models in the context of model-based reinforcement learning. We provide a novel bound on multi-step prediction error of Lipschitz models where we quantify the error using the Wasserstein metric. We go on to prove an error bound for the value-function estimate arising from Lipschitz models and show that the estimated value function is itself Lipschitz. We conclude with empirical results that show the benefits of controlling the Lipschitz constant of neural-network models.

1. Introduction

The model-based approach to reinforcement learning (RL) focuses on predicting the dynamics of the environment to plan and make high-quality decisions (Kaelbling et al., 1996; Sutton & Barto, 1998). Although the behavior of model-based algorithms in tabular environments is well understood and can be effective (Sutton & Barto, 1998), scaling up to the approximate setting can cause instabilities. Even small model errors can be magnified by the planning process resulting in poor performance (Talvitie, 2014).

In this paper, we study model-based RL through the lens of Lipschitz continuity, intuitively related to the smoothness of a function. We show that the ability of a model to make accurate multi-step predictions is related to the model’s one-step accuracy, but also to the magnitude of the Lipschitz constant (smoothness) of the model. We further show that the dependence on the Lipschitz constant carries over to the value-prediction problem, ultimately influencing the quality of the policy found by planning.

We consider a setting with continuous state spaces and stochastic transitions where we quantify the distance between distributions using the Wasserstein metric. We

^{*}Equal contribution ¹Department of Computer Science, Brown University, Providence, USA ²Department of Computer Science and Cornell Tech, Cornell University, New York, USA. Correspondence to: Kavosh Asadi <kavosh@brown.edu>.

introduce a novel characterization of models, referred to as a Lipschitz model class, that represents stochastic dynamics using a set of component deterministic functions. This allows us to study any stochastic dynamic using the Lipschitz continuity of its component deterministic functions. To learn a Lipschitz model class in continuous state spaces, we provide an Expectation-Maximization algorithm (Dempster et al., 1977).

One promising direction for mitigating the effects of inaccurate models is the idea of limiting the complexity of the learned models or reducing the horizon of planning (Jiang et al., 2015). Doing so can sometimes make models more useful, much as regularization in supervised learning can improve generalization performance (Tibshirani, 1996). In this work, we also examine a type of regularization that comes from controlling the Lipschitz constant of models. This regularization technique can be applied efficiently, as we will show, when we represent the transition model by neural networks.

2. Background

We consider the Markov decision process (MDP) setting in which the RL problem is formulated by the tuple $\langle \mathcal{S}, \mathcal{A}, R, T, \gamma \rangle$. Here, by \mathcal{S} we mean a continuous state space and by \mathcal{A} we mean a discrete action set. The functions $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ and $T : \mathcal{S} \times \mathcal{A} \rightarrow \text{Pr}(\mathcal{S})$ denote the reward and transition dynamics. Finally, $\gamma \in [0, 1)$ is the discount rate. If $|\mathcal{A}| = 1$, the setting is called a Markov reward process (MRP).

2.1. Lipschitz Continuity

Our analyses leverage the “smoothness” of various functions, quantified as follows.

Definition 1. Given two metric spaces (M_1, d_1) and (M_2, d_2) consisting of a space and a distance metric, a function $f : M_1 \mapsto M_2$ is Lipschitz continuous (sometimes simply Lipschitz) if the Lipschitz constant, defined as

$$K_{d_1, d_2}(f) := \sup_{s_1 \in M_1, s_2 \in M_1} \frac{d_2(f(s_1), f(s_2))}{d_1(s_1, s_2)}, \quad (1)$$

is finite.

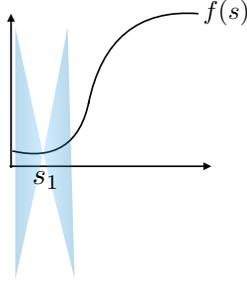


Figure 1. An illustration of Lipschitz continuity. Pictorially, Lipschitz continuity ensures that f lies in between the two affine functions (colored in blue) with slopes K and $-K$.

Equivalently, for a Lipschitz f ,

$$\forall s_1, \forall s_2 \quad d_2(f(s_1), f(s_2)) \leq K_{d_1, d_2}(f) d_1(s_1, s_2) .$$

The concept of Lipschitz continuity is visualized in Figure 1.

A Lipschitz function f is called a *non-expansion* when $K_{d_1, d_2}(f) = 1$ and a *contraction* when $K_{d_1, d_2}(f) < 1$. Lipschitz continuity, in one form or another, has been a key tool in the theory of reinforcement learning (Bertsekas, 1975; Bertsekas & Tsitsiklis, 1995; Littman & Szepesvári, 1996; Müller, 1996; Ferns et al., 2004; Hinderer, 2005; Rachelson & Lagoudakis, 2010; Szepesvári, 2010; Pazis & Parr, 2013; Pirota et al., 2015; Pires & Szepesvári, 2016; Berkenkamp et al., 2017; Bellemare et al., 2017) and bandits (Kleinberg et al., 2008; Bubeck et al., 2011). Below, we also define Lipschitz continuity over a subset of inputs.

Definition 2. A function $f : M_1 \times \mathcal{A} \mapsto M_2$ is uniformly Lipschitz continuous in \mathcal{A} if

$$K_{d_1, d_2}^{\mathcal{A}}(f) := \sup_{a \in \mathcal{A}} \sup_{s_1, s_2} \frac{d_2(f(s_1, a), f(s_2, a))}{d_1(s_1, s_2)} , \quad (2)$$

is finite.

Note that the metric d_1 is defined only on M_1 .

2.2. Wasserstein Metric

We quantify the distance between two distributions using the following metric:

Definition 3. Given a metric space (M, d) and the set $\mathbb{P}(M)$ of all probability measures on M , the Wasserstein metric (or the 1st Kantorovich metric) between two probability distributions μ_1 and μ_2 in $\mathbb{P}(M)$ is defined as

$$W(\mu_1, \mu_2) := \inf_{j \in \Lambda} \int \int j(s_1, s_2) d(s_1, s_2) ds_2 ds_1 , \quad (3)$$

where Λ denotes the collection of all joint distributions j on $M \times M$ with marginals μ_1 and μ_2 (Vaserstein, 1969).

Sometimes referred to as “Earth Mover’s distance”, Wasserstein is the minimum expected distance between pairs of points where the joint distribution j is constrained to match the marginals μ_1 and μ_2 . New applications of this metric are discovered in machine learning, namely in the context of generative adversarial networks (Arjovsky et al., 2017) and value distributions in reinforcement learning (Bellemare et al., 2017).

Wasserstein is linked to Lipschitz continuity using duality:

$$W(\mu_1, \mu_2) = \sup_{f: K_{d, d_{\mathbb{R}}}(f) \leq 1} \int (f(s)\mu_1(s) - f(s)\mu_2(s)) ds . \quad (4)$$

This equivalence, known as Kantorovich-Rubinstein duality (Villani, 2008), lets us compute Wasserstein by maximizing over a Lipschitz set of functions $f : \mathcal{S} \mapsto \mathbb{R}$, a relatively easier problem to solve. In our theory, we utilize both definitions, namely the primal definition (3) and the dual definition (4).

3. Lipschitz Model Class

We introduce a novel representation of stochastic MDP transitions in terms of a distribution over a set of deterministic components.

Definition 4. Given a metric state space $(\mathcal{S}, d_{\mathcal{S}})$ and an action space \mathcal{A} , we define F_g as a collection of functions: $F_g = \{f : \mathcal{S} \mapsto \mathcal{S}\}$ distributed according to $g(f | a)$ where $a \in \mathcal{A}$. We say that F_g is a Lipschitz model class if

$$K_F := \sup_{f \in F_g} K_{d_{\mathcal{S}}, d_{\mathcal{S}}}(f) ,$$

is finite.

Our definition captures a subset of stochastic transitions, namely ones that can be represented as a state-independent distribution over deterministic transitions. An example is provided in Figure 2. We further prove in the appendix (see Claim 1) that any finite MDP transition probabilities can be decomposed into a state-independent distribution g over a finite set of deterministic functions f .

Associated with a Lipschitz model class is a transition function given by:

$$\widehat{T}(s' | s, a) = \sum_f \mathbb{1}(f(s) = s') g(f | a) .$$

Given a state distribution $\mu(s)$, we also define a generalized notion of transition function $\widehat{T}_{\mathcal{G}}(\cdot | \mu, a)$ given by:

$$\widehat{T}_{\mathcal{G}}(s' | \mu, a) = \int_s \underbrace{\sum_f \mathbb{1}(f(s) = s') g(f | a)}_{\widehat{T}(s' | s, a)} \mu(s) ds .$$

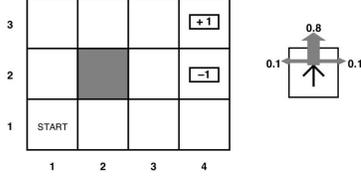


Figure 2. An example of a Lipschitz model class in a gridworld environment (Russell & Norvig, 1995). The dynamics are such that any action choice results in an attempted transition in the corresponding direction with probability 0.8 and in the neighboring directions with probabilities 0.1 and 0.1. We can define $F_g = \{f^{up}, f^{right}, f^{down}, f^{left}\}$ where each f outputs a deterministic next position in the grid (factoring in obstacles). For $a = up$, we have: $g(f^{up} | a = up) = 0.8$, $g(f^{right} | a = up) = g(f^{left} | a = up) = 0.1$, and $g(f^{down} | a = up) = 0$. Defining distances between states as their Manhattan distance in the grid, then $\forall f \sup_{s_1, s_2} (d(f(s_1), f(s_2)) / d(s_1, s_2)) = 2$, and so $K_F = 2$. So, the four functions and g comprise a Lipschitz model class.

We are primarily interested in $K_{d,d}^A(\widehat{T}_G)$, the Lipschitz constant of \widehat{T}_G . However, since \widehat{T}_G takes as input a probability distribution and also outputs a probability distribution, we require a notion of distance between two distributions. This notion is quantified using Wasserstein and is justified in the next section.

4. On the Choice of Probability Metric

We consider the stochastic model-based setting and show through an example that the Wasserstein metric is a reasonable choice compared to other common options.

Consider a uniform distribution over states $\mu(s)$ as shown in black in Figure 3 (top). Take a transition function T_G in the environment that, given an action a , uniformly randomly adds or subtracts a scalar c_1 . The distribution of states after one transition is shown in red in Figure 3 (middle). Now, consider a transition model \widehat{T}_G that approximates T_G by uniformly randomly adding or subtracting the scalar c_2 . The distribution over states after one transition using this imperfect model is shown in blue in Figure 3 (bottom). We desire a metric that captures the similarity between the output of the two transition functions. We first consider Kullback-Leibler (KL) divergence and observe that:

$$KL(T_G(\cdot | \mu, a), \widehat{T}_G(\cdot | \mu, a)) := \int T_G(s' | \mu, a) \log \frac{T_G(s' | \mu, a)}{\widehat{T}_G(s' | \mu, a)} ds' = \infty,$$

unless the two constants are exactly the same.

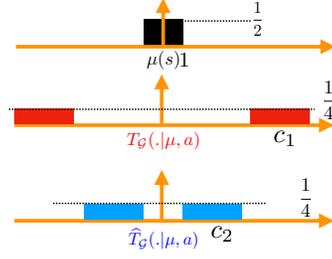


Figure 3. A state distribution $\mu(s)$ (top), a stochastic environment that randomly adds or subtracts c_1 (middle), and an approximate transition model that randomly adds or subtracts a second scalar c_2 (bottom).

The next possible choice is Total Variation (TV) defined as:

$$TV(T_G(\cdot | \mu, a), \widehat{T}_G(\cdot | \mu, a)) := \frac{1}{2} \int |T_G(s' | \mu, a) - \widehat{T}_G(s' | \mu, a)| ds' = 1,$$

if the two distributions have disjoint supports regardless of how far the supports are from each other.

In contrast, Wasserstein is sensitive to how far the constants are as:

$$W(T_G(\cdot | \mu, a), \widehat{T}_G(\cdot | \mu, a)) = |c_1 - c_2|.$$

It is clear that, of the three, Wasserstein corresponds best to the intuitive sense of how closely T_G approximates \widehat{T}_G . This is particularly important in high-dimensional spaces where the true distribution is known to usually lie in low-dimensional manifolds. (Narayanan & Mitter, 2010)

5. Understanding the Compounding Error Phenomenon

To extract a prediction with a horizon $n > 1$, model-based algorithms typically apply the model for n steps by taking the state input in step t to be the state output from the step $t - 1$. Previous work has shown that model error can result in poor long-horizon predictions and ineffective planning (Talvitie, 2014; 2017). Observed even beyond reinforcement learning (Lorenz, 1972; Venkatraman et al., 2015), this is referred to as the *compounding error phenomenon*. The goal of this section is to provide a bound on multi-step prediction error of a model. We formalize the notion of model accuracy below:

Definition 5. Given an MDP with a transition function T , we identify a Lipschitz model F_g as Δ -accurate if its induced \widehat{T} satisfies:

$$\forall s \forall a \quad W(\widehat{T}(\cdot | s, a), T(\cdot | s, a)) \leq \Delta.$$

We want to express the multi-step Wasserstein error in terms of the single-step Wasserstein error and the Lipschitz constant of the transition function \widehat{T}_G . We provide a bound on the Lipschitz constant of \widehat{T}_G using the following lemma:

Lemma 1. *A generalized transition function \widehat{T}_G induced by a Lipschitz model class F_G is Lipschitz with a constant:*

$$K_{W,W}^A(\widehat{T}_G) := \sup_a \sup_{\mu_1, \mu_2} \frac{W(\widehat{T}_G(\cdot | \mu_1, a), \widehat{T}_G(\cdot | \mu_2, a))}{W(\mu_1, \mu_2)} \leq K_F$$

Intuitively, Lemma 1 states that, if the two input distributions are similar, then for any action the output distributions given by \widehat{T}_G are also similar up to a K_F factor. We prove this lemma, as well as the subsequent lemmas, in the appendix.

Given the one-step error (Definition 5), a start state distribution μ and a fixed sequence of actions a_0, \dots, a_{n-1} , we desire a bound on n -step error:

$$\delta(n) := W(\widehat{T}_G^n(\cdot | \mu), T_G^n(\cdot | \mu)),$$

where $\widehat{T}_G^n(\cdot | \mu) := \underbrace{\widehat{T}_G(\cdot | \widehat{T}_G(\cdot | \dots \widehat{T}_G(\cdot | \mu, a_0) \dots, a_{n-2}), a_{n-1})}_{n \text{ recursive calls}}$

and $T_G^n(\cdot | \mu)$ is defined similarly. We provide a useful lemma followed by the theorem.

Lemma 2. (Composition Lemma) *Define three metric spaces (M_1, d_1) , (M_2, d_2) , and (M_3, d_3) . Define Lipschitz functions $f : M_2 \mapsto M_3$ and $g : M_1 \mapsto M_2$ with constants $K_{d_2, d_3}(f)$ and $K_{d_1, d_2}(g)$. Then, $h : f \circ g : M_1 \mapsto M_3$ is Lipschitz with constant $K_{d_1, d_3}(h) \leq K_{d_2, d_3}(f)K_{d_1, d_2}(g)$.*

Similar to composition, we can show that summation preserves Lipschitz continuity with a constant bounded by the sum of the Lipschitz constants of the two functions. We omitted this result due to brevity.

Theorem 1. *Define a Δ -accurate \widehat{T}_G with the Lipschitz constant K_F and an MDP with a Lipschitz transition function T_G with constant K_T . Let $\bar{K} = \min\{K_F, K_T\}$. Then $\forall n \geq 1$:*

$$\delta(n) := W(\widehat{T}_G^n(\cdot | \mu), T_G^n(\cdot | \mu)) \leq \Delta \sum_{i=0}^{n-1} (\bar{K})^i.$$

Proof. We construct a proof by induction. Using Kantorovich-Rubinstein duality (Lipschitz property of f not shown for brevity) we first prove the base of induction:

$$\begin{aligned} \delta(1) &:= W(\widehat{T}_G(\cdot | \mu, a_0), T_G(\cdot | \mu, a_0)) \\ &:= \sup_f \int \int (\widehat{T}(s' | s, a_0) - T(s' | s, a_0)) f(s') \mu(s) ds ds' \\ &\leq \int \sup_f \int (\widehat{T}(s' | s, a_0) - T(s' | s, a_0)) f(s') ds' \mu(s) ds \\ &\quad = W(\widehat{T}(\cdot | s, a_0), T(\cdot | s, a_0)) \text{ due to duality (4)} \\ &= \int \underbrace{W(\widehat{T}(\cdot | s, a_0), T(\cdot | s, a_0))}_{\leq \Delta \text{ due to Definition 5}} \mu(s) ds \\ &\leq \int \Delta \mu(s) ds = \Delta. \end{aligned}$$

We now prove the inductive step. Assuming $\delta(n-1) := W(\widehat{T}_G^{n-1}(\cdot | \mu), T_G^{n-1}(\cdot | \mu)) \leq \Delta \sum_{i=0}^{n-2} (K_F)^i$ we can write:

$$\begin{aligned} \delta(n) &:= W(\widehat{T}_G^n(\cdot | \mu), T_G^n(\cdot | \mu)) \\ &\leq W(\widehat{T}_G^n(\cdot | \mu), \widehat{T}_G(\cdot | T_G^{n-1}(\cdot | \mu), a_{n-1})) \\ &\quad + W(\widehat{T}_G(\cdot | T_G^{n-1}(\cdot | \mu), a_{n-1}), T_G^n(\cdot | \mu)) \text{ (Triangle ineq)} \\ &= W(\widehat{T}_G(\cdot | \widehat{T}_G^{n-1}(\cdot | \mu), a_{n-1}), \widehat{T}_G(\cdot | T_G^{n-1}(\cdot | \mu), a_{n-1})) \\ &\quad + W(\widehat{T}_G(\cdot | T_G^{n-1}(\cdot | \mu), a_{n-1}), T_G(\cdot | T_G^{n-1}(\cdot | \mu), a_{n-1})) \end{aligned}$$

We now use Lemma 1 and Definition 5 to upper bound the first and the second term of the last line respectively.

$$\begin{aligned} \delta(n) &\leq K_F W(\widehat{T}_G^{n-1}(\cdot | \mu), T_G^{n-1}(\cdot | \mu)) + \Delta \\ &= K_F \delta(n-1) + \Delta \leq \Delta \sum_{i=0}^{n-1} (K_F)^i. \end{aligned} \quad (5)$$

Note that in the triangle inequality, we may replace $\widehat{T}_G(\cdot | T_G^{n-1}(\cdot | \mu))$ with $T_G(\cdot | \widehat{T}_G^{n-1}(\cdot | \mu))$ and follow the same basic steps to get:

$$W(\widehat{T}_G^n(\cdot | \mu), T_G^n(\cdot | \mu)) \leq \Delta \sum_{i=0}^{n-1} (K_T)^i. \quad (6)$$

Combining (5) and (6) allows us to write:

$$\begin{aligned} \delta(n) &= W(\widehat{T}_G^n(\cdot | \mu), T_G^n(\cdot | \mu)) \\ &\leq \min \left\{ \Delta \sum_{i=0}^{n-1} (K_T)^i, \Delta \sum_{i=0}^{n-1} (K_F)^i \right\} \\ &= \Delta \sum_{i=0}^{n-1} (\bar{K})^i, \end{aligned}$$

which concludes the proof. \square

There exist similar results in the literature relating one-step transition error to multi-step transition error and sub-optimality bounds for planning with an approximate

Lemma 4. *The following operators (Asadi & Littman, 2017) are Lipschitz with constants:*

1. $K_{\|\cdot\|_\infty, d_R}(\max(x)) = K_{\|\cdot\|_\infty, d_R}(\text{mean}(x)) = K_{\|\cdot\|_\infty, d_R}(\epsilon\text{-greedy}(x)) = 1$
2. $K_{\|\cdot\|_\infty, d_R}(\text{mm}_\beta(x) := \frac{\log \sum_i e^{\beta x_i}}{\beta^n}) = 1$
3. $K_{\|\cdot\|_\infty, d_R}(\text{boltz}_\beta(x) := \frac{\sum_{i=1}^n x_i e^{\beta x_i}}{\sum_{i=1}^n e^{\beta x_i}}) \leq \sqrt{|A|} + \beta V_{\max} |A|$

Theorem 3. *For any non-expansion backup operator f outlined in Lemma 4, GVI computes a value function with a Lipschitz constant bounded by $\frac{K_{d_S, d_R}^A(R)}{1 - \gamma K_{d_S, W}^A(T)}$ if $\gamma K_{d_S, W}^A(T) < 1$.*

Proof. From Algorithm 1, in the n th round of GVI updates:

$$\widehat{Q}_{n+1}(s, a) \leftarrow R(s, a) + \gamma \int T(s' | s, a) f(\widehat{Q}_n(s', \cdot)) ds'.$$

Now observe that:

$$\begin{aligned} & K_{d_S, d_R}^A(\widehat{Q}_{n+1}) \\ & \leq K_{d_S, d_R}^A(R) + \gamma K_{d_S, d_R}^A \left(\int T(s' | s, a) f(\widehat{Q}_n(s', \cdot)) ds' \right) \\ & \leq K_{d_S, d_R}^A(R) + \gamma K_{d_S, W}^A(T) K_{d_S, \mathbb{R}} \left(f(\widehat{Q}_n(s, \cdot)) \right) \\ & \leq K_{d_S, d_R}^A(R) + \gamma K_{d_S, W}^A(T) K_{\|\cdot\|_\infty, d_R}(f) K_{d_S, d_R}^A(\widehat{Q}_n) \\ & = K_{d_S, d_R}^A(R) + \gamma K_{d_S, W}^A(T) K_{d_S, d_R}^A(\widehat{Q}_n) \end{aligned}$$

Where we used Lemmas 3, 2, and 4 for the second, third, and fourth inequality respectively. Equivalently:

$$\begin{aligned} K_{d_S, d_R}^A(\widehat{Q}_{n+1}) & \leq K_{d_S, d_R}^A(R) \sum_{i=0}^n (\gamma K_{d_S, W}^A(T))^i \\ & \quad + (\gamma K_{d_S, W}^A(T))^n K_{d_S, d_R}^A(\widehat{Q}_0). \end{aligned}$$

By computing the limit of both sides, we get:

$$\begin{aligned} \lim_{n \rightarrow \infty} K_{d_S, d_R}^A(\widehat{Q}_{n+1}) & \leq \lim_{n \rightarrow \infty} K_{d_S, d_R}^A(R) \sum_{i=0}^n (\gamma K_{d_S, W}^A(T))^i \\ & \quad + \lim_{n \rightarrow \infty} (\gamma K_{d_S, W}^A(T))^n K_{d_S, d_R}^A(\widehat{Q}_0) \\ & = \frac{K_{d_S, d_R}^A(R)}{1 - \gamma K_{d_S, W}^A(T)} + 0, \end{aligned}$$

This concludes the proof. \square

Two implications of this result: First, PAC exploration in continuous state spaces is shown assuming a Lipschitz value function (Pazis & Parr, 2013). However, the theorem shows

that it is sufficient to have a Lipschitz model, an assumption perhaps easier to confirm. The second implication relates to value-aware model learning (VAML) objective (Farahmand et al., 2017). Using the above theorem, we can show that minimizing Wasserstein is equivalent to minimizing the VAML objective (Asadi et al., 2018).

8. Experiments

Our first goal in this section¹ is to compare TV, KL, and Wasserstein in terms of the ability to best quantify error of an imperfect model. To this end, we built finite MRPs with random transitions, $|\mathcal{S}| = 10$ states, and $\gamma = 0.95$. In the first case the reward signal is randomly sampled from $[0, 10]$, and in the second case the reward of an state is the index of that state, so small Euclidean norm between two states is an indication of similar values. For 10^5 trials, we generated an MRP and a random model, and then computed model error and planning error (Figure 4). We understand a good metric as the one that computes a model error with a high correlation with value error. We show these correlations for different values of γ in Figure 5.

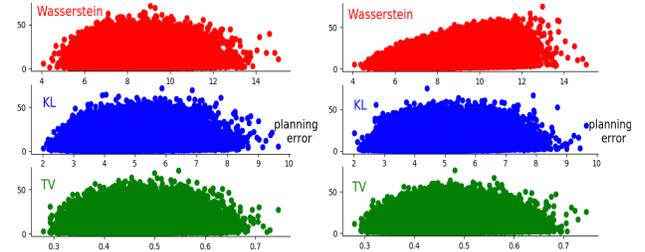


Figure 4. Value error (x axis) and model error (y axis). When the reward is the index of the state (right), correlation between Wasserstein error and value-prediction error is high. This highlights the fact that when closeness in the state-space is an indication of similar values, Wasserstein can be a powerful metric for model-based RL. Note that Wasserstein provides no advantage given random rewards (left).

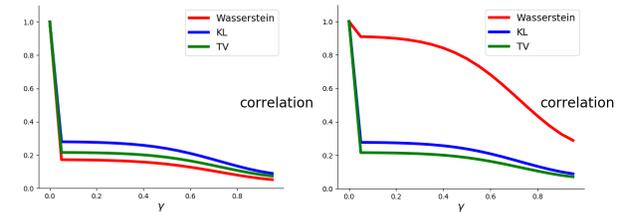


Figure 5. Correlation between value-prediction error and model error for the three metrics using random rewards (left) and index rewards (right). Given a useful notion of state similarities, low Wasserstein error is a better indication of planning error.

¹We release the code here: github.com/kavosh8/Lip

Function f	Definition	Lipschitz constant $K_{\ \cdot\ _p, \ \cdot\ _p}(f)$		
		$p = 1$	$p = 2$	$p = \infty$
$\text{ReLU} : \mathbb{R}^n \rightarrow \mathbb{R}^n$	$\text{ReLU}(x)_i := \max\{0, x_i\}$	1	1	1
$+b : \mathbb{R}^n \rightarrow \mathbb{R}^n, \forall b \in \mathbb{R}^n$	$+b(x) := x + b$	1	1	1
$\times W : \mathbb{R}^n \rightarrow \mathbb{R}^m, \forall W \in \mathbb{R}^{m \times n}$	$\times W(x) := Wx$	$\sum_j \ W_j\ _\infty$	$\sqrt{\sum_j \ W_j\ _2^2}$	$\sup_j \ W_j\ _1$

Table 1. Lipschitz constant for various functions used in a neural network. Here, W_j denotes the j th row of a weight matrix W .

It is known that controlling the Lipschitz constant of neural nets can help in terms of improving generalization error due to a lower bound on Rademacher complexity (Neysshabur et al., 2015; Bartlett & Mendelson, 2002). It then follows from Theorems 1 and 2 that controlling the Lipschitz constant of a learned transition model can achieve better error bounds for multi-step and value predictions. To enforce this constraint during learning, we bound the Lipschitz constant of various operations used in building neural network. The bound on the constant of the entire neural network then follows from Lemma 2. In Table 1, we provide Lipschitz constant for operations (see Appendix for proof) used in our experiments. We quantify these results for different p -norms $\|\cdot\|_p$.

Given these simple methods for enforcing Lipschitz continuity, we performed empirical evaluations to understand the impact of Lipschitz continuity of transition models, specifically when the transition model is used to perform multi-step state-predictions and policy improvements. We chose two standard domains: Cart Pole and Pendulum. In Cart Pole, we trained a network on a dataset of $15 * 10^3$ tuples $\langle s, a, s' \rangle$. During training, we ensured that the weights of the network are smaller than k . For each k , we performed 20 independent model estimation, and chose the model with median cross-validation error.

Using the learned model, along with the actual reward signal of the environment, we then performed stochastic actor-critic RL. (Barto et al., 1983; Sutton et al., 2000) This required an interaction between the policy and the learned model for relatively long trajectories. To measure the usefulness of the model, we then tested the learned policy on the actual domain. We repeated this experiment on Pendulum. To train the neural transition model for this domain we used 10^4 samples. Notably, we used deterministic policy gradient (Silver et al., 2014) for training the policy network with the hyper parameters suggested by Lillicrap et al. (2015). We report these results in Figure 6.

Observe that an intermediate Lipschitz constant yields the best result. Consistent with the theory, controlling the Lipschitz constant in practice can combat the compounding errors and can help in the value estimation problem. This ultimately results in learning a better policy.

We next examined if the benefits carry over to stochastic

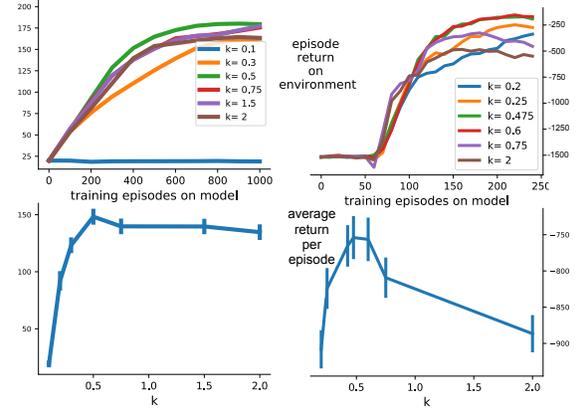


Figure 6. Impact of Lipschitz constant of learned models in Cart Pole (left) and Pendulum (right). An intermediate value of k (Lipschitz constant) yields the best performance.

settings. To capture stochasticity we need an algorithm to learn a Lipschitz model class (Definition 4). We used an EM algorithm to jointly learn a set of functions f , parameterized by $\theta = \{\theta^f : f \in F_g\}$, and a distribution over functions g . Note that in practice our dataset only consists of a set of samples $\langle s, a, s' \rangle$ and does not include the function the sample is drawn from. Hence, we consider this as our latent variable z . As is standard with EM, we start with the log-likelihood objective (for simplicity of presentation we assume a single action in the derivation):

$$\begin{aligned}
 L(\theta) &= \sum_{i=1}^N \log p(s_i, s_i'; \theta) \\
 &= \sum_{i=1}^N \log \sum_f p(z_i = f, s_i, s_i'; \theta) \\
 &= \sum_{i=1}^N \log \sum_f q(z_i = f | s_i, s_i') \frac{p(z_i = f, s_i, s_i'; \theta)}{q(z_i = f | s_i, s_i')} \\
 &\geq \sum_{i=1}^N \sum_f q(z_i = f | s_i, s_i') \log \frac{p(z_i = f, s_i, s_i'; \theta)}{q(z_i = f | s_i, s_i')},
 \end{aligned}$$

where we used Jensen's inequality and concavity of log in the last line. This derivation leads to the following EM algorithm.

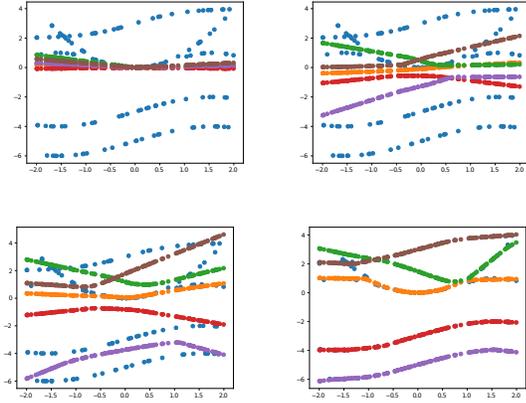


Figure 7. A stochastic problem solved by training a Lipschitz model class using EM. The top left figure shows the functions before any training (iteration 0), and the bottom right figure shows the final results (iteration 50).

In the M step, find θ_t by solving for:

$$\operatorname{argmax}_{\theta} \sum_{i=1}^N \sum_f q_{t-1}(z_i = f | s_i, s_i') \log \frac{p(z_i = f, s_i, s_i'; \theta)}{q_{t-1}(z_i = f | s_i, s_i')}$$

In the E step, compute posteriors:

$$q_t(z_i = f | s_i, s_i') = \frac{p(s_i, s_i' | z_i = f; \theta_t^f) g(z_i = f; \theta_t)}{\sum_f p(s_i, s_i' | z_i = f; \theta_t^f) g(z_i = f; \theta_t)}$$

Note that we assume each point is drawn from a neural network f with probability:

$$p(s_i, s_i' | z_i = f; \theta_t^f) = \mathcal{N}(|s_i' - f(s_i, \theta_t^f)|, \sigma^2),$$

and with a fixed variance σ^2 tuned as a hyper-parameter.

We used a supervised-learning domain to evaluate the EM algorithm. We generated 30 points from 5 functions (written at the end of Appendix) and trained 5 neural networks to fit these points. Iterations of a single run is shown in Figure 7 and the summary of results is presented in Figure 8. Observe that the EM algorithm is effective, and that controlling the Lipschitz constant is again useful.

We next applied EM to train a transition model for an RL setting, namely the gridworld domain from Moerland et al. (2017). Here a useful model needs to capture the stochastic behavior of the two ghosts. We modify the reward to be -1 whenever the agent is in the same cell as either one of the ghosts and 0 otherwise. We performed environmental interactions for 1000 time-steps and measured the return. We compared against standard tabular methods (Sutton & Barto, 1998), and a deterministic model that predicts expected next state (Sutton et al., 2008; Parr et al., 2008). In all cases we used value iteration for planning.

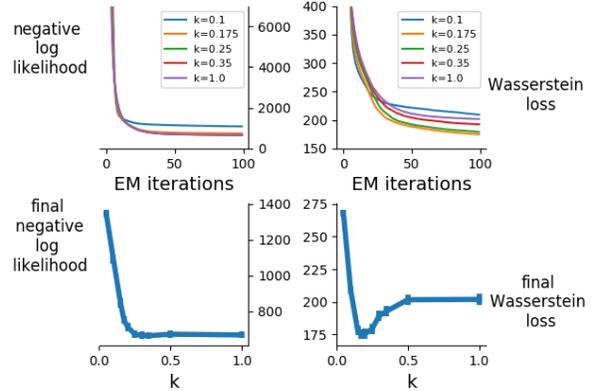


Figure 8. Impact of controlling the Lipschitz constant in the supervised-learning domain. Notice the U-shape of final Wasserstein loss with respect to Lipschitz constant k .

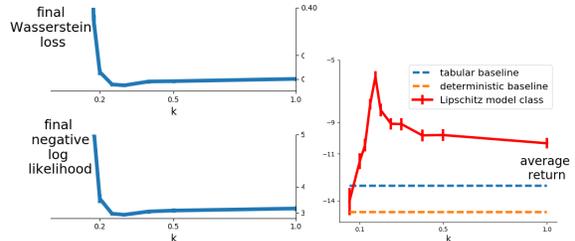


Figure 9. Performance of a Lipschitz model class on the gridworld domain. We show model test accuracy (left) and quality of the policy found using the model (right). Notice the poor performance of tabular and expected models.

Results in Figure 9 show that tabular models fail due to no generalization, and expected models fail since the ghosts do not move on expectation, a prediction not useful for planner. Performing value iteration with a Lipschitz model class outperforms the baselines.

9. Conclusion

We took an important step towards understanding model-based RL with function approximation. We showed that Lipschitz continuity of an estimated model plays a central role in multi-step prediction error, and in value-estimation error. We also showed the benefits of employing Wasserstein for model-based RL. An important future work is to apply these ideas to larger problems.

10. Acknowledgements

The authors recognize the assistance of Eli Upfal, John Langford, George Konidaris, and members of Brown's Rlab specifically Cameron Allen, David Abel, and Evan Cater. The authors also thank anonymous ICML reviewer 1 for insights on a value-aware interpretation of Wasserstein.

References

- Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.
- Asadi, K. and Littman, M. L. An alternative softmax operator for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 243–252, 2017.
- Asadi, K., Cater, E., Misra, D., and Littman, M. L. Equivalence between wasserstein and value-aware model-based reinforcement learning. *arXiv preprint arXiv:1806.01265*, 2018.
- Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.
- Barto, A. G., Sutton, R. S., and Anderson, C. W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, pp. 834–846, 1983.
- Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458, 2017.
- Bellman, R. A markovian decision process. *Journal of Mathematics and Mechanics*, pp. 679–684, 1957.
- Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pp. 908–919, 2017.
- Bertsekas, D. Convergence of discretization procedures in dynamic programming. *IEEE Transactions on Automatic Control*, 20(3):415–419, 1975.
- Bertsekas, D. P. and Tsitsiklis, J. N. Neuro-dynamic programming: an overview. In *Decision and Control, 1995, Proceedings of the 34th IEEE Conference on*, volume 1, pp. 560–564. IEEE, 1995.
- Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. X-armed bandits. *Journal of Machine Learning Research*, 12(May):1655–1695, 2011.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.
- Farahmand, A.-M., Barreto, A., and Nikovski, D. Value-Aware Loss Function for Model-based Reinforcement Learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1486–1494, 2017.
- Ferns, N., Panangaden, P., and Precup, D. Metrics for finite markov decision processes. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 162–169. AUAI Press, 2004.
- Fox, R., Pakman, A., and Tishby, N. G-learning: Taming the noise in reinforcement learning via soft updates. *Uncertainty in Artificial Intelligence*, 2016.
- Gao, B. and Pavel, L. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.
- Hinderer, K. Lipschitz continuity of value functions in Markovian decision processes. *Mathematical Methods of Operations Research*, 62(1):3–22, 2005.
- Jiang, N., Kulesza, A., Singh, S., and Lewis, R. The dependence of effective planning horizon on model accuracy. In *Proceedings of AAMAS*, pp. 1181–1189, 2015.
- Kaelbling, L. P., Littman, M. L., and Moore, A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
- Kakade, S., Kearns, M. J., and Langford, J. Exploration in metric state spaces. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 306–312, 2003.
- Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3): 209–232, 2002.
- Kleinberg, R., Slivkins, A., and Upfal, E. Multi-armed bandits in metric spaces. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pp. 681–690. ACM, 2008.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- Littman, M. L. and Szepesvári, C. A generalized reinforcement-learning model: Convergence and applications. In *Proceedings of the 13th International Conference on Machine Learning*, pp. 310–318, 1996.
- Lorenz, E. *Predictability: does the flap of a butterfly's wing in Brazil set off a tornado in Texas?* na, 1972.

- Moerland, T. M., Broekens, J., and Jonker, C. M. Learning multimodal transition dynamics for model-based reinforcement learning. *arXiv preprint arXiv:1705.00470*, 2017.
- Müller, A. Optimal selection from distributions with unknown parameters: Robustness of bayesian models. *Mathematical Methods of Operations Research*, 44(3): 371–386, 1996.
- Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. *arXiv preprint arXiv:1702.08892*, 2017.
- Narayanan, H. and Mitter, S. Sample complexity of testing the manifold hypothesis. In *Advances in Neural Information Processing Systems*, pp. 1786–1794, 2010.
- Neu, G., Jonsson, A., and Gómez, V. A unified view of entropy-regularized Markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.
- Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. In *Proceedings of The 28th Conference on Learning Theory*, pp. 1376–1401, 2015.
- Parr, R., Li, L., Taylor, G., Painter-Wakefield, C., and Littman, M. L. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 752–759. ACM, 2008.
- Pazis, J. and Parr, R. Pac optimal exploration in continuous space markov decision processes. In *AAAI*, 2013.
- Pires, B. Á. and Szepesvári, C. Policy error bounds for model-based reinforcement learning with factored linear models. In *Conference on Learning Theory*, pp. 121–151, 2016.
- Pirota, M., Restelli, M., and Bascetta, L. Policy gradient in lipschitz Markov decision processes. *Machine Learning*, 100(2-3):255–283, 2015.
- Rachelson, E. and Lagoudakis, M. G. On the locality of action domination in sequential decision making. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2010, Fort Lauderdale, Florida, USA, January 6-8, 2010*, 2010.
- Russell, S. J. and Norvig, P. *Artificial intelligence: A modern approach*, 1995.
- Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *ICML*, 2014.
- Strehl, A. L., Li, L., and Littman, M. L. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009.
- Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.
- Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.
- Sutton, R. S., Szepesvári, C., Geramifard, A., and Bowling, M. H. Dyna-style planning with linear function approximation and prioritized sweeping. In *UAI 2008, Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence, Helsinki, Finland, July 9-12, 2008*, pp. 528–536, 2008.
- Szepesvári, C. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1):1–103, 2010.
- Talvitie, E. Model regularization for stable sample rollouts. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 780–789. AUAI Press, 2014.
- Talvitie, E. Self-correcting models for model-based reinforcement learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017.
- Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- Vaserstein, L. N. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72, 1969.
- Venkatraman, A., Hebert, M., and Bagnell, J. A. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 2015.
- Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Appendix

Claim 1. *In a finite MDP, transition probabilities can be expressed using a finite set of deterministic functions and a distribution over the functions.*

Proof. Let $Pr(s, a, s')$ denote the probability of a transition from s to s' when executing the action a . Define an ordering over states s_1, \dots, s_n with an additional unreachable state s_0 . Now define the cumulative probability distribution:

$$C(s, a, s_i) := \sum_{j=0}^i Pr(s, a, s_j).$$

Further define L as the set of distinct entries in C :

$$L := \left\{ C(s, a, s_i) \mid s \in \mathcal{S}, i \in [0, n] \right\}.$$

Note that, since the MDP is assumed to be finite, then $|L|$ is finite. We sort the values of L and denote, by c_i , i th smallest value of the set. Note that $c_0 = 0$ and $c_{|L|} = 1$. We now build deterministic set of functions $f_1, \dots, f_{|L|}$ as follows: $\forall i = 1$ to $|L|$ and $\forall j = 1$ to n , define $f_i(s) = s_j$ if and only if:

$$C(s, a, s_{j-1}) < c_i \leq C(s, a, s_j).$$

We also define the probability distribution g over f as follows:

$$g(f_i|a) := c_i - c_{i-1}.$$

Given the functions $f_1, \dots, f_{|L|}$ and the distribution g , we can now compute the probability of a transition to s_j from s after executing action a :

$$\begin{aligned} & \sum_i \mathbb{1}(f_i(s) = s_j) g(f_i|a) \\ &= \sum_i \mathbb{1}(C(s, a, s_{j-1}) < c_i \leq C(s, a, s_j)) (c_i - c_{i-1}) \\ &= C(s, a, s_j) - C(s, a, s_{j-1}) \\ &= Pr(s, a, s_j), \end{aligned}$$

where $\mathbb{1}$ is a binary function that outputs one if and only if its condition holds. We reconstructed the transition probabilities using distribution g and deterministic functions $f_1, \dots, f_{|L|}$. \square

Claim 2. *Given a deterministic and linear transition model, and a linear reward signal, the bounds provided in Theorems 1 and 2 are both tight.*

Assume a linear transition function T defined as:

$$T(s) = Ks$$

Assume our learned transition function \hat{T} :

$$\hat{T}(s) := Ks + \Delta$$

Note that:

$$\max_s |T(s) - \hat{T}(s)| = \Delta$$

and that:

$$\min\{K_T, K_{\hat{T}}\} = K$$

First observe that the bound in Theorem 2 is tight for $n = 2$:

$$\forall s \quad \left| T(T(s)) - \hat{T}(\hat{T}(s)) \right| = \left| K^2s - K^2s + \Delta(1 + K) \right| = \Delta \sum_{i=0}^1 K^i$$

and more generally and after n compositions of the models, denoted by T^n and \hat{T}^n , the following equality holds:

$$\forall s \quad \left| T^n(s) - \hat{T}^n(s) \right| = \Delta \sum_{i=0}^{n-1} K^i$$

Lets further assume that the reward is linear:

$$R(s) = K_R s$$

Consider the state $s = 0$. Note that clearly $v(0) = 0$. We now compute the value predicted using \hat{T} , denoted by $\hat{v}(0)$:

$$\begin{aligned} \hat{v}(0) &= R(0) + \gamma R(0 + \Delta \sum_{i=0}^0 K^i) + \gamma^2 R(0 + \Delta \sum_{i=0}^1 K^i) + \gamma^3 R(0 + \Delta \sum_{i=0}^2 K^i) + \dots \\ &= 0 + \gamma K_R \Delta \sum_{i=0}^0 K^i + \gamma^2 K_R \Delta \sum_{i=0}^1 K^i + \gamma^3 K_R \Delta \sum_{i=0}^2 K^i + \dots \\ &= \gamma K_R \Delta \sum_{n=0}^{\infty} \gamma^n \sum_{i=0}^{n-1} K^i = \frac{\gamma K_R \Delta}{(1-\gamma)(1-\gamma K)} \end{aligned}$$

and so:

$$|v(0) - \hat{v}(0)| = \frac{\gamma K_R \Delta}{(1-\gamma)(1-\gamma K)}$$

Note that this exactly matches the bound derived in our Theorem 2.

Lemma 1. A generalized transition function $\hat{T}_{\mathcal{G}}$ induced by a Lipschitz model class F_g is Lipschitz with a constant:

$$K_{W,W}^A(\hat{T}_{\mathcal{G}}) := \sup_a \sup_{\mu_1, \mu_2} \frac{W(\hat{T}_{\mathcal{G}}(\cdot | \mu_1, a), \hat{T}_{\mathcal{G}}(\cdot | \mu_2, a))}{W(\mu_1, \mu_2)} \leq K_F$$

Proof.

$$\begin{aligned} W(\hat{T}(\cdot | \mu_1, a), \hat{T}(\cdot | \mu_2, a)) &:= \inf_j \int_{s'_1} \int_{s'_2} j(s'_1, s'_2) d(s'_1, s'_2) ds'_1 ds'_2 \\ &= \inf_j \int_{s_1} \int_{s_2} \int_{s'_1} \int_{s'_2} \sum_f \mathbb{1}(f(s_1) = s'_1 \wedge f(s_2) = s'_2) j(s_1, s_2, f) d(s'_1, s'_2) ds'_1 ds'_2 ds_1 ds_2 \\ &= \inf_j \int_{s_1} \int_{s_2} \sum_f j(s_1, s_2, f) d(f(s_1), f(s_2)) ds_1 ds_2 \\ &\leq K_F \inf_j \int_{s_1} \int_{s_2} \sum_f g(f|a) j(s_1, s_2) d(s_1, s_2) ds_1 ds_2 \\ &= K_F \sum_f g(f|a) \inf_j \int_{s_1} \int_{s_2} j(s_1, s_2) d(s_1, s_2) ds_1 ds_2 \\ &= K_F \sum_f g(f|a) W(\mu_1, \mu_2) = K_F W(\mu_1, \mu_2) \end{aligned}$$

Dividing by $W(\mu_1, \mu_2)$ and taking sup over a, μ_1 , and μ_2 , we conclude:

$$K_{W,W}^A(\hat{T}) = \sup_a \sup_{\mu_1, \mu_2} \frac{W(\hat{T}(\cdot | \mu_1, a), \hat{T}(\cdot | \mu_2, a))}{W(\mu_1, \mu_2)} \leq K_F .$$

We can also prove this using the Kantorovich-Rubinstein duality theorem:

For every μ_1, μ_2 , and $a \in \mathcal{A}$ we have:

$$\begin{aligned}
 W(\widehat{T}_{\mathcal{G}}(\cdot | \mu_1, a), \widehat{T}_{\mathcal{G}}(\cdot | \mu_2, a)) &= \sup_{f: K_{d_{\mathcal{S}}, \mathbb{R}}(f) \leq 1} \int_{\mathcal{S}} (\widehat{T}_{\mathcal{G}}(s | \mu_1, a) - \widehat{T}_{\mathcal{G}}(s | \mu_2, a)) f(s) ds \\
 &= \sup_{f: K_{d_{\mathcal{S}}, \mathbb{R}}(f) \leq 1} \int_{\mathcal{S}} \int_{s_0} (\widehat{T}(s | s_0, a) \mu_1(s_0) - \widehat{T}(s | s_0, a) \mu_2(s_0)) f(s) ds ds_0 \\
 &= \sup_{f: K_{d_{\mathcal{S}}, \mathbb{R}}(f) \leq 1} \int_{\mathcal{S}} \int_{s_0} \widehat{T}(s | s_0, a) (\mu_1(s_0) - \mu_2(s_0)) f(s) ds ds_0 \\
 &= \sup_{f: K_{d_{\mathcal{S}}, \mathbb{R}}(f) \leq 1} \int_{\mathcal{S}} \int_{s_0} \sum_t g(t | a) \mathbb{1}(t(s_0) = s) (\mu_1(s_0) - \mu_2(s_0)) f(s) ds ds_0 \\
 &= \sup_{f: K_{d_{\mathcal{S}}, \mathbb{R}}(f) \leq 1} \sum_t g(t | a) \int_{s_0} \int_{\mathcal{S}} \mathbb{1}(t(s_0) = s) (\mu_1(s_0) - \mu_2(s_0)) f(s) ds ds_0 \\
 &= \sup_{f: K_{d_{\mathcal{S}}, \mathbb{R}}(f) \leq 1} \sum_t g(t | a) \int_{s_0} (\mu_1(s_0) - \mu_2(s_0)) f(t(s_0)) ds_0 \\
 &\leq \sum_t g(t | a) \sup_{f: K_{d_{\mathcal{S}}, \mathbb{R}}(f) \leq 1} \int_{s_0} (\mu_1(s_0) - \mu_2(s_0)) f(t(s_0)) ds_0 \\
 &\quad \text{composition of } f, t \text{ is Lipschitz with constant upper bounded by } K_F. \\
 &= K_F \sum_t g(t | a) \sup_{f: K_{d_{\mathcal{S}}, \mathbb{R}}(f) \leq 1} \int_{s_0} (\mu_1(s_0) - \mu_2(s_0)) \frac{f(t(s_0))}{K_F} ds_0 \\
 &\leq K_F \sum_t g(t | a) \sup_{h: K_{d_{\mathcal{S}}, \mathbb{R}}(h) \leq 1} \int_{s_0} (\mu_1(s_0) - \mu_2(s_0)) h(s_0) ds_0 \\
 &= K_F \sum_t g(t | a) W(\mu_1, \mu_2) = K_F W(\mu_1, \mu_2)
 \end{aligned}$$

Again we conclude by dividing by $W(\mu_1, \mu_2)$ and taking sup over a, μ_1 , and μ_2 . \square

Lemma 2. (Composition Lemma) Define three metric spaces (M_1, d_1) , (M_2, d_2) , and (M_3, d_3) . Define Lipschitz functions $f : M_2 \mapsto M_3$ and $g : M_1 \mapsto M_2$ with constants $K_{d_2, d_3}(f)$ and $K_{d_1, d_2}(g)$. Then, $h : f \circ g : M_1 \mapsto M_3$ is Lipschitz with constant $K_{d_1, d_3}(h) \leq K_{d_2, d_3}(f) K_{d_1, d_2}(g)$.

Proof.

$$\begin{aligned}
 K_{d_1, d_3}(h) &= \sup_{s_1, s_2} \frac{d_3(f(g(s_1)), f(g(s_2)))}{d_1(s_1, s_2)} \\
 &= \sup_{s_1, s_2} \frac{d_2(g(s_1), g(s_2))}{d_1(s_1, s_2)} \frac{d_3(f(g(s_1)), f(g(s_2)))}{d_2(g(s_1), g(s_2))} \\
 &\leq \sup_{s_1, s_2} \frac{d_2(g(s_1), g(s_2))}{d_1(s_1, s_2)} \sup_{s_1, s_2} \frac{d_3(f(s_1), f(s_2))}{d_2(s_1, s_2)} \\
 &= K_{d_1, d_2}(g) K_{d_2, d_3}(f).
 \end{aligned}$$

\square

Lemma 3. Given a Lipschitz function $f : \mathcal{S} \mapsto \mathbb{R}$ with constant $K_{d_{\mathcal{S}}, d_{\mathbb{R}}}(f)$:

$$K_{d_{\mathcal{S}}, d_{\mathbb{R}}}^A \left(\int \widehat{T}(s' | s, a) f(s') ds' \right) \leq K_{d_{\mathcal{S}}, d_{\mathbb{R}}}(f) K_{d_{\mathcal{S}}, W}^A(\widehat{T}).$$

Proof.

$$\begin{aligned}
 K_{d_S, d_R}^A \left(\int_{s'} \widehat{T}(s'|s, a) f(s') ds' \right) &= \sup_a \sup_{s_1, s_2} \frac{|\int_{s'} (\widehat{T}(s'|s_1, a) - \widehat{T}(s'|s_2, a)) f(s') ds'|}{d(s_1, s_2)} \\
 &= \sup_a \sup_{s_1, s_2} \frac{|\int_{s'} (\widehat{T}(s'|s_1, a) - \widehat{T}(s'|s_2, a)) f(s') \frac{K_{d_S, d_R}(f)}{K_{d_S, d_R}(f)} ds'|}{d(s_1, s_2)} \\
 &= K_{d_S, d_R}(f) \sup_a \sup_{s_1, s_2} \frac{|\int_{s'} (\widehat{T}(s'|s_1, a) - \widehat{T}(s'|s_2, a)) \frac{f(s')}{K_{d_S, d_R}(f)} ds'|}{d(s_1, s_2)} \\
 &\leq K_{d_S, d_R}(f) \sup_a \sup_{s_1, s_2} \frac{|\sup_{g: K_{d_S, d_R}(g) \leq 1} \int_{s'} (\widehat{T}(s'|s_1, a) - \widehat{T}(s'|s_2, a)) g(s') ds'|}{d(s_1, s_2)} \\
 &= K_{d_S, d_R}(f) \sup_a \sup_{s_1, s_2} \frac{\sup_{g: K_{d_S, d_R}(g) \leq 1} \int_{s'} (\widehat{T}(s'|s_1, a) - \widehat{T}(s'|s_2, a)) g(s') ds'}{d(s_1, s_2)} \\
 &= K_{d_S, d_R}(f) \sup_a \sup_{s_1, s_2} \frac{W(\widehat{T}(\cdot|s_1, a), \widehat{T}(\cdot|s_2, a))}{d(s_1, s_2)} \\
 &= K_{d_S, d_R}(f) K_{d_S, W}^A(\widehat{T}).
 \end{aligned}$$

□

Lemma 4. *The following operators (Asadi & Littman, 2017) are Lipschitz with constants:*

1. $K_{\|\cdot\|_\infty, d_R}(\max(x)) = K_{\|\cdot\|_\infty, d_R}(\text{mean}(x)) = K_{\|\cdot\|_\infty, d_R}(\epsilon\text{-greedy}(x)) = 1$
2. $K_{\|\cdot\|_\infty, d_R}(mm_\beta(x) := \frac{\log \frac{\sum_i e^{\beta x_i}}{n}}{\beta}) = 1$
3. $K_{\|\cdot\|_\infty, d_R}(\text{boltz}_\beta(x) := \frac{\sum_{i=1}^n x_i e^{\beta x_i}}{\sum_{i=1}^n e^{\beta x_i}}) \leq \sqrt{|A|} + \beta V_{\max} |A|$

Proof. 1 was proven by Littman & Szepesvári (1996), and 2 is proven several times (Fox et al., 2016; Asadi & Littman, 2017; Nachum et al., 2017; Neu et al., 2017). We focus on proving 3. Define

$$\rho(x)_i = \frac{e^{\beta x_i}}{\sum_{i=1}^n e^{\beta x_i}},$$

and observe that $\text{boltz}_\beta(x) = x^\top \rho(x)$. Gao & Pavel (2017) showed that ρ is Lipschitz:

$$\|\rho(x_1) - \rho(x_2)\|_2 \leq \beta \|x_1 - x_2\|_2 \quad (7)$$

Using their result, we can further show:

$$\begin{aligned}
 &|\rho(x_1)^\top x_1 - \rho(x_2)^\top x_2| \\
 &\leq |\rho(x_1)^\top x_1 - \rho(x_1)^\top x_2| + |\rho(x_1)^\top x_2 - \rho(x_2)^\top x_2| \\
 &\leq \|\rho(x_1)\|_2 \|x_1 - x_2\|_2 \\
 &\quad + \|x_2\|_2 \|\rho(x_1) - \rho(x_2)\|_2 \quad (\text{Cauchy-Shwartz}) \\
 &\leq \|\rho(x_1)\|_2 \|x_1 - x_2\|_2 \\
 &\quad + \|x_2\|_2 \beta \|x_1 - x_2\|_2 \quad (\text{from Eqn 7}) \\
 &\leq (1 + \beta V_{\max} \sqrt{|A|}) \|x_1 - x_2\|_2 \\
 &\leq (\sqrt{|A|} + \beta V_{\max} |A|) \|x_1 - x_2\|_\infty,
 \end{aligned}$$

dividing both sides by $\|x_1 - x_2\|_\infty$ leads to 3. □

Below, we derive the Lipschitz constant for various functions mentioned in Table 1.

ReLU non-linearity We show that ReLu : $\mathbb{R}^n \rightarrow \mathbb{R}^n$ has Lipschitz constant 1 for p .

$$\begin{aligned}
 K_{\|\cdot\|_p, \|\cdot\|_p}(\text{ReLU}) &= \sup_{x_1, x_2} \frac{\|\text{ReLU}(x_1) - \text{ReLU}(x_2)\|_p}{\|x_1 - x_2\|_p} \\
 &= \sup_{x_1, x_2} \frac{(\sum_i |\text{ReLU}(x_1)_i - \text{ReLU}(x_2)_i|^p)^{\frac{1}{p}}}{\|x_1 - x_2\|_p} \\
 &\quad (\text{We can show that } |\text{ReLU}(x_1)_i - \text{ReLU}(x_2)_i| \leq |x_{1,i} - x_{2,i}| \text{ and so) :} \\
 &\leq \sup_{x_1, x_2} \frac{(\sum_i |x_{1,i} - x_{2,i}|^p)^{\frac{1}{p}}}{\|x_1 - x_2\|_p} \\
 &= \sup_{x_1, x_2} \frac{\|x_1 - x_2\|_p}{\|x_1 - x_2\|_p} = 1
 \end{aligned}$$

Matrix multiplication Let $W \in \mathbb{R}^{n \times m}$. We derive the Lipschitz continuity for the function $\times W(x) = Wx$.

For $p = \infty$ we have:

$$\begin{aligned}
 &K_{\|\cdot\|_\infty, \|\cdot\|_\infty}(\times W(x_1)) \\
 &= \sup_{x_1, x_2} \frac{\|\times W(x_1) - \times W(x_2)\|_\infty}{\|x_1 - x_2\|_\infty} = \sup_{x_1, x_2} \frac{\|Wx_1 - Wx_2\|_\infty}{\|x_1 - x_2\|_\infty} = \sup_{x_1, x_2} \frac{\|W(x_1 - x_2)\|_\infty}{\|x_1 - x_2\|_\infty} \\
 &= \sup_{x_1, x_2} \frac{\sup_j |W_j(x_1 - x_2)|}{\|x_1 - x_2\|_\infty} \\
 &\leq \sup_{x_1, x_2} \frac{\sup_j \|W_j\| \|x_1 - x_2\|_\infty}{\|x_1 - x_2\|_\infty} \quad (\text{H\"older's inequality}) \\
 &= \sup_j \|W_j\|_1,
 \end{aligned}$$

where W_j refers to j th row of the weight matrix W . Similarly, for $p = 1$ we have:

$$\begin{aligned}
 &K_{\|\cdot\|_1, \|\cdot\|_1}(\times W(x_1)) \\
 &= \sup_{x_1, x_2} \frac{\|\times W(x_1) - \times W(x_2)\|_1}{\|x_1 - x_2\|_1} = \sup_{x_1, x_2} \frac{\|Wx_1 - Wx_2\|_1}{\|x_1 - x_2\|_1} = \sup_{x_1, x_2} \frac{\|W(x_1 - x_2)\|_1}{\|x_1 - x_2\|_1} \\
 &= \sup_{x_1, x_2} \frac{\sum_j |W_j(x_1 - x_2)|}{\|x_1 - x_2\|_1} \\
 &\leq \sup_{x_1, x_2} \frac{\sum_j \|W_j\|_\infty \|x_1 - x_2\|_1}{\|x_1 - x_2\|_1} = \sum_j \|W_j\|_\infty,
 \end{aligned}$$

and finally for $p = 2$:

$$\begin{aligned}
 &K_{\|\cdot\|_2, \|\cdot\|_2}(\times W(x_1)) \\
 &= \sup_{x_1, x_2} \frac{\|\times W(x_1) - \times W(x_2)\|_2}{\|x_1 - x_2\|_2} = \sup_{x_1, x_2} \frac{\|Wx_1 - Wx_2\|_2}{\|x_1 - x_2\|_2} = \sup_{x_1, x_2} \frac{\|W(x_1 - x_2)\|_2}{\|x_1 - x_2\|_2} \\
 &= \sup_{x_1, x_2} \frac{\sqrt{\sum_j |W_j(x_1 - x_2)|^2}}{\|x_1 - x_2\|_2} \\
 &\leq \sup_{x_1, x_2} \frac{\sqrt{\sum_j \|W_j\|_2^2 \|x_1 - x_2\|_2^2}}{\|x_1 - x_2\|_2} = \sqrt{\sum_j \|W_j\|_2^2}.
 \end{aligned}$$

Vector addition We show that $+b : \mathbb{R}^n \rightarrow \mathbb{R}^n$ has Lipschitz constant 1 for $p = 0, 1, \infty$ for all $b \in \mathbb{R}^n$.

$$\begin{aligned} K_{\|\cdot\|_p, \|\cdot\|_p}(\text{ReLU}) &= \sup_{x_1, x_2} \frac{\|+b(x_1) - +b(x_2)\|_p}{\|x_1 - x_2\|_p} \\ &= \sup_{x_1, x_2} \frac{\|(x_1 + b) - (x_2 + b)\|_p}{\|x_1 - x_2\|_p} = \frac{\|x_1 - x_2\|_p}{\|x_1 - x_2\|_p} = 1 \end{aligned}$$

Supervised-learning domain We used the following 5 functions to generate the dataset:

$$\begin{aligned} f_0(x) &= \tanh(x) + 3 \\ f_1(x) &= x * x \\ f_2(x) &= \sin(x) - 5 \\ f_3(x) &= \sin(x) - 3 \\ f_4(x) &= \sin(x) * \sin(x) \end{aligned}$$

We sampled each function 30 times, where the input was chosen uniformly randomly from $[-2, 2]$ each time.