Value-Aware Loss Function for Model-based Reinforcement Learning

Amir-massoud Farahmand
Mitsubishi Electric Research Laboratories (MERL), USA

André M.S. Barreto
National Laboratory for Scientific Computing (LNCC), Brazil

Daniel N. Nikovski
Mitsubishi Electric Research Laboratories (MERL), USA

Abstract

We consider the problem of estimating the transition probability kernel to be used by a model-based reinforcement learning (RL) algorithm. We argue that estimating a generative model that minimizes a probabilistic loss, such as the log-loss, is an overkill because it does not take into account the underlying structure of decision problem and the RL algorithm that intends to solve it. We introduce a loss function that takes the structure of the value function into account. We provide a finite-sample upper bound for the loss function showing the dependence of the error on model approximation error, number of samples, and the complexity of the model space. We also empirically compare the method with the maximum likelihood estimator on a simple problem.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

1 INTRODUCTION

The standard approach to model-based reinforcement learning (RL) [Sutton and Barto, 1998; Szepesvári, 2010] is to use data $\mathcal{D}_n = \{(X_i, A_i, R_i, X_i')\}_{i=1}^n$ to estimate the transition probability kernel $\mathcal{P}^*$ by $\hat{\mathcal{P}}$ and the expected reward function r by $\hat{r}$. The learned model is then used to generate new samples, see e.g., [Sutton et al., 2008; Farahmand et al., 2009; Hester and Stone, 2013; Deisenroth et al., 2013, 2015]. A standard RL/Planning algorithm can use these samples to find a close to optimal policy, possibly by first finding an approximation to the optimal (action-)value function. Estimating $\mathcal{P}^*$ by $\hat{\mathcal{P}}$ is the problem of conditional probability (density/distribution) estimation and the estimating r by $\hat{r}$ is a regression problem. In the rest of this work, we only focus on learning $\mathcal{P}^*$.[1]

There are several general approaches to estimate $\mathcal{P}^*$ such as Maximum Likelihood Estimation (MLE), Maximum Entropy (MaxEnt) estimation, the Maximum A Posteriori (MAP) estimation, and Bayesian posterior inference. We argue that these conventional approaches to find a generative model might be an overkill, thus may not be required.

For example, consider the ML estimate, which is the minimizer of the empirical negative-log loss, which in turn is an empirical approximation to the KL divergence $\mathsf{KL}(P_1||P_2) = \sum_{x\in\mathcal{X}} P_1(x) \log \frac{P_1(x)}{P_2(x)}$.[2] Minimizing the KL divergence is generally seen as a desirable goal for learning a probabilistic model because $\mathsf{KL}(P_1||P_2) = 0$ if and only if P_1 and P_2 are the same almost surely. Given dataset $\mathcal{D}_n = \{X_i\}_{i=1}^n$ with $X_i \sim P^*$, we define the empirical measure $P_n(\cdot) = \frac{1}{n}\sum_{i=1}^n \delta_{X_i}(\cdot)$. The MLE within a probability model space $\mathcal{M}$ is

$$\hat{P} \leftarrow \operatorname*{argmin}_{P\in\mathcal{M}} \mathsf{KL}(P_n||P) \equiv \operatorname*{argmax}_{P\in\mathcal{M}} \frac{1}{n} \sum_{X_i\in\mathcal{D}_n} \log P(X_i). \tag{1}$$

In the context of model-based RL, learning $\hat{\mathcal{P}}$ that minimizes a negative-log loss or other "probabilistic" losses leads to an estimate that tries to model all aspects of the environment. This might be beyond the requirement of solving the RL problem effectively. It might be the case that some aspects of the environment are irrelevant to find a good or optimal policy. For example, consider a visually-enabled robot that is supposed to learn how to navigate in a building. If we consider the camera image as a part of the state of the robot, trying to learn a transition probability kernel means that we have to learn how the camera image changes when the robot takes certain actions. This is a very high-dimensional state space and trying to learn such a conditional distribution with high enough accuracy, in the log-loss sense, is quite difficult. Nonetheless, modeling the probability distribution at that level of accuracy is not required to learn a policy that can navigate the robot in the building just fine. The only aspect of the model that is really required is a crude model that describes the building's topology as well as distances between rooms, and maybe the location of the objects. The robot does not really need to know the detail of paintings on the walls, the texture of objects, and many other visual detail of the building. On the other hand, if the goal is to have an interior decorator robot that suggests how to redecorate the building to make it visually appealing, all those visual information is required.

The difference between the navigator robot and the decorator one is not in the transition model of their environment, but is in the decision problem that they have to solve. The difference in the decision problem is reflected in the difference in the reward functions and as a result in the value functions. It is desirable to have a model learning formalism that takes the decision problem, or at least some aspects of it, into account.

Furthermore, the implicit assumption that model approximation error can be made zero, that is, $\mathcal{P}^*$ belongs to $\mathcal{M}$ used for estimation, may not be correct for many estimators. When we have the model approximation error, the model learning method must make a compromise in the choice of the estimate: None of the models in $\mathcal{M}$ would be the same as $\mathcal{P}^*$ (e.g., in the almost sure sense), so the estimation method has to choose a model with a minimum error with respect to (w.r.t.) some loss function. The choice of the loss function becomes important then. A loss function that is designed for a particular decision problem in hand provides a better approximation, for the task of solving the very same decision problem, than a probabilistic one that does not take the decision problem into account.

[1] An extended abstract version of this paper has been presented at European Workshop on Reinforcement Learning [Farahmand et al., 2016].

[2] We use P, $\hat{P}$, etc. to denote an unconditional probability distribution over $\mathcal{X}$, and we use $\mathcal{P}$, $\hat{\mathcal{P}}$, etc. to denote a conditional transition probability kernel.

Algorithm 1 Generic Model-based Reinforcement Learning Algorithm

// MDP $(\mathcal{X}, \mathcal{A}, \mathcal{R}^*, \mathcal{P}^*, \gamma)$
// K: Number of interaction episodes
// $\mathcal{M}$: Space of transition probability kernels
// $\mathcal{G}$: Space of reward functions
Initialize a policy π_0
for $k = 0$ to $K - 1$ **do**
  Generate training set $\mathcal{D}_n^{(k)} = \{(X_i, A_i, R_i, X_i')\}_{i=1}^n$ by interacting with the true environment (potentially using π_k), i.e., $(X_i, A_i) \sim \nu_k$ with $X_i' \sim \mathcal{P}^*(\cdot|X_i, A_i)$ and $R_i \sim \mathcal{R}^*(\cdot|X_i, A_i)$.
  $\hat{\mathcal{P}} \leftarrow \operatorname{argmin}_{\mathcal{P} \in \mathcal{M}} \text{Loss}_{\mathcal{P}}(\mathcal{P}; \cup_{i=0}^k \mathcal{D}_n^{(i)})$ {e.g., by the gradient descent specified in Theorem 1 for VAML or (14) for MLE}
  $\hat{r} \leftarrow \operatorname{argmin}_{r \in \mathcal{G}} \text{Loss}_{\mathcal{R}}(r; \cup_{i=0}^k \mathcal{D}_n^{(i)})$
  $\pi_{k+1} \leftarrow \mathsf{Planner}(\hat{\mathcal{P}}, \hat{\mathcal{R}})$ {e.g., Fitted Q-Iteration}
end for
return π_K

These arguments suggest that generic distribution estimation approaches such as MLE, which minimizes the KL-divergence w.r.t. the empirical distribution, might not be the best candidate for learning a model to be used within a model-based RL framework. Can we design a better "decision-aware" loss function that takes the decision problem into account?

This paper is a step towards incorporating some aspects of the underlying decision problem into model learning. We go beyond the "vanilla" model learning, and define a new loss function that incorporate the structure of the value function to learn the transition model (Section 2). We call the approach based on this loss function *Value-Aware Model Learning (VAML)*. We also provide a finite-sample upper bound guarantee for VAML in Section 3 showing the effect of the model approximation error, number of training samples, and the complexity of the model space. This guarantees the soundness of the algorithm. We empirically study the model learned by VAML/MLE within a complete model-based RL framework in a simple finite MDP problem. Moreover, we analyze the model approximation properties of VAML vs. MLE through a series of simple, but illuminating, examples (Section 4). We also provide additional empirical results studying various aspects of VAML vs. MLE. The general conclusion of these results is that VAML is superior to MLE whenever we have a model approximation error, i.e., the true transition model does not belong to the class of models in which our estimator is selected.

2 VALUE-AWARE MODEL LEARNING

Algorithm 1 describes a generic model-based RL agent. It interacts with the environment, which is specified by an unknown Markov Decision Process (MDP) $(\mathcal{X}, \mathcal{A}, \mathcal{R}^*, \mathcal{P}^*, \gamma)$, to collect data $\mathcal{D}_n$. Here $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}^*$ is the reward distribution, and $\mathcal{P}^*$ is the transition probability kernel, and $0 \leq \gamma < 1$ is the discount factor [Szepesvári, 2010]. The data is used to learn an estimate $\hat{\mathcal{P}}$ of the true transition probability $\mathcal{P}^*$ of the environment and an estimate $\hat{r}$ of the expected reward. The model learning step is usually done using an MLE, e.g., counting the number of transitions from state-action pair (x, a) to another state x' in a finite state-action MDP is such an estimate. The learned model is then used by a planning algorithm Planner to find a policy, with the goal of finding a close to optimal policy. The new policy might be used to generate more data and improve the model.

There are many variations to each step of this generic algorithm such as how to collect new data points (cf. [Hester and Stone, 2013]) or what Planner should we use from all possible value-based, policy search, etc. algorithms. Moreover, the interaction with the environment might be in a one-shot batch setting ($K = 1$) or in a continual online setting, and the spectrum in between.

The main thesis of this work is that estimating the model should be influenced by the way Planner is going to use it. So we focus on estimating the transition probability and we study how we should define a loss function $\text{Loss}_{\mathcal{P}}$. We ignore all other important issues regarding designing a model-based RL algorithm for the moment in the rest of this work, except in the section on empirical studies (Section 5) where we choose a particular algorithm as Planner.

Let Planner be an algorithm that receives a model $\hat{\mathcal{P}}$ and returns a policy $\pi \leftarrow \mathsf{Planner}(\hat{\mathcal{P}})$. We assume that the reward function is already known to Planner, so we do not explicitly pass it as an argument. For a user-defined

initial probability distribution $\rho \in \bar{\mathcal{M}}(\mathcal{X})$, with $\bar{\mathcal{M}}(\mathcal{X})$ being the space of probability distributions on $\mathcal{X}$, we evaluate the performance of π by

$$J(\pi) = \int \mathrm{d}\rho(x) V^\pi(x). \tag{2}$$

The goal of a successful model learner can then be defined as follows: Given a dataset $\mathcal{D}_n = \{(X_i, A_i, X_i')\}_{i=1}^n$ with $Z_i = (X_i, A_i) \sim \nu(\mathcal{X} \times \mathcal{A}) \in \bar{\mathcal{M}}(\mathcal{X} \times \mathcal{A})$, potentially different from ρ, and $X_i' \sim \mathcal{P}^*(\cdot|X_i, A_i)$, find $\hat{\mathcal{P}}$ such that $J(\pi)$ for $\pi \leftarrow \mathsf{Planner}(\hat{\mathcal{P}})$ is as large as possible. This is a very generic goal. To make it more concrete, we have to make a few choices. First suppose that $\mathsf{Planner}$ uses the Bellman optimality operator defined based on $\hat{\mathcal{P}}$ to find a $\hat{Q}^*$, that is $\hat{T}^* : Q \mapsto r + \gamma \hat{\mathcal{P}} \max_a Q$, and then outputs $\pi = \hat{\pi}(\cdot; \hat{Q}^*)$, the greedy policy w.r.t. $\hat{Q}^*$ defined as $\hat{\pi}(x; Q) = \mathrm{argmax}_{a \in \mathcal{A}} Q(x, a)$. The use of the Bellman [optimality] operator is central to value-based approaches such as the class of (Approximate) Value Iteration or (Approximate) Policy Iteration algorithms.

This is still too general, so we focus on the more specified goal of finding a $\hat{\mathcal{P}}$ such that the difference between T^*Q and $\hat{T}^*Q$ is not large. We may express this goal by defining the following cost (loss):

$$\begin{aligned} c(\hat{\mathcal{P}}, \mathcal{P}^*; V)(x, a) &= \left| \left[\mathcal{P}^*(\cdot|x, a) - \hat{\mathcal{P}}(\cdot|x, a) \right] V(\cdot) \right| = \left| \left\langle \mathcal{P}^*(\cdot|x, a) - \hat{\mathcal{P}}(\cdot|x, a) \,,\, V \right\rangle \right| \\ &= \left| \int \left[\mathcal{P}^*(\mathrm{d}x'|x, a) - \hat{\mathcal{P}}(\mathrm{d}x'|x, a) \right] V(x') \right|, \end{aligned} \tag{3}$$

in which we substituted $\max_a Q(\cdot, a)$ with V to simplify the presentation. In the rest of the paper, we may sometimes use $\mathcal{P}_z(\cdot)$ with $z = (x, a) \in \mathcal{Z} = \mathcal{X} \times \mathcal{A}$ to refer to the probability distribution $\mathcal{P}(\cdot|x, a)$, so $\mathcal{P}_z V = \int \mathcal{P}(\mathrm{d}y|x, a) V(\mathrm{d}y)$.

It might be argued that since

$$\left| \left\langle \mathcal{P}^*(\cdot|x, a) - \hat{\mathcal{P}}(\cdot|x, a) \,,\, V \right\rangle \right| \le \left\| \mathcal{P}^*(\cdot|x, a) - \hat{\mathcal{P}}(\cdot|x, a) \right\|_1 \|V\|_\infty , \tag{4}$$

it is enough to learn $\hat{\mathcal{P}}$ such that the ℓ_1-norm of its difference with the true $\mathcal{P}^*$ is small. This can be achieved by minimizing the KL divergence because Pinsker's inequality shows that for two probability distributions P_1 and P_2, we have[3]

$$\|P_1 - P_2\|_1 \le \sqrt{2 \mathsf{KL}(P_1 || P_2)}. \tag{5}$$

These two upper bounds together justify the use of MLE since MLE is the minimizer of the empirical approximation of the KL divergence, as shown in Section 1. This is the argument, sometimes implicit, behind most model-based RL algorithms that use a log-loss or a similar "probabilistic" loss to estimate the model.

Finding a minimizer for the KL divergence, Hellinger distance, ℓ_1 loss, or other losses that depend only on the probabilities, however, ignores the underlying decision problem, which is specified through the reward/value function. As an extreme example, suppose that $r(x) = c$ for all $x \in \mathcal{X}$, so V^π is constant for all policies, and the optimal policy would not have any preference over any of the actions. So even if $\mathcal{X}$ is a very large space (e.g., a subset of $\mathbb{R}^d$ with a large d), and however complex $\mathcal{P}^*$ is (e.g., the dynamics is not very regular), learning a $\hat{\mathcal{P}}$ sufficient to find the optimal policy is indeed very easy: Any transition probability distribution suffices to find the optimal policy. In contrast, $\|\hat{\mathcal{P}} - \mathcal{P}^*\|_1$ goes to zero at a convergence rate that depends on dimension d and regularities of $\mathcal{P}^*$, and can be very slow, e.g., $O(n^{-1/2d})$. An estimator for $\mathcal{P}^*$ that ignores this extra information requires more samples in order to provide a guarantee that the error in the model-based planning is small.[4]

Moreover, and maybe more importantly, if the true transition kernel $\mathcal{P}^*$ does not belong to the model space $\mathcal{M}$ from which we estimate the model $\hat{\mathcal{P}}$, we can only hope to find the "closest" model within $\mathcal{M}$ to $\mathcal{P}^*$. The notion of closeness, however, depends on the distance measure. A distance measure that explicitly takes into account the decision problem and what really matters for $\mathsf{Planner}$ can be superior to the one that does not.

Returning to (3), there are three hurdles that should be addressed. The first is that $c(\hat{\mathcal{P}}, \mathcal{P}^*; V)(x, a)$ is defined as a pointwise measure of error, but we would like to learn a model that is valid for the whole state-action

[3] Or with a different constant depending on the base of the logarithm used. Here it is with the natural logarithm.

[4] The relationship between the probabilistic loss and the LHS of (4) is a bit more subtle than what we portrayed here, but this should be enough for our current discussion. Refer to Section 4 for a discussion on this issue.

space $\mathcal{X} \times \mathcal{A}$. The second is that V itself is not known, so one cannot optimize this cost as is. The third is that $\mathcal{P}^*$, which is the main object of interest, is not known. Instead we have $\mathcal{D}_n = \{(X_i, A_i, X'_i)\}_{i=1}^n$ and as a result the empirical conditional distribution $\mathcal{P}_n(\cdot|x,a) = \frac{1}{n}\sum_{i=1}^n \delta_{X'_i|X_i,A_i}(\cdot|x,a)$. Here the conditional Dirac's delta function is defined as follows: For a measurable set S, $\delta_{X'_i|X_i,A_i}(S|x,a) = 1$ whenever $(x,a) = (X_i, A_i)$ and $X'_i \in S$, and 0 otherwise.

We can easily address the first concern by defining the cost functional as the expected squared pointwise cost w.r.t. a probability distribution $\nu \in \bar{\mathcal{M}}(\mathcal{X} \times \mathcal{A})$, i.e.,

$$c^2_{2,\nu}(\hat{\mathcal{P}}, \mathcal{P}^*; V) = \int \mathrm{d}\nu(x,a) \left| \int \left[\mathcal{P}^*(\mathrm{d}x'|x,a) - \hat{\mathcal{P}}(\mathrm{d}x'|x,a)\right] V(x') \right|^2 . \tag{6}$$

The choice of the $L_2(\nu)$-norm of the pointwise cost is motivated by the relation between the performance loss $J(\pi^*) - J(\pi) = \|V^* - V^\pi\|_{1,\rho}$ and the $L_2(\nu)$ of quantities such as the Bellman error $Q - T^\pi Q$ in API or the approximation error $T^* Q_k - Q_{k+1}$ in AVI [Farahmand et al., 2010]. Somehow looser relationship also exists between the performance loss and the $L_1(\nu)$ error [Munos, 2007], but working with the squared error is easier in our future derivations. One could use the supremum norm too, but it would be too conservative. The choice of ν determines where in the state-action space we have to emphasize the accuracy of the model, in the sense of how well it can approximate the effect of the Bellman operator evaluated at that point. From the error propagation results such as Munos [2007]; Farahmand et al. [2010] we know that the exact relation between ν and the performance loss, which is defined w.r.t. ρ, is often complicated, so choosing a ν such that the performance loss is minimized is far from trivial. The distribution ν is often selected to be the distribution of data $(X, A) \sim \nu$, and this is what we assume in the rest of this work, but it can be different too, e.g., by using importance sampling. We do not study the question of how to choose ν in this work, and we assume that it is given.When the choice of ν is clear from the context, we may simply write $c(\hat{\mathcal{P}}, \mathcal{P}^*; V)$.

To address the second concern, not knowing V, we may take a robust approach w.r.t. the choice of value function. We define the cost function to reflect that our goal is to find a $\hat{\mathcal{P}}$ that is suitable for all V in a given value function space $\mathcal{F}$. Therefore, we define

$$c^2_{2,\nu}(\hat{\mathcal{P}}, \mathcal{P}^*) = \int \mathrm{d}\nu(x,a) \sup_{V \in \mathcal{F}} \left| \int \left[\mathcal{P}^*(\mathrm{d}x'|x,a) - \hat{\mathcal{P}}(\mathrm{d}x'|x,a)\right] V(x') \right|^2 . \tag{7}$$

To understand this loss better, let us focus on a single state-action pair (x,a) and study the pointwise cost.[5] Note that even though

$$\sup_{V \in \mathcal{F}} \left| [\mathcal{P}^*(\cdot|x,a) - \hat{\mathcal{P}}(\cdot|x,a)] V(\cdot) \right| \le \left\| \hat{\mathcal{P}}(\cdot|x,a) - \mathcal{P}^*(\cdot|x,a) \right\|_1 \sup_{V \in \mathcal{F}} \|V\|_\infty , \tag{8}$$

the LHS is often much smaller than the upper bound. They would only become equal when $\mathcal{F}$ is the space of bounded measurable functions, which is much larger than the usual function spaces that we often deal with, e.g., defined based on a finite set of basis or even a reproducing kernel Hilbert space (RKHS). As the goal is to minimize the LHS of (8), and because its RHS can be a loose upper bound for most choices of $\mathcal{F}$, directly optimizing the LHS can lead to better models compared to minimizing the ℓ_1 loss or the KL distance (minimized by MLE), which itself is yet another level of upper bounding according to (5).

The loss function (7) reflects the influence of the value function on the model-learning objective. If we happen to know that V has certain regularities, e.g., it belongs to the Sobolev space $\mathbb{W}^k(\mathbb{R}^d)$ or a reproducing kernel Hilbert space, this loss function lets us focus on learning a model that can discriminate between such value functions, and not more.

To address the last concern, one approach is to follow the usual recipe in machine learning and statistics, the Empirical Risk Minimization (ERM), by replacing the true state transition kernel $\mathcal{P}^*$ with the observed empirical distribution $\mathcal{P}_n$ and $\nu \in \bar{\mathcal{M}}(\mathcal{X} \times \mathcal{A})$ with the empirical measure $\nu_n(\cdot) = \frac{1}{n}\sum_{i=1}^n \delta_{(X_i,A_i)}(\cdot)$. Theorem 2 shows

[5] One might argue that it would be better to have the supremum over V *outside* the integral over state-actions. We study this a bit further in Appendix A, but we do not pursue this path much more as it does not seem to be as computationally appealing as the current formulation.

that under certain standard conditions, this is indeed a sound procedure. The result would be the following cost functional:

$$c_{2,n}^2(\hat{\mathcal{P}}) = c_{2,\nu_n}^2(\hat{\mathcal{P}}, \mathcal{P}_n) = \frac{1}{n} \sum_{(X_i,A_i)\in\mathcal{D}_n} \sup_{V\in\mathcal{F}} \left| \int \left[\mathcal{P}_n(\mathrm{d}x'|X_i,A_i) - \hat{\mathcal{P}}(\mathrm{d}x'|X_i,A_i) \right] V(x') \right|^2$$
$$= \frac{1}{n} \sum_{(X_i,A_i)\in\mathcal{D}_n} \sup_{V\in\mathcal{F}} \left| V(X_i') - \int \hat{\mathcal{P}}(\mathrm{d}x'|X_i,A_i) V(x') \right|^2 . \tag{9}$$

The output of VAML is

$$\hat{\mathcal{P}} \leftarrow \operatorname*{argmin}_{\mathcal{P}\in\mathcal{M}} c_{2,n}^2(\hat{\mathcal{P}}). \tag{10}$$

To completely specify the algorithm, we have to choose $\mathcal{F}$ and $\mathcal{M}$. We do this in the rest of this section.

2.1 Derivation of $\sup_{f\in\mathcal{F}} \left\langle P - \hat{P}\,,\, f \right\rangle$

In this section, we derive $\sup_{f\in\mathcal{F}} \left\langle P - \hat{P}\,,\, f \right\rangle$ for $P, \hat{P} \in \bar{\mathcal{M}}(\mathcal{X})$ and $f : \mathcal{X} \to \mathbb{R}$. This is similar to what we have in (7) with the difference that instead of working with conditional probabilities, we work with probability distributions defined on $\mathcal{X}$. To distinguish these two cases we use P, $\hat{P}$, and f instead of $\mathcal{P}^*$, $\hat{\mathcal{P}}$, and V.

Let $\phi : \mathcal{X} \to \mathbb{R}^p$ be a feature map, and define the function space $\mathcal{F}_B = \left\{ f_\theta(x) = \phi^\top(x)\theta \,:\, \theta \in \mathbb{R}^p, \|\theta\|_2 \le B \right\}$. In what follows, we may use $\langle \cdot, \cdot \rangle_\Theta$ to denote the inner product in the space of parameters, i.e., $\mathbb{R}^p$, and similarly for $\langle \cdot, \cdot \rangle_\mathcal{X}$. We have

$$\sup_{f_\theta\in\mathcal{F}_B} \left\langle P - \hat{P}\,,\, f \right\rangle_\mathcal{X} = \sup_{f_\theta\in\mathcal{F}_B} \int (P - \hat{P})(\mathrm{d}x)\phi^\top(x)\theta$$
$$= \sup_{\theta\in\mathbb{R}^p, \|\theta\|_2\le B} \left\langle \int (P - \hat{P})(\mathrm{d}x)\phi^\top(x)\,,\, \theta \right\rangle_\Theta$$
$$= B \left\| \int (P - \hat{P})(\mathrm{d}x)\phi^\top(x) \right\|_2 .$$

The last equality is because of the relationship between the ℓ_2-norm and its corresponding inner product, which can be seen by noticing that the Cauchy-Schwarz inequality becomes an equality whenever two vectors are linearly dependent.

This equality shows that minimizing the LHS w.r.t. $\hat{P}$ is equivalent to minimizing the ℓ_2-norm of the projection of $P - \hat{P}$ onto the vector-valued function ϕ. The error in distribution $P - \hat{P}$ does not contribute to the LHS whenever it is orthogonal to the features. This is appealing as it shows that the aspects of the probability distribution that cannot possibly contribute to the expectation of any $f \in \mathcal{F}_B$ should not contribute to the loss that is used for learning the probability distribution P.

To find the minimum, we can follow the gradient descent of the squared norm. Before doing that, we substitute P with the empirical distribution P_n defined based on $\mathcal{D}_n = \{X_i\}_{i=1}^n$. Also suppose that $\mathcal{M} = \left\{ \hat{P}_w \,:\, w \in \mathbb{R}^{p'} \right\}$ in which $\hat{P}_w$ is an exponential family defined by features $\psi : \mathcal{X} \to \mathbb{R}^{p'}$ and parameter $w \in \mathbb{R}^{p'}$, i.e.,

$$\hat{P}_w(\mathrm{d}x) = \frac{\exp\left(\psi^\top(x)w\right)}{\int \exp\left(\psi^\top(x')w\right) \mathrm{d}x'}. \tag{11}$$

For this model, one can verify that

$$\nabla_w \hat{P}_w(x) = \hat{P}_w(x) \left[\psi^\top(x) - \int \hat{P}_w(\mathrm{d}x')\psi^\top(x') \right]. \tag{12}$$

Therefore, for the squared pointwise error we have

$$\begin{aligned}
&\nabla_w \left\| \int \hat{P}_w(\mathrm{d}x)\phi^\top(x) - \frac{1}{n}\sum_{i=1}^n \phi(X_i) \right\|_2^2 = \\
&2\left[\int \hat{P}_w(\mathrm{d}x)\phi^\top(x) - \frac{1}{n}\sum_{i=1}^n \phi(X_i) \right]^\top \int \phi(x)\nabla_w \hat{P}_w(x)\mathrm{d}x = \\
&2\left[\mathbb{E}_{\hat{P}_w}\left[\phi(X)\right] - \mathbb{E}_n\left[\phi(X_i)\right] \right]^\top \left[\mathbb{E}_{\hat{P}_w}\left[\phi(X)\psi^\top(X)\right] - \mathbb{E}_{\hat{P}_w}\left[\phi(X)\right]\mathbb{E}_{\hat{P}_w}\left[\psi^\top(X)\right] \right] = \\
&2\left[\mathbb{E}_{\hat{P}_w}\left[\phi(X)\right] - \mathbb{E}_n\left[\phi(X_i)\right] \right]^\top \mathbf{Cov}_{\hat{P}_w}\left(\phi(X), \psi(X)\right),
\end{aligned}$$

in which we used $\mathbb{E}_n[\cdot]$ to denote the expectation w.r.t. the empirical distribution defined based on $\mathcal{D}_n$, and $\mathbf{Cov}_{\hat{P}_w}(\phi(X), \psi(X))$ as the cross-covariance between $\phi(X)$ and $\psi(X)$ w.r.t. the distribution $\hat{P}_w$.

2.2 The Gradient of $c_{2,n}^2(\hat{\mathcal{P}})$

Following derivations similar to the simpler case of Section 2.1, we can obtain the gradient of (9). Extending (11) to conditional distributions, we choose $\hat{\mathcal{P}} = \hat{\mathcal{P}}_w$ as an exponential family described by features $\phi' : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}^{p'}$ and the weight vector $w \in \mathbb{R}^{p'}$, i.e.,

$$\hat{\mathcal{P}}_w(\mathrm{d}x'|x,a) = \frac{\exp\left(\phi'^\top(x'|x,a)w\right)}{\int \exp\left(\phi'^\top(x''|x,a)w\right)\mathrm{d}x''}\mathrm{d}x'. \tag{13}$$

We consider the case that the value function belongs to the function space $\mathcal{F} = \mathcal{F}_B = \left\{ V_\theta(x) = \phi^\top(x)\theta \,:\, \theta \in \mathbb{R}^p, \|\theta\|_2 \le B \right\}$ with $\phi : \mathcal{X} \to \mathbb{R}^p$ being the feature map.

The choice of linear value function approximator is common in the RL literature [Sutton and Barto, 1998; Szepesvári, 2010]. The features ϕ might either be manually-designed (e.g., [Silver et al., 2007; Liang et al., 2016]) or automatically generated data-dependently, e.g., [Petrik, 2007; Mahadevan and Maggioni, 2007; Parr et al., 2007; Mahadevan and Liu, 2010; Geramifard et al., 2011; Farahmand and Precup, 2012; Böhmer et al., 2013; Milani Fard et al., 2013]. Also note that even some deep neural network-based RL algorithms, such as DQN [Mnih et al., 2015], use a linear output layer to represent the value function. Therefore, whenever their hidden layers are fixed, which means that ϕ would be fixed, they have the same structure as $\mathcal{F}$ here.

Note that in general features $\phi'(x'|x,a)$ of the probability model $\hat{\mathcal{P}}$ are different from the features $\phi(x)$ used to represent the value function. Of course, we may choose them to be related, for example by defining $\phi'(x|x,a) = h(x,a)\phi(x')$ for some function $h(x,a)$.

With similar derivations, we obtain the following theorem.

Theorem 1. *Consider the parameterization (13) of the estimated probability transition kernel $\hat{\mathcal{P}}_w$, and the value function space $\mathcal{F} = \mathcal{F}_B$. The gradient of $c_{2,n}^2(\hat{\mathcal{P}}_w)$ w.r.t. the parameter w is*

$$\begin{aligned}
&\nabla_w c_{2,n}^2(\hat{\mathcal{P}}_w) = \\
&\frac{2B^2}{n}\sum_{i=1}^n \left[\int \hat{\mathcal{P}}_w(\mathrm{d}x'|X_i,A_i)\phi^\top(x') - \phi(X_i') \right]^\top \left[\int \hat{\mathcal{P}}_w(\mathrm{d}x'|X_i,A_i)\phi(x')\phi'^\top(x'|X_i,A_i) - \right. \\
&\qquad\qquad \left. \int \hat{\mathcal{P}}_w(\mathrm{d}x'|X_i,A_i)\phi(x') \int \hat{\mathcal{P}}_w(\mathrm{d}x'|X_i,A_i)\phi'^\top(x'|X_i,A_i) \right] = \\
&\frac{2B^2}{n}\sum_{i=1}^n \left[\mathbb{E}_{X'\sim\hat{\mathcal{P}}_w(\cdot|X_i,A_i)}\left[\phi(X')\right] - \phi(X_i') \right]^\top \mathbf{Cov}_{X'\sim\hat{\mathcal{P}}_w(\cdot|X_i,A_i)}\left(\phi(X'), \phi'(X'|X_i,A_i)\right).
\end{aligned}$$

The loss function has two main terms. The first term computes the difference between the empirical average of the *value* function features $\phi(X_i')$ and its expectation $\mathbb{E}_{X'\sim\hat{\mathcal{P}}_w(\cdot|X_i,A_i)}[\phi(X')]$ according to the model with parameter w. So this term encourages finding a model that "matches" according to the features ϕ of the value

function space $\mathcal{F}$. The other term is the cross-covariance between the features ϕ of the value function and the features ϕ' of the model. This term might be seen as a weighting term for the first one.

It is instructive to compare this gradient with the gradient of the negative log-loss (1) with the same exponential model, which is

$$\frac{1}{n}\sum_{i=1}^{n}\left[\mathbb{E}_{X'\sim\hat{\mathcal{P}}_w(\cdot|X_i,A_i)}\left[\phi'^{\top}(X'|X_i,A_i)\right]-\phi'^{\top}(X_i'|X_i,A_i)\right]. \tag{14}$$

One can interpret this by saying that MLE is trying to find $\hat{\mathcal{P}}_w$ such that the expected value of *model* features ϕ' evaluated at the next-state matches the empirical values. The matching is based on model features ϕ' and not the value features ϕ. One might see that for finite MDPs with exact representation of both value function and the model (i.e., lookup tables for both ϕ and ϕ'), the asymptotic solutions of VAML and MLE are the same, but since their gradients on the way are not, they approach that point differently.

Working with exponential family can be computationally expensive, no matter whether we use MLE or VAML. To begin with, even sampling from (13) requires the computation of the normalizing factor (i.e., partition function), which except in special cases such as for Gaussian distributions, does not have a closed-form solution. The second issue is that to compute the gradients required for MLE or VAML, we require to estimate certain expectations (and covariance matrices). This can be challenging too. On the positive side, however, these computations, have been the subject of many years of research; and there are already many methods, exact or approximate, to evaluate this general family of probability distributions, e.g., various Monte Carlo estimates or variational methods [MacKay, 2003; Goodfellow et al., 2016]. Moreover, we may not really need to have a very accurate estimate of the gradients in VAML or MLE in order to minimize the cost function. It might be enough to only have a few samples from the learned distribution to estimate the "direction" of the gradient correctly. This is one of the ideas behind Contrastive Divergence [Carreira-Perpinan and Hinton, 2005], which we can use for VAML too. In our empirical studies (Section 5), only a small number of Monte Carlo samples is sufficient to produce good results, e.g., $m = 5$.

3 STATISTICAL ANALYSIS OF VAML

We provide a finite sample error upper bound showing that VAML is indeed a sound algorithm in the sense that the minimizer $\hat{\mathcal{P}}$ of the empirical loss $c_{2,n}^2$, if attained, has a small error (7), given enough data points n and under standard capacity condition on the function spaces $\mathcal{M}$. The result of this section is not limited to exponential models of $\mathcal{M}$.

Consider a family of probability distributions $\mathcal{M}_0$ and a pseudo-norm $J : \mathcal{M}_0 \to [0,\infty)$. Let the set $\mathcal{M}$ used by VAML be a subset $M = M_B = \{\,\mathcal{P}\in\mathcal{M}_0 : J(\mathcal{P})\le B\,\}$ for some $B > 0$. We can think of J of a measure of complexity of functions in $\mathcal{M}_0$, so $\mathcal{M}$ would be a ball with a fixed radius B w.r.t. J. If $\mathcal{M}_0$ is defined based on an RKHS, we can think of J as the inner product norm of the RKHS. We have the following assumptions on the metric entropy (logarithm of the covering number) of $\mathcal{M}$.

Assumption A1 (Capacity of Function Space) For $B > 0$, let $M = M_B = \{\,\mathcal{P}\in\mathcal{M}_0 : J(\mathcal{P})\le B\,\}$. There exist constants $C > 0$ and $0 < \alpha < 1$ such that for any $u, B > 0$ and all sequence $z_1,\dots,z_n\in\mathcal{Z}$, the following metric entropy condition is satisfied:

$$\log\mathcal{N}\left(u,\mathcal{M}_B,L_2(P_{z_{1:n}})\right)\le C\left(\frac{B}{u}\right)^{2\alpha}.$$

Metric entropy of $\mathcal{M}_B$ is a measure of the size of $\mathcal{M}_B$, and roughly speaking, it is the logarithm of the minimum number of balls with radius u that are required to completely cover $\mathcal{M}_B$. In general, it is more difficult to estimate a function when the metric entropy grows fast when u decreases. Here $L_2(P_{z_{1:n}})$ is the L_2-norm defined w.r.t. the empirical measure (cf. e.g., Section 9.3 of [Györfi et al., 2002]; see also Appendix B). For many examples of the metric entropy results, refer to [van de Geer, 2000; Györfi et al., 2002; Giné and Nickl, 2015]. After stating this assumption, we are ready to state the theorem.

Theorem 2. *Given a dataset $\mathcal{D}_n = \{(X_i,A_i,X_i')\}_{i=1}^n$ with independent and identically distributed samples $(X_i,A_i)\sim\nu$, with $X_i'\sim\mathcal{P}^*(\cdot|X_i,A_i)$, let $\hat{\mathcal{P}}$ be the minimizer of the VAML algorithm, i.e., $\hat{\mathcal{P}}\leftarrow$*

argmin$_{\mathcal{P}\in\mathcal{M}} c_{2,n}^2(\hat{\mathcal{P}})$, *with the previously specified choice of value function space* $\mathcal{F}$. *Let Assumption A1 hold. Furthermore, assume that* $\sup_{x\in\mathcal{X}} \|\phi(x)\|_\infty \le 1$ *and* $\sup_{x\in\mathcal{X}} \|\phi(x)\|_2 \le 1$. *Fix* $\delta > 0$. *There exists a constant* $c > 0$ *such that*

$$\mathbb{E}\left[\sup_{V\in\mathcal{F}} \left|(\hat{\mathcal{P}}_Z - \mathcal{P}_Z^*)V\right|^2\right] \le \inf_{\mathcal{P}\in\mathcal{M}} \mathbb{E}\left[\sup_{V\in\mathcal{F}} |(\mathcal{P}_Z - \mathcal{P}_Z^*)V|^2\right] + c(1+B^\alpha)p\sqrt{\frac{\log(p/\delta)}{n}} + \frac{16\log(4/\delta)}{3n},$$

with probability at least $1-\delta$.

This upper bound shows the usual model (or function) approximation error (first term) and the estimation error (second and third terms). The dominant term in the estimation error behaves $O(n^{-1/2})$, which is the usual behaviour of the supremum of the empirical process for models that are not very large. The size of the function space $\mathcal{M}$, specified by B in Assumption A1, appears in the bound. We also see the effect of size of ϕ vector, specifying the value function space $\mathcal{F}$, appears linearly. We believe that this dependence on p is suboptimal, and can be improved further.

Maybe more interesting is the effect of the model approximation error. The bound shows that the error of the estimated $\hat{\mathcal{P}}$ is comparable to the error of the best choice in the model class $\mathcal{M}$, i.e., $\inf_{\mathcal{P}\in\mathcal{M}} \mathbb{E}\left[\sup_{V\in\mathcal{F}} |(\mathcal{P}_Z - \mathcal{P}_Z^*)V|^2\right]$. This is reassuring since VAML was motivated by the fact that the important property of an estimated model $\hat{\mathcal{P}}$ should be that $\left|\left\langle \hat{\mathcal{P}}(\cdot|z) - \mathcal{P}^*(\cdot|z)\,,\, V\right\rangle\right|$ is small only for $V \in \mathcal{F}$ that might be encountered by the algorithm, and not necessarily for all possible value functions, which cannot even be represented by the value-based algorithm.

One could obtain faster estimation error (i.e., $O(n^{-1})$) by studying the modulus of the continuity of the empirical process instead of the supremum of the empirical process, as we do here. We decided not to provide such a result because of two reasons. The first is that faster rates require increasing the constant in front of the approximation error (it would not be 1 anymore). In the regime that we have an approximation error ($\mathcal{P}^* \notin \mathcal{M}$), which is the regime that can make VAML superior to MLE, this would lead to asymptotically worse results. The other reason is to simplify the proofs and making them more accessible.[6]

A few other short remarks are in order. The first is that this is a statistical guarantee, and is valid under the condition that $\min_{\mathcal{P}\in\mathcal{M}} c_{2,n}^2(\hat{\mathcal{P}})$ is indeed attained. We have not shown that this minimum can be achieved by following the gradient of Theorem 1, especially since the VAML's objective is not necessarily convex. Another remark is that we do not analyze the effect of the model estimation error on the quality of the policy obtained by $\mathsf{Planner}(\hat{\mathcal{P}})$. Ávila Pires and Szepesvári [2016] provide such a policy error bound.

Proof of Theorem 2. To simplify the equations, we define a few notations. The pointwise loss function for $\mathcal{P}$ is

$$l(z;\mathcal{P}) = \|(\mathcal{P}_z - \mathcal{P}_z^*)\phi\|_2^2\,.$$

The expected loss is

$$L(\mathcal{P}) = \mathbb{E}\left[l(Z;\mathcal{P})\right],$$

in which $Z \sim \nu$. Given the dataset $\mathcal{D}_n = \{(X_i, A_i, X_i')\}_{i=1}^n$ with $Z_i = (X_i, A_i) \sim \nu$ and $X_i' \sim \mathcal{P}_{Z_i}^*$, we define the "ideal" empirical loss as

$$L_n(\mathcal{P}) = \mathbb{E}_n\left[l(Z;\mathcal{P})\right] = \frac{1}{n}\sum_{i=1}^n l(Z_i;\mathcal{P}).$$

Note that this is not the empirical loss that is minimized by the algorithm as the algorithm does not have access to $\mathcal{P}_{Z_i}^*$.

For any z and a corresponding $X' \sim \mathcal{P}_z^*$, we denote

$$\hat{l}(z, X';\mathcal{P}) = \|\mathcal{P}_z\phi - \phi(X')\|_2^2\,.$$

[6] Because of certain steps of the proof, we could not use already available results such as Theorem 3.3 of Bartlett et al. [2005].

If X' is clear from the context, we may use $\hat{l}(z;\mathcal{P})$ to refer to $\hat{l}(z,X';\mathcal{P})$. With this notation, we define the empirical loss, given $\mathcal{D}_n$, as

$$\hat{L}_n(\mathcal{P}) = \mathbb{E}_n\left[\hat{l}(Z_i,X_i';\mathcal{P})\right] = \frac{1}{n}\sum_{i=1}^n \hat{l}(Z_i,X_i';\mathcal{P}).$$

We also define

$$\hat{L}(\mathcal{P}) = \mathbb{E}\left[\hat{l}(Z;X';\mathcal{P})\right].$$

Note that by the definition of the VAML algorithm, we have[7]

$$\hat{\mathcal{P}} \leftarrow \operatorname*{argmin}_{\mathcal{P}\in\mathcal{M}} \hat{L}_n(\mathcal{P}). \tag{15}$$

For any $\mathcal{P}$, we have

$$\begin{aligned}\hat{l}(z;\mathcal{P}) &= \|\mathcal{P}_z\phi - \phi(X')\|_2^2 = \|\mathcal{P}_z\phi - \mathcal{P}_z^*\phi + \mathcal{P}_z^*\phi - \phi(X')\|_2^2 \\ &= \|(\mathcal{P}_z - \mathcal{P}_z^*)\phi\|_2^2 + \|\mathcal{P}_z^*\phi - \phi(X')\|_2^2 + 2\left\langle (\mathcal{P}_z - \mathcal{P}_z^*)\phi\,,\, \mathcal{P}_z^*\phi - \phi(X')\right\rangle.\end{aligned}$$

By reordering and taking summation over $\mathcal{D}_n$, we get

$$L_n(\mathcal{P}) = \mathbb{E}_n\left[\left\|(\mathcal{P}_{Z_i} - \mathcal{P}_{Z_i}^*)\phi\right\|_2^2\right] = \underbrace{\mathbb{E}_n\left[\|\mathcal{P}_{Z_i}\phi - \phi(X_i')\|_2^2\right]}_{=\hat{L}_n(\mathcal{P})} - \underbrace{\mathbb{E}_n\left[\left\|\mathcal{P}_{Z_i}^*\phi - \phi(X_i')\right\|_2^2\right]}_{\triangleq e_\sigma} + \underbrace{2\frac{1}{n}\sum_{i=1}^n \left\langle (\mathcal{P}_{Z_i}^* - \mathcal{P}_{Z_i})\phi\,,\, \mathcal{P}_{Z_i}^*\phi - \phi(X_i')\right\rangle}_{\triangleq e_I(\mathcal{P})}.$$

Consider any $\tilde{\mathcal{P}} \in \mathcal{M}$. We may upper bound the true loss of $\hat{\mathcal{P}}$, that is $L(\hat{\mathcal{P}})$, through the following sequence of inequalities:

$$\begin{aligned}L(\hat{\mathcal{P}}) &= L_n(\hat{\mathcal{P}}) + L(\hat{\mathcal{P}}) - L_n(\hat{\mathcal{P}}) \\ &= \hat{L}_n(\hat{\mathcal{P}}) - e_\sigma + 2e_I(\hat{\mathcal{P}}) + \left[L(\hat{\mathcal{P}}) - L_n(\hat{\mathcal{P}})\right] \\ &\overset{(i)}{\le} \hat{L}_n(\tilde{\mathcal{P}}) - e_\sigma + 2e_I(\hat{\mathcal{P}}) + \left[L(\hat{\mathcal{P}}) - L_n(\hat{\mathcal{P}})\right] \\ &= \left[\hat{L}_n(\tilde{\mathcal{P}}) - e_\sigma + 2e_I(\tilde{\mathcal{P}})\right] + 2e_I(\hat{\mathcal{P}}) - 2e_I(\tilde{\mathcal{P}}) + \left[L(\hat{\mathcal{P}}) - L_n(\hat{\mathcal{P}})\right] \\ &\le L_n(\tilde{\mathcal{P}}) + 4\sup_{\mathcal{P}\in\mathcal{M}}|e_I(\mathcal{P})| + \sup_{\mathcal{P}\in\mathcal{M}}|L(P) - L_n(P)| \\ &= L(\tilde{\mathcal{P}}) + \left[L_n(\tilde{\mathcal{P}}) - L(\tilde{\mathcal{P}})\right] + 4\sup_{\mathcal{P}\in\mathcal{M}}|e_I(\mathcal{P})| + \sup_{\mathcal{P}\in\mathcal{M}}|L(P) - L_n(P)| \\ &\le L(\tilde{\mathcal{P}}) + 2\sup_{\mathcal{P}\in\mathcal{M}}|L(P) - L_n(P)| + 4\sup_{\mathcal{P}\in\mathcal{M}}|e_I(\mathcal{P})|.\end{aligned} \tag{16}$$

Here the step (i) is because of the optimizer property of $\hat{\mathcal{P}}$.

We need to have upper bounds for $\sup_{\mathcal{P}\in\mathcal{M}}|L(P) - L_n(P)|$ and $\sup_{\mathcal{P}\in\mathcal{M}}|e_I(\mathcal{P})|$. Propositions 3 and 4, to be proved soon, provide these.

Fix $\delta > 0$. Proposition 3 states that for a constant $c_1 > 0$, with probability at least $1 - \delta/2$, it holds that

$$\sup_{\mathcal{P}\in\mathcal{M}}|e_I(\mathcal{P})| \le c_1(1 + B^\alpha)p\sqrt{\frac{\log(2p/\delta)}{n}}.$$

[7] We assume that the minimizer exists and is attained by a $\hat{\mathcal{P}}$ within $\mathcal{M}$.

Proposition 4 states that for a constant $c_2 > 0$, with probability at least $1 - \delta/2$, it holds that

$$\sup_{\mathcal{P}\in\mathcal{M}} |L(P) - L_n(P)| \le \frac{c_2 B^\alpha}{\sqrt{n}} + 2\sqrt{\frac{2\log(4/\delta)}{n}} + \frac{16\log(4/\delta)}{3n}.$$

We substitute these two inequalities in (16) to obtain that for any $\tilde{\mathcal{P}} \in \mathcal{M}$, we have

$$L(\hat{\mathcal{P}}) \le L(\tilde{\mathcal{P}}) + c_3(1 + B^\alpha)p\sqrt{\frac{\log(p/\delta)}{n}} + \frac{16\log(4/\delta)}{3n},$$

with probability at least $1 - \delta$. Taking $\tilde{\mathcal{P}}$ to be the minimizer of $L(\mathcal{P})$ within $\mathcal{M}$ finishes the proof. □

Proposition 3. *Under the same conditions as in Theorem 2, there exists a constant $c > 0$ such that for any fixed $\delta > 0$, we have*

$$\sup_{\mathcal{P}\in\mathcal{M}} \left| \frac{1}{n}\sum_{i=1}^{n} \left\langle (\mathcal{P}^*_{Z_i} - \mathcal{P}_{Z_i})\phi \,,\, \mathcal{P}^*_{Z_i}\phi - \phi(X'_i) \right\rangle \right| \le c(1 + B^\alpha)p\sqrt{\frac{\log(p/\delta)}{n}},$$

with probability at least $1 - \delta$.

Proof. For $i = 1, \dots, n$, define the vector-valued random variables $W_i = \mathcal{P}^*_{Z_i}\phi - \phi(X'_i) \in \mathbb{R}^p$. Notice that because $X'_i \sim \mathcal{P}^*_{Z_i}(\cdot)$, we have $\mathbb{E}[W_i] = 0$. We refer to each component of W_i by $W_i^{(j)}$ for $j = 1, \dots, p$, i.e., $W_i = [W_i^{(1)} \cdots W_i^{(p)}]$.

Define the function spaces

$$\mathcal{H} = \{\, h(z;\mathcal{P}) = (\mathcal{P}^* - \mathcal{P})\phi \in \mathbb{R}^p \,:\, \mathcal{P} \in \mathcal{M} \,\},$$
$$\mathcal{H}^{(j)} = \left\{\, h^{(j)}(z;\mathcal{P}) = (\mathcal{P}^* - \mathcal{P})\phi_j \in \mathbb{R} \,:\, \mathcal{P} \in \mathcal{M} \right\}. \qquad (j = 1, \dots, p)$$

The vector-valued $h(z;\mathcal{P})$ is identified as $[h^{(1)}(z;\mathcal{P}) \cdots h^{(p)}(z;\mathcal{P})]$.

We upper bound the probability that the supremum of the inner product be larger than $t > 0$: Let the positive sequence $(\eta_j)_{j=1}^p$ be such that $\sum_{j=1}^p \eta_j \le 1$. We provide a concrete choice for this sequence shortly. We have

$$\begin{aligned}
\mathbb{P}\left\{ \sup_{h\in\mathcal{H}} \left| \frac{1}{n}\sum_{i=1}^{n} \langle\, h(Z_i) \,,\, W_i \,\rangle \right| > t \right\} &= \mathbb{P}\left\{ \sup_{h\in\mathcal{H}} \left| \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} h^{(j)}(Z_i) W_i^{(j)} \right| > t \right\} \\
&\le \sum_{j=1}^{p} \mathbb{P}\left\{ \sup_{h^{(j)}\in\mathcal{H}^{(j)}} \left| \frac{1}{n}\sum_{i=1}^{n} h^{(j)}(Z_i) W_i^{(j)} \right| > \eta_j t \right\}. \qquad (17)
\end{aligned}$$

We use Theorem 19.1 of Györfi et al. [2002], quoted in Appendix C as Lemma 9, to upper bound each of the terms in the RHS (cf. Lemma 3.2 of van de Geer [2000]). We now verify the conditions of that lemma.

Because $|W_i^{(j)}| = |\mathcal{P}^*_z\phi - \phi(X')| \le 2$, we may choose $L = 2$. Also because $|h^{(j)}(z)| = |(\mathcal{P}^*_z - \mathcal{P}_z)\phi_j| \le 2$, we may choose $R = 2$. The lemma also requires that

$$\sqrt{n}(\eta_j t) \ge 36(RL),$$

so if we choose $\eta_j = \frac{1}{p}$, we get that it is sufficient to have

$$t \ge \frac{144p}{\sqrt{n}}. \qquad (18)$$

For the metric entropy condition, we have

$$\sqrt{n}(\eta_j t) \ge 48\sqrt{2}L \int_0^R \sqrt{\log \mathcal{N}(u, \mathcal{H}^{(j)}, L_2(P(z_{1:n}))}\mathrm{d}u, \qquad (19)$$

in which $L_2(P(z_{1:n}))$ is the L_2-norm based on the empirical measure defined for any choice of sequence $z_{1:n} \triangleq (z_1, \ldots, z_n) \subset \mathcal{Z}$.

To compute this integral, we need to relate the covering number of $\mathcal{H}^{(j)}$, i.e., $\mathcal{N}(u, \mathcal{H}^{(j)}, L_2(P(z_{1:n}))$, to the covering number of $\mathcal{M}$. Proposition 5 shows that for all $j = 1, \ldots, p$,

$$\mathcal{N}\left(u, \mathcal{H}^{(j)}, L_2(P_{z_{1:n}})\right) \leq \mathcal{N}\left(u, \mathcal{M}, L_2(P_{z_{1:n}})\right).$$

Under our assumption on the covering number of $\mathcal{M}$, we obtain that

$$\int_0^R \sqrt{\log \mathcal{N}(u, \mathcal{H}^{(j)}, L_2(P(z_{1:n}))} \mathrm{d}u \leq \frac{\sqrt{C} B^\alpha}{1-\alpha} R^{1-\alpha},$$

so by the choice of $R = 2$ and $\eta_j = 1/p$, we see that it is sufficient to satisfy

$$t \geq \frac{c_1(\alpha) B^\alpha p}{\sqrt{n}}, \tag{20}$$

for some choice of $c_1(\alpha) > 0$, which is independent of B, n, or p, to satisfy the metric entropy condition (19).

Upon the satisfaction of (18) and (20), we may apply Lemma 9 to each term in the RHS of (17) to get that

$$\mathbb{P}\left\{\sup_{h \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^n \langle\, h(Z_i)\,,\, W_i \,\rangle \right| > t \right\} \leq \sum_{j=1}^p 5 \exp\left(-\frac{n(\frac{1}{p}t)^2}{2304(2 \times 2)^2}\right).$$

Setting the LHS to a fix $\delta > 0$ and solving for t, we obtain that

$$t \leq 192 p \sqrt{\frac{\log(5p/\delta)}{n}}, \tag{21}$$

with probability at least $1 - \delta$. Conditions (18), (20), and (21) imply the desired result. □

Proposition 4. *Under the same conditions as in Theorem 2, there exists a constant $c > 0$ such that for any fixed $\delta > 0$, we have*

$$\sup_{\mathcal{P} \in \mathcal{M}} \left| \mathbb{E}\left[\|(\mathcal{P}_Z - \mathcal{P}^*_Z)\phi\|_2^2 \right] - \frac{1}{n} \sum_{i=1}^n \|(\mathcal{P}_{Z_i} - \mathcal{P}^*_{Z_i})\phi\|_2^2 \right| \leq \frac{cB^\alpha}{\sqrt{n}} + 2\sqrt{\frac{2\log(2/\delta)}{n}} + \frac{16\log(2/\delta)}{3n},$$

with probability at least $1 - \delta$.

Proof. Define the function space

$$\mathcal{G} = \left\{ g(z; \mathcal{P}) = \|(\mathcal{P}_z - \mathcal{P}^*_z)\phi\|_2^2 \,:\, \mathcal{P} \in \mathcal{M} \right\}.$$

Based on the assumption that $\sup_{x \in \mathcal{X}} \|\phi(x)\|_2 \leq 1$, we have that for any $g \in \mathcal{G}$

$$\begin{aligned}
\sup_z |g(z)| &= \sup_z \left\| \int (\mathcal{P}_z - \mathcal{P}^*_z)(\mathrm{d}y)\phi(y) \right\|_2 \leq \sup_z \left[\left\| \int \mathcal{P}_z(\mathrm{d}y)\phi(y) \right\|_2 + \left\| \int \mathcal{P}^*_z(\mathrm{d}y)\phi(y) \right\|_2 \right] \\
&\leq \sup_z \left[\int \mathcal{P}_z(\mathrm{d}y) \|\phi(y)\|_2 + \int \mathcal{P}^*_z(\mathrm{d}y) \|\phi(y)\|_2 \right] \\
&= \sup_{x \in \mathcal{X}} \|\phi(x)\|_2 \left[\int \mathcal{P}_z(\mathrm{d}y) + \int \mathcal{P}^*_z(\mathrm{d}y) \right] \leq 2.
\end{aligned}$$

So functions in $\mathcal{G}$ are bounded by $B = 2$. Because of this, their variance is also bounded: $\mathrm{Var}\left[g(Z; P)\right] \leq \mathbb{E}\left[g^2(Z; P)\right] \leq B^2 = 4$. We now apply Lemma 8 with the choice of $\alpha = 1$. For a fixed $\delta > 0$, we get that

$$\sup_{\mathcal{P} \in \mathcal{M}} \left| \mathbb{E}\left[\|(\mathcal{P}_Z - \mathcal{P}^*_Z)\phi\|_2^2 \right] - \frac{1}{n} \|(\mathcal{P}_{Z_i} - \mathcal{P}^*_{Z_i})\phi\|_2^2 \right| \leq 4\mathbb{E}\left[R_n(\mathcal{G})\right] + 2\sqrt{\frac{2\log(2/\delta)}{n}} + \frac{16\ln(2/\delta)}{3n} \tag{22}$$

holds with probability at least $1-\delta$.

To upper bound the Rademacher complexity of the function space $\mathcal{G}$, we use Dudley's integral to relate the Rademacher complexity of $\mathcal{G}$ to the covering number of $\mathcal{G}$; we then use Proposition 5 to relate the covering number of $\mathcal{G}$ to that of $\mathcal{M}$; and finally, we use our assumption on the covering number of $\mathcal{M}$.

$$\begin{aligned}\mathbb{E}\left[R_n(\mathcal{G})\right] &\leq \frac{4\sqrt{2}}{\sqrt{n}}\mathbb{E}\left[\int_0^{\mathrm{diam}(\mathcal{G})}\sqrt{\log 2\mathcal{N}\left(u,\mathcal{G},L_2(P_{Z_{1:n}})\right)}\mathrm{d}u\right]\\ &\leq \frac{4\sqrt{2}}{\sqrt{n}}\mathbb{E}\left[\int_0^{\mathrm{diam}(\mathcal{M})}\sqrt{\log 2\mathcal{N}\left(u/2,\mathcal{M},L_2(P_{Z_{1:n}})\right)}\mathrm{d}u\right]\\ &\leq \frac{cB^\alpha}{\sqrt{n}},\end{aligned}$$

for some constant $c>0$.[8] This upper bound on the Rademacher complexity and (22) lead to the desired result. □

Proposition 5. *Let*

$$\begin{aligned}\mathcal{H}^{(j)} &= \left\{\, h(z;\mathcal{P}) = (\mathcal{P}^* - \mathcal{P})\phi_j \,:\, \mathcal{P}\in\mathcal{M}\,\right\}, \qquad (j=1,\dots,p)\\ \mathcal{G} &= \left\{\, g(z;\mathcal{P}) = \|(\mathcal{P}_z - \mathcal{P}_z^*)\phi\|_2^2 \,:\, \mathcal{P}\in\mathcal{M}\,\right\}.\end{aligned}$$

Assume that for any sequence $z_{1:n}=(z_1,\dots,z_n)\subset\mathcal{Z}$, the empirical covering number $\mathcal{N}(u,\mathcal{M},L_2(P_{z_{1:n}}))<\infty$ for all $u>0$.

Part 1) Assume that $\|\phi_j\|_\infty \leq 1$ for all $j=1,\dots,p$. We then have

$$\mathcal{N}\left(u,\mathcal{H}^{(j)},L_2(P_{z_{1:n}})\right) \leq \mathcal{N}\left(u,\mathcal{M},L_2(P_{z_{1:n}})\right). \qquad (j=1,\dots,p)$$

Part 2) Assume that $\sup_{x\in\mathcal{X}}\|\phi\|_2\leq 1$. We then have

$$\mathcal{N}\left(u,\mathcal{G},L_2(P_{z_{1:n}})\right) \leq \mathcal{N}\left(\frac{u}{2},\mathcal{M},L_2(P_{z_{1:n}})\right).$$

Proof. First we prove the covering number result for the function space $\mathcal{H}^{(j)}$ for any $j=1,\dots,p$. Let $h_1,h_2\in\mathcal{H}^{(j)}$ with their corresponding $\mathcal{P}^{(1)},\mathcal{P}^{(2)}\in\mathcal{M}$. We have

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n |h_1(z_i)-h_2(z_i)|^2 &= \frac{1}{n}\sum_{i=1}^n \left|\left[\mathcal{P}^{(1)}(\cdot|z_i)-\mathcal{P}^{(2)}(\cdot|z_i)\right]\phi_j\right|^2\\ &= \frac{1}{n}\sum_{i=1}^n \left|\int\left[\mathcal{P}^{(1)}(\mathrm{d}y|z_i)-\mathcal{P}^{(2)}(\mathrm{d}y|z_i)\right]\phi_j(y)\right|^2\\ &\leq \frac{1}{n}\sum_{i=1}^n \int\left[\mathcal{P}^{(1)}(\mathrm{d}y|z_i)-\mathcal{P}^{(2)}(\mathrm{d}y|z_i)\right]^2\phi_j^2(y)\\ &\leq \sup_x \phi_j^2(x)\,\frac{1}{n}\sum_{i=1}^n\int\left|\mathcal{P}^{(1)}(\mathrm{d}y|z_i)-\mathcal{P}^{(2)}(\mathrm{d}y|z_i)\right|^2.\end{aligned}$$

Under the assumption that $\|\phi_j\|_\infty\leq 1$, this entails that a u-cover of $\mathcal{M}$ w.r.t. $L_2(P(z_{1:n}))$ is also a u-cover of $\mathcal{H}^{(j)}$ w.r.t. the empirical norm on $\mathcal{H}^{(j)}$.

To prove the second part, we consider $g_1,g_2\in\mathcal{G}$ with their corresponding $\mathcal{P}^{(1)},\mathcal{P}^{(2)}\in\mathcal{M}$. We have

[8] Here our specific version of Dudley's integral is from Theorem 2.3.7 of Giné and Nickl [2015] and we use it similar to the argument in the proof of Theorem 3.5.1 and the comments after that. Or one may use Theorem A.7 by Bartlett et al. [2005] (originally from Dudley) and note that the upper bound of the integral does not need to go up to infinity when the function space is bounded in the norm.

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}|g_1(z_i)-g_2(z_i)|^2 &= \frac{1}{n}\sum_{i=1}^{n}\left|\left\|(\mathcal{P}^{(1)}_{z_i}-\mathcal{P}^{*}_{z_i})\phi\right\|_2^2-\left\|(\mathcal{P}^{(2)}_{z_i}-\mathcal{P}^{*}_{z_i})\phi\right\|_2^2\right|^2\\ &= \frac{1}{n}\sum_{i=1}^{n}\left|\left\langle\,(\mathcal{P}^{(1)}_{z_i}+\mathcal{P}^{(2)}_{z_i}-2\mathcal{P}^{*}_{z_i})\phi\,,\,(\mathcal{P}^{(1)}_{z_i}-\mathcal{P}^{(2)}_{z_i})\phi\,\right\rangle\right|^2\\ &\le \frac{1}{n}\sum_{i=1}^{n}\left\|(\mathcal{P}^{(1)}_{z_i}+\mathcal{P}^{(2)}_{z_i}-2\mathcal{P}^{*}_{z_i})\phi\right\|_2^2\left\|(\mathcal{P}^{(1)}_{z_i}-\mathcal{P}^{(2)}_{z_i})\phi\right\|_2^2.\end{aligned}$$

To upper bound the first multiplicative terms on the RHS, notice that due to the convexity of the norm and Jensen's inequality, for any $\mathcal{P}$ we have

$$\|\mathcal{P}_z\phi\|_2^2 \le \mathcal{P}_z\,\|\phi\|_2^2 = \int \mathcal{P}(\mathrm{d}y|z)\,\|\phi(y)\|_2^2 \le \sup_{x\in\mathcal{X}}\|\phi(x)\|_2^2\,.$$

For the second multiplicative terms on the RHS, we again use Jensen's inequality to obtain

$$\begin{aligned}\left\|(\mathcal{P}^{(1)}_{z}-\mathcal{P}^{(2)}_{z})\phi\right\|_2^2 &= \sum_{j=1}^{p}\left|\int\left[\mathcal{P}^{(1)}(\mathrm{d}y|z)-\mathcal{P}^{(2)}(\mathrm{d}y|z)\right]\phi_j(y)\right|^2\\ &\le \sup_{x\in\mathcal{X}}\|\phi(y)\|_2^2\int\left|\mathcal{P}^{(1)}(\mathrm{d}y|z)-\mathcal{P}^{(2)}(\mathrm{d}y|z)\right|^2.\end{aligned}$$

Therefore, we have

$$\frac{1}{n}\sum_{i=1}^{n}|g_1(z_i)-g_2(z_i)|^2 \le 4\sup_x\|\phi(x)\|^4\;\frac{1}{n}\sum_{i=1}^{n}\int\left|\mathcal{P}^{(1)}_{z}(\mathrm{d}y|z)-\mathcal{P}^{(2)}(\mathrm{d}y|z)\right|^2.$$

Similar to the previous case, under the assumption that $\sup_{x\in\mathcal{X}}\|\phi(x)\|_2 \le 1$, this entails that a u-cover of $\mathcal{M}$ w.r.t. $L_2(P(z_{1:n}))$ is also a $2u$-cover of $\mathcal{G}$ w.r.t. the same empirical norm. □

4 ON THE POLICY APPROXIMATION ERROR OF VAML AND MLE

In this section, we develop some intuitions on the policy approximation error of VAML and MLE. We do this through analyzing some simple, but illustrative, examples for which we can analytically find the solutions of these methods. To study policy approximation error we focus on the population version of the loss functions. Also for most of this section, we only study the unconditional distribution estimation problem.

Let us assume that the true model is $P^* \in \bar{\mathcal{M}}$ and we want to find a $P \in \mathcal{M} \subset \bar{\mathcal{M}}$ such that the loss function that matters for computing the expected value function is small. As argued in Section 2, we only know that $V \in \mathcal{F}$, but the exact V is not known, so we use the following loss function to compare the behaviour of different estimators:

$$c_{\mathrm{VAML}}(P,P^*) = \sup_{V\in\mathcal{F}}\int[P(\mathrm{d}x')-P^*(\mathrm{d}x')]\,V(x'),$$

which is the unconditional version of (7). The estimators are

$$P_{\mathrm{VAML}} \leftarrow \operatorname*{argmin}_{P\in\mathcal{M}}\sup_{V\in\mathcal{F}}\int[P(\mathrm{d}x')-P^*(\mathrm{d}x')]\,V(x'),$$

and

$$P_{\mathrm{MLE}} \leftarrow \operatorname*{argmin}_{P\in\mathcal{M}}\mathsf{KL}(P^*||P).$$

Evidently, the minimizer of a loss function is not worse, if not better, than a minimizer of another loss function when the evaluation is based on the original loss. In our case, by the very definition of the minimizer, assuming that it is attained, we have

$$c_{\mathrm{VAML}}(P_{\mathrm{VAML}},P^*) \le c_{\mathrm{VAML}}(P_{\mathrm{MLE}},P^*).$$

The interesting question is whether MLE might perform as well as VAML, i.e., we have an equality.

Suppose that all $V \in \mathcal{F}$ are $V_{\max}$-bounded. From (4) and (5), we have

$$c_{\text{VAML}}(P_{\text{MLE}}, P^*) \leq V_{\max} \|P_{\text{VAML}} - P^*\|_1 \leq V_{\max} \sqrt{2\mathsf{KL}(P^*||P_{\text{MLE}})}. \tag{23}$$

If the model space $\mathcal{M}$ is rich enough such that $P^* \in \mathcal{M}$, by choosing $P_{\text{MLE}} = P^*$, we have $\mathsf{KL}(P^*||P_{\text{MLE}}) = 0$. So if the true model is within the model space (we are in the realizable model learning setting), there is no difference between using VAML or MLE from the model approximation error viewpoint (the estimation errors might behave differently though. But that is not the subject of this section).

Of course, the realizability assumption of $P^* \in \mathcal{M}$ might be unrealistic, especially for parametric models. When $P^* \notin \mathcal{M}$, we have $\mathsf{KL}(P^*||P_{\text{MLE}}) > 0$. We may still use (23) to provide a nonzero upper bound on $c_{\text{VAML}}(P_{\text{MLE}}, P^*)$.

Interestingly, even though we might have a model approximation error measured according to the KL-divergence, the VAML loss $c_{\text{VAML}}(P_{\text{MLE}}, P^*)$ might still be zero. To see this, consider the simplistic case that the value function space has only bounded constant functions, i.e., $\mathcal{F} = \{ V_c(x) = c : -V_{\max} \leq c \leq V_{\max} \}$. In this case, for any probability distribution P, including P_{MLE} and P^*, we have that for $V_c \in \mathcal{F}$, the expectation $\int P(\mathrm{d}x')V(x') = c$. So the supremum of their difference is zero too, that is, $c_{\text{VAML}}(P_{\text{MLE}}, P^*) = 0$.

This simple example shows that even though a probabilistic loss (KL-divergence in this case) does not explicitly take into account the regularities of the value function (or more generally, the decision problem), an estimate based on it (MLE in this case) might still perform as well as an estimate from an approach that explicitly takes the value into account.

In the rest of this section, we provide some more complex cases, for which we can find solutions, or at least reveal some structure of them, analytically. In Section 5, we perform some empirical studies for even more complex problems. We see that there are some situations, similar to the example we just mentioned, where MLE performs as well as VAML, but there are cases where it does not.

4.1 Aggregation-based Models for $\mathcal{M}$ and $\mathcal{F}$

In this section, we consider aggregation-based estimators for both value function and probability distribution. Consider two sets of countable partitions $\{I_i\}_i$ (to represent P) and $\{J_j\}_j$ (to represent V) of the state space $\mathcal{X}$, i.e., $\cup_i I_i = \mathcal{X}$, $I_i \cap I_{i'} = \emptyset$ for $i \neq i'$ and similarly for $\{J_j\}$. We define the space of all probability models as

$$\mathcal{M} = \left\{ x \mapsto \sum_i p_i \mathbb{I}\{x \in I_i\} \ : \ \sum_i p_i = 1, p_i \geq 0 \ \forall i \right\},$$

and the space of value functions as

$$\mathcal{F} = \left\{ x \mapsto \sum_j v_j \mathbb{I}\{x \in J_j\} \ : \ |v_j| \leq \bar{v}_j \right\}, \tag{24}$$

for some sequence of $(\bar{v}_j)$. For example if we set $\bar{v}_j = B > 0$ for all j, it defines an B-bounded function space $\mathcal{F}$. We would like to compare $c_{\text{VAML}}(P_{\text{VAML}}, P^*)$ and $c_{\text{VAML}}(P_{\text{MLE}}, P^*)$ to see when MLE performs as well as VAML and when it is worse.

We consider two separate cases. The first is when the partition of the probability model $\{I_i\}$ is finer than the partition of the value function $\{J_j\}$. This means that for any J_j, we can find a subset of $\{I_{i'}\} \subset \{I_i\}$ such that $\cup_{i'} I_{i'} = J_j$. We see that based on this definition of finer, each I_i is a subset of only one of J_j, i.e., it does not intersect with more than one J_j. The second case is the opposite. We consider when the partition of the value function $\{J_j\}_j$ is finer than the partition of the probability model $\{I_i\}_i$, with a similar definition.

4.1.1 P has a finer partition than V

We use the double-indexed $\{I_{i,j}\}_i$ to refer to the elements of $\{I_i\}$ that are a subset of J_j for a particular value of j. So $\cup_i I_{i,j} = J_j$. Because of the way we defined "finer", we have that $I_{i,j} \cap J_{j'} = \emptyset$ for $j \neq j'$. With this notation, the model space is written as $\mathcal{M} = \left\{ x \mapsto \sum_{i,j} p_{i,j} \mathbb{I}\{x \in I_{i,j}\} \ : \ \sum_{i,j} p_{i,j} = 1, p_{i,j} \geq 0 \ \forall i, j \right\}$.

For a $V \in \mathcal{F}$ and $P \in \mathcal{M}$, as defined above, we have

$$\begin{aligned}\int [P(\mathrm{d}x') - P^*(\mathrm{d}x')]\, V(x') &= \sum_j \int_{J_j} [P(\mathrm{d}x') - P^*(\mathrm{d}x')]\, V(x') \\ &= \sum_j \int_{J_j} [P(\mathrm{d}x') - P^*(\mathrm{d}x')]\, v_j \\ &= \sum_j v_j \sum_i \int_{I_{i,j}} [P(\mathrm{d}x') - P^*(\mathrm{d}x')] \\ &= \sum_j v_j \sum_i [P(I_{i,j}) - P^*(I_{i,j})]\,.\end{aligned}$$

Without loss of generality, suppose that $\bar{v}_j = 1$ for all j in (24). We have

$$\begin{aligned}\sup_{V \in \mathcal{F}} \int [P(\mathrm{d}x') - P^*(\mathrm{d}x')]\, V(x') &= \max_{(v_j)\ \text{s.t.}\ |v_j| \le 1} \sum_j v_j \sum_i [P(I_{i,j}) - P^*(I_{i,j})] \\ &= \sum_j \left| \sum_i [P(I_{i,j}) - P^*(I_{i,j})] \right|. \end{aligned} \tag{25}$$

For any $P \in \mathcal{M}$, by definition, $P(I_{i,j}) = p_{i,j}$. So by choosing $p_{i,j} = P^*(I_{i,j})$, we see that a minimizer of the loss exists within $\mathcal{M}$ and that minimizer makes the objective equal to zero. This is the VAML's solution, so $c_{\mathrm{VAML}}(P_{\mathrm{VAML}}, P^*) = 0$.

Interestingly, the MLE's solution $P_{\mathrm{MLE}} \leftarrow \mathrm{argmin}_{P \in \mathcal{M}} \mathsf{KL}(P^* || P)$ also makes the VAML's objective zero. To see this, we solve the equivalent optimization problem

$$\max_{\substack{p_{i,j} \ge 0 \\ \text{s.t.}\ \sum_{i,j} p_{i,j} = 1}} \sum_{i,j} P^*(I_{i,j}) \log p_{i,j},$$

which leads to the solution $p_{i,j} = P^*(I_{i,j})$. This solution belongs to $\mathcal{M}$, and makes (25) zero, i.e., $c_{\mathrm{VAML}}(P_{\mathrm{MLE}}, P^*) = 0$.

Note that if we changed our choice of $\bar{v}_j$ in the definition of $\mathcal{F}$ to any other bounded sequence, we would still obtain a result similar to (25) (with different weighting). In any case, the solutions of both VAML and MLE make the objective function zero, so we have $c_{\mathrm{VAML}}(P_{\mathrm{VAML}}, P^*) = c_{\mathrm{VAML}}(P_{\mathrm{MLE}}, P^*) = 0$.

This shows that if the aggregation for P is finer than that of V (with our very definition of how the aggregation should be), the choice of $\mathcal{F}$ does not affect the model approximation error of neither VAML nor MLE. This case extends the example in the beginning of Section 4 from a value function space with bounded constant functions to a more general case of aggregation estimator of V. This extends the range of function spaces $\mathcal{F}$ for which MLE performs as well as VAML. Next we study some other situation when MLE is no more performing as well.

4.1.2 V has a finer partition than P

In this case, for any I_i in the partition for P, we have a set of disjoint $\{J_{i,j}\}_j$ such that $\cup_j J_{i,j} = I_i$. Here the value function space can be written as

$$\mathcal{F} = \left\{ x \mapsto \sum_{i,j} v_{i,j} \mathbb{I}\{x \in J_{i,j}\} \;:\; |v_{i,j}| \le \bar{v}_{i,j} \right\}.$$

Similar to the previous case, we decompose $\int [P(\mathrm{d}x') - P^*(\mathrm{d}x')]\, V(x')$ to partition-dependent components. The

difference is only in the order of integration. For a $P \in \mathcal{M}$, we have

$$\begin{aligned}
\sup_{V \in \mathcal{F}} \int \left[P(\mathrm{d}x') - P^*(\mathrm{d}x')\right] V(x') &= \sup_{V \in \mathcal{F}} \sum_{i,j} \int_{J_{i,j}} \left[P(\mathrm{d}x') - P^*(\mathrm{d}x')\right] V(x') \\
&= \max_{(v_{i,j})} \sum_{i,j} \int_{J_{i,j}} \left[P(\mathrm{d}x') - P^*(\mathrm{d}x')\right] v_{i,j} \\
&= \max_{(v_{i,j})} \sum_{i,j} v_{i,j} \left[P(J_{i,j}) - P^*(J_{i,j})\right] \\
&= \sum_{i,j} \bar{v}_{i,j} \left|P(J_{i,j}) - P^*(J_{i,j})\right|,
\end{aligned}$$

with the understanding that the max over $(v_{i,j})$ satisfies the constraints in the definition of $\mathcal{F}$.

We need to compute $P(I_{i,j})$ for a $P \in \mathcal{M}$. Let λ denote the Lebesgue measure over $\mathcal{X}$ (or uniform measure over a compact $\mathcal{X}$). Note that by the definition of $\mathcal{M}$, for any (measurable) subset S of I_i, we have $P(S) = \int_S p_i \mathbb{I}\{x \in I_i\} \mathrm{d}x = p_i \lambda(S)$. In particular, $p_i = \frac{P(I_i)}{\lambda(I_i)}$. Therefore,

$$P(J_{i,j}) = p_i \lambda(J_{i,j}) = P(I_i) \frac{\lambda(J_{i,j})}{\lambda(I_i)} = p(I_i) \lambda(J_{i,j}|I_i).$$

As a result, for this choice of $\mathcal{F}$ and for any $P \in \mathcal{M}$, we get that

$$c_{\mathrm{VAML}}(P, P^*) = \sum_{i,j} \bar{v}_{i,j} \left|P(I_i)\lambda(J_{i,j}|I_i) - P^*(J_{i,j})\right|. \tag{26}$$

For the MLE, we have $P_{\mathrm{MLE}}(I_i) = P^*(I_i)$, so after some manipulations, we obtain

$$c_{\mathrm{VAML}}(P_{\mathrm{MLE}}, P^*) = \sum_i P^*(I_i) \sum_j \bar{v}_{i,j} \left|\lambda(J_{i,j}|I_i) - P^*(J_{i,j}|I_i)\right|. \tag{27}$$

As opposed to the previous case, the MLE solution is not making this objective equal to zero unless the conditional distribution $P^*(J_{i,j}|I_i)$ is uniform.

When the distribution of $P^*(J_{i,j})$ is varying between each $\{J_{i,j}\}_j$ within I_i, a single $P(I_i)$ cannot make the terms in the summation of (26) equal to zero. Both MLE and VAML provide a constant $P(I_i)$ (and as a result, $P(J_{i,j})$ for all $J_{i,j}$ within I_i), but the MLE solution ignores the weighting $\bar{v}_{i,j}$ while VAML does not. VAML can exploit the structure in the value function (here in the form of upper bounds on the value function in each partition). The exact amount that VAML can exploit depends on P^* and $(\bar{v}_{i,j})$, and is problem dependent. We numerically study this in Section 5.1.

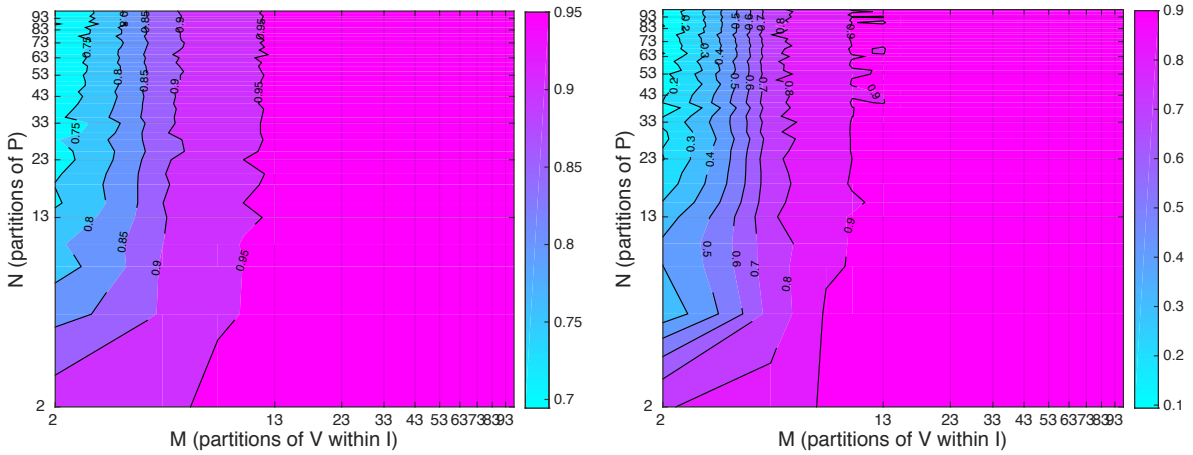
5 EMPIRICAL STUDIES

5.1 Model Approximation Error for V Having a Finer Partition than P

We numerically study the model approximation error for the case of "V having a finer partition than P", as discussed in Section 4.1.2, with different regularities of P^* and $\mathcal{F}$. With the same notation as that section, we define the problem as follows. We choose $N = 2, 3, \ldots$ equal partitions $\{I_i\}$ of $\mathcal{X}$ to represent the probability model $\mathcal{M}$. For each I_i, we equally partition it to $M = 2, 3, \ldots$ partitions $\{J_{i,j}\}_j$ to represent $\mathcal{F}$ (so $\mathcal{F}$ is represented by NM partitions).

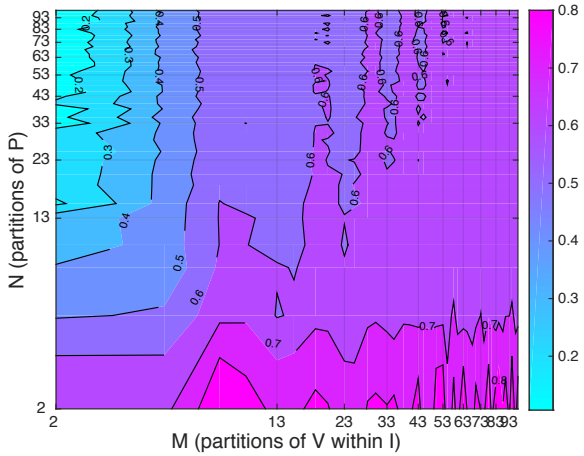
The first case is when there is "low" amount of regularity in P^* or $\mathcal{F}$. We choose $\bar{v}_{i,j} \propto j$ for all i and j. For each run, the values of $P^*(J_{i,j})$ are selected proportional to i.i.d. samples from a random variable with a log-normal distribution with log-mean of 1.0 and log-standard deviation of 0.5 (we normalize the random variables so that P^* be a probability distribution). Since all $\bar{v}_{i,j}$ are the same for the same value of i and all of $P^*(J_{i,j})$ have the same distribution, we expect that there is not much structure to be exploited by VAML. We compare the ratio

$$\frac{c_{\mathrm{VAML}}(P_{\mathrm{VAML}}, P^*)}{c_{\mathrm{VAML}}(P_{\mathrm{MLE}}, P^*)}, \tag{28}$$



(a) Low regularity

(b) Medium regularity



(c) High regularity

Figure 1: Numerical results for V has a finer partition than P for different amount of regularities in P^* and $\mathcal{F}$. This is the contour graph of the average over 30 independent runs of the ratio (28).

which is always smaller than or equal to 1. To optimize (26), we used CVX, a package for specifying and solving convex programs [Grant and Boyd, 2014].

Figure 1(a) depicts the result for the range of value $N = 2, \ldots, 100$ and $M = 2, \ldots, 100$. As expected, the ratio is always smaller than 1, and it ranges between around 0.75 to 0.95. The better performance of VAML compared to MLE is due to VAML's ability to exploit random fluctuation of $P^*(J_{i,j})$.

Next, we consider a more structured problem by changing the definition of $\mathcal{F}$. Instead of having an upper bound $\bar{v}_{i,j} \propto j$, which allows the contribution of all $J_{i,j}$ within I_i to various degrees, we assume that in each set I_i, there are two types of sets $J_{i,j}$ with significantly different amount of contribution to the value function:

$$\bar{v}_{i,j} = \begin{cases} 1 & j = 1, \ldots, \lfloor \frac{M}{2} \rfloor \\ 0 & j = \lfloor \frac{M}{2} \rfloor + 1, \ldots, M \end{cases}$$

This essentially indicates that estimating the probabilities $P^*(J_{i,j})$ within the second half of each I_i is irrelevant in the value function estimation. MLE cannot benefit from this information, while VAML can. The result of the ratio of their losses is depicted in Figure 1(b).

We observe much smaller ratios, which shows that VAML has a much smaller model approximation error compared to MLE. The ratio is smaller particularly for smaller values of M and larger values of N. The decrease in the ratio

for larger N is likely because of the extra degrees of freedom that VAML has in minimizing the loss function (26). We also see that when we have many partitions to represent V within each partition to represent P (i.e., M increases), the ratio increases too and VAML loses its edge compared to MLE.

Finally, we add some structure to $P^*(J_{i,j})$ too. We generate a set of i.i.d. random variables $p_{i,j}$ as before, drawn from the same log-normal distribution. But we sort each sequence $(p_{i,j})_{j=1,\dots,M}$ in either ascending or descending order. The choice of ascending or descending is determined by an independent unbiased coin flip. We then set $P^*(J_{i,j})$ proportional to $p_{i,j}$—with proper normalization. As a result, there is considerable structure within each I_i, but there is not much structure among different I_is. The choice of $\bar{v}_{i,j}$ is as in the previous case, so a good model estimation method is better to focus more on the first half of $J_{i,j}$ within each I_i. But depending on whether the sorting has been done ascending or descending, some partitions I_i are contributing more to the total loss than others. As before MLE is oblivious to this structure, while VAML is not. Figure 1(c) shows the result. We see that even compared to the previous case, VAML performed better.

These examples show that we can expect VAML to have smaller policy approximation error compared to MLE's whenever there is much structure in P^* and $\mathcal{F}$.

5.2 Effect of Model Learning Method in Value Function Estimation

In this section, we empirically study the performance of VAML and MLE-based estimators when they are used within a complete model-based RL algorithm.

In the previous section, we studied $c_{\text{VAML}}(\mathcal{P}_{\text{VAML}}, \mathcal{P}^*)$ and $c_{\text{VAML}}(P_{\text{MLE}}, \mathcal{P}^*)$. The motivation is that having a small $c_{\text{VAML}}(\hat{\mathcal{P}}; \mathcal{P}^*)$ leads to a small error in applying the Bellman optimality operator. The relation between the Bellman operator and the quality of outcome policy, however, is quite complex. In this section, we empirically study the performance of a VAML and MLE-based estimators when they are used within a model-based RL algorithm.

We consider a finite MDP. We choose several partition-based (i.e., aggregation) model space $\mathcal{M}$ to which the true model $\mathcal{P}^*$ does not belong. Also we consider a partition-based value function space $\mathcal{F}^{|\mathcal{A}|}$. When the resolution of the partitioning is lower than the number of states, the optimal value function Q^* might be outside $\mathcal{F}^{|\mathcal{A}|}$. We use the approximate value iteration as Planner. That is, given a model $\hat{\mathcal{P}}$, which is chosen to be either the true model $\mathcal{P}^*$ of the MDP or the estimated models $\mathcal{P}_{\text{VAML}}$ or $\mathcal{P}_{\text{MLE}}$, we repeatedly apply

$$\hat{Q}_{k+1} \leftarrow \Pi_{\mathcal{F}^{|\mathcal{A}|}}(\hat{T}^*_{\hat{\mathcal{P}}} \hat{Q}_k)$$

to obtain an approximation $\hat{Q}^*_{\mathcal{P}^*/\mathcal{P}_{\text{VAML}}/\mathcal{P}_{\text{MLE}}}$ to Q^*, the true optimal action-value function. Here $\Pi_{\mathcal{F}^{|\mathcal{A}|}}$ is the orthogonal projection onto $\mathcal{F}^{|\mathcal{A}|}$, hence the "approximate" part of AVI. We obtain Q^* (and hence V^*) using exact VI with $\mathcal{P}^*$. Note that $\hat{Q}^* = \hat{Q}^*_{\mathcal{P}^*}$ is only an approximation to Q^* because $\hat{Q}^*$ is obtained by AVI, so it is forced to be within $\mathcal{F}^{|\mathcal{A}|}$, but Q^* in general may not belong to $\mathcal{F}^{|\mathcal{A}|}$. The approximations $\hat{V}^*_{\mathcal{P}^*/\mathcal{P}_{\text{VAML}}/\mathcal{P}_{\text{MLE}}}$ are defined similarly.

After obtaining the various approximations of the optimal value function, we compute their corresponding greedy policies $\pi_{\text{MLE/VAML}} \leftarrow \text{Planner}(\mathcal{P}_{\text{MLE}}/\mathcal{P}_{\text{VAML}})$. In particular, we are interested in $V^{\pi_{\text{MLE}}}$ and $V^{\pi_{\text{VAML}}}$, the true value function of the policies obtained by the estimated models.

We use two criteria to evaluate the quality of the estimated models. The first is that how close $\hat{V}^*_{\mathcal{P}_{\text{VAML}}/\mathcal{P}_{\text{MLE}}}$ is to $\hat{V}^*_{\mathcal{P}^*}$. We use the L_2-norm of these distances, i.e., $\|\hat{V}^*_{\mathcal{P}^*} - \hat{V}^*_{\mathcal{P}_{\text{VAML}}/\mathcal{P}_{\text{MLE}}}\|_2$. By comparing to $\hat{V}^*_{\mathcal{P}^*}$, instead of V^*, we separate the error caused by the choice of model space and the model estimation procedure (MLE or VAML), which is our main object of study, from the error caused by the choice of value function space $\mathcal{F}^{|\mathcal{A}|}$. The second criterion is the performance loss of the obtained policies compared to the optimal policy, that is, $\sum_{x \in \mathcal{X}} [V^*(x) - V^{\pi_{\text{MLE/VAML}}}(x)] = \|V^* - V^{\pi_{\text{MLE/VAML}}}\|_1$, where $\pi_{\mathcal{P}_{\text{VAML}}} = \hat{\pi}(\cdot; \hat{Q}^*_{\mathcal{P}_{\text{VAML}}})$ and $\pi_{\mathcal{P}_{\text{MLE}}} = \hat{\pi}(\cdot; \hat{Q}^*_{\mathcal{P}_{\text{MLE}}})$, the greedy policies w.r.t. $\hat{Q}^*_{\mathcal{P}_{\text{VAML}}}$ and $\hat{Q}^*_{\mathcal{P}_{\text{MLE}}}$. Because of the value function and model approximation errors, the performance loss might be non-zero. But it is possible that even though $\hat{Q}_{\mathcal{P}_{\text{VAML}}/\mathcal{P}_{\text{MLE}}} \neq Q^*$, the greedy policy is still an optimal policy, as we shortly see. This is due to the action-gap phenomenon [Farahmand, 2011].

Let us define the parameters of the problem more concretely. We choose a finite state random-walk MDP with $|\mathcal{X}| = 25$, $\mathcal{A} = \{\text{left}, \text{right}\}$, and $\gamma = 0.9$. The choice of $a =$ "right" moves the agent to one of the four right-side states with equal probability (with a total probability of 0.7), does not move it (with probability of 0.2), and

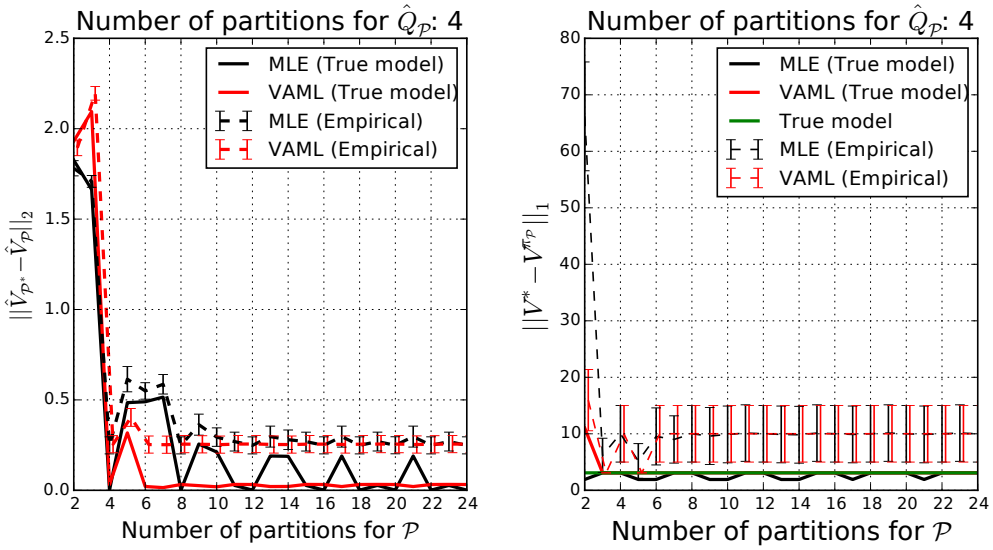


Figure 2: The effect of the number of partitions N of $\hat{\mathcal{P}}$ on MLE and VAML when $\mathcal{F}^{|\mathcal{A}|}$ has $M = 4$ partitions. The left figure shows the value function approximation error. The right figure shows the performance loss of using the obtained greedy policies. The dashed curves correspond to the empirical model. The error bars depict one standard error with the number of independent runs equal 20.

moves it to the left-side state (with probability of 0.1). The opposite holds for $a =$ "left". The boundaries are not connected, and the behaviour changes accordingly. The value function space $\mathcal{F}$ is defined based on partitioning of the state space $\mathcal{X}$ to M subsets. So we have $\mathcal{F} = \left\{ x \mapsto \sum_{j=1}^{M} v_j \mathbb{I}\{x \in J_j\} : v \in \mathbb{R}^M \right\}$. The action-value function space $\mathcal{F}^{|\mathcal{A}|}$ is simply $|\mathcal{A}| = 2$ copies of $\mathcal{F}$. We change M in our experiments.

The model space, used by both VAML and MLE, is an exponential family defined based on N partitions, cf. (13). The features ϕ' are defined so that each of them is an indicator function of whether given a state-action pair (x, a), the next-state x' would be in one of the N partitions or not. More precisely,

$$\mathcal{M} = \Bigg\{ \hat{\mathcal{P}}_w(x'|x,a) = \frac{\exp\left(\phi'^{\top}(x'|x,a)w\right)}{\sum_{x''} \exp\left(\phi'^{\top}(x''|x,a)w\right)} : \\ \phi'_{i,k,l}(x'|x,a) = \mathbb{I}\{x' \in I_i, x = k, a = l\}, i = 1, \dots, N, k = 1, \dots, |\mathcal{X}|, l = 1, \dots, |\mathcal{A}|, w \in \mathbb{R}^{N|\mathcal{X}||\mathcal{A}|} \Bigg\}.$$

Note that the partitioning is only on the next-state x', and not the current state x. We change N in our experiments. A small detail is that because the state space is finite and partitions $\{I_i\}$ and $\{J_j\}$ have integer-valued lengths, the lengths of all of them are not exactly $|\mathcal{X}|/M$ (or $|\mathcal{X}|/N$). A state x belongs to J_j with $j = \lfloor \frac{x}{M} \rfloor$, and similarly for I_is.

We use gradient descent to optimize the loss functions for both VAML and MLE. In particular, we use ADAM by Kingma and Ba [2015] with the choice of $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\alpha = 0.25$, and $\varepsilon = 10^{-8}$ in their notations. We followed ADAM for 1000 iterations.

Figure 2 and Figure 3 present the results of the experiments for $M = 4$ and $M = 11$, respectively. The bold curves in each figure show the results when $\mathcal{P}^*$ is given as the input to VAML or MLE (so $c^2_{2,\nu}(\hat{\mathcal{P}}, \mathcal{P}^*)$ of (7) is minimized instead of $c^2_{2,\nu_n}(\hat{\mathcal{P}}, \mathcal{P}_n)$ of (9); and similarly for MLE), while the dashed curves are for when samples (i.e., empirical measure $\mathcal{P}_n$) are used for minimization. For the empirical measure, for each choice of a, we draw 100 states X_i uniformly from $\mathcal{X}$, and then draw 25 samples from the next state according to $\mathcal{P}(\cdot|X_i, a)$. So in total, we have $2 \times 100 \times 25 = 5000$ samples. Studying the behaviour of the algorithms under both the true distribution and the empirical distribution allows us to separate the errors due to model approximation error and the estimation error.

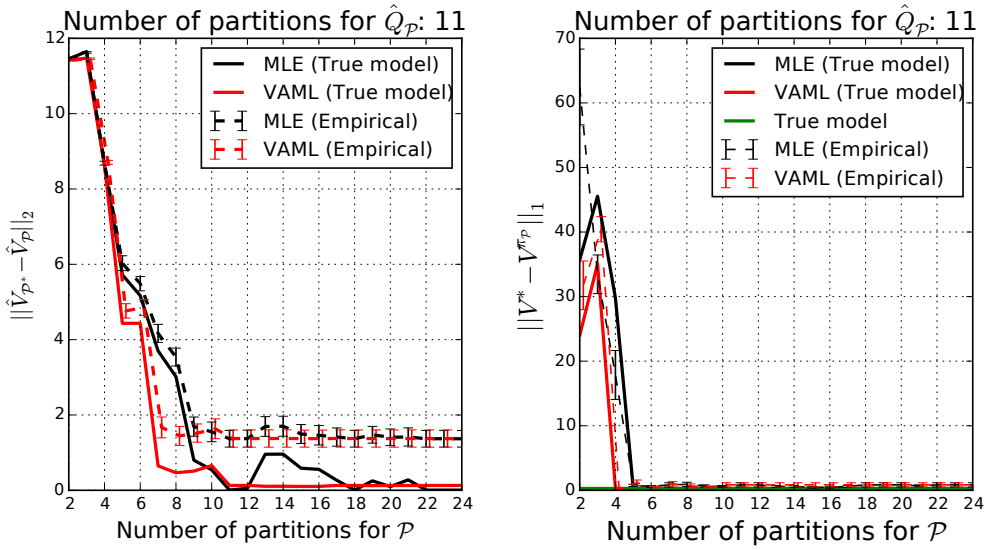


Figure 3: The effect of the number of partitions N of $\hat{\mathcal{P}}$ on MLE and VAML when $\mathcal{F}^{|\mathcal{A}|}$ has $M = 11$ partitions. The left figure shows the value function approximation error. The right figure shows the performance loss of using the obtained greedy policies. The dashed curves correspond to the empirical model. The error bars depict one standard error with the number of independent runs equal 20.

The left-side figures show the value function estimation error $\|\hat{V}^*_{\mathcal{P}^*} - \hat{V}^*_{\mathcal{P}_{\text{VAML}}/\mathcal{P}_{\text{MLE}}}\|_2$. We observe that as the number of partitions N for the model space increases, the error decreases too. The errors for the VAML model for most Ns are smaller than MLE's, sometimes significantly.

A curious observation is that when N is an integer multiply of M, the number of partitions in the representation of $\mathcal{F}$, the error of MLE becomes very small—sometimes even slightly smaller than VAML's (when $N = 4, 8, 12, \ldots$ in Figure 2, and $N = 11, 22$ in Figure 3) . This is aligned with our analysis in Section 4.1.1, when P has a finer partition than V does. We observe a similar smallness of errors in the neighbourhood of those integer multiplies because the structure of the partition would be similar to the aforementioned case. When empirical data is used, the difference between VAML and MLE become less significant since the estimation error dominates the model approximation error. We also note that $\|\hat{V}^*_{\mathcal{P}^*} - V^*\|_2$ is 9.19 for $M = 4$ and 1.40 for $M = 11$.

The right-side figures show the performance loss $\|V^* - V^{\pi_{\text{MLE/VAML}}}\|_1$ for both cases. If the performance loss of a policy is zero, it means that it is optimal. As a baseline, the green line shows the performance loss of the greedy policy w.r.t. $\hat{Q}^*_{\mathcal{P}^*}$. For $M = 4$, the performance loss is 3.09 and for $M = 11$, it is 0.277. It is quite possible that even though there is a significant value function error, due the action-gap phenomenon, the greedy policy is behaving close to optimal or the best value function in the class. That is why we observe that in case of $M = 11$, the performance of $\pi_{\mathcal{P}_{\text{VAML}}}$ and $\pi_{\mathcal{P}_{\text{MLE}}}$ is as good as good as $\hat{\pi}(\cdot; \hat{Q}^*_{\mathcal{P}^*})$ after $N = 4$ for VAML and $N = 5$ for MLE. For $M = 4$, we observe a similar behaviour, but at slightly earlier N. It is curious to note that the performance loss of MLE is sometimes slightly better than that of $\hat{\pi}(\cdot; \hat{Q}^*_{\mathcal{P}^*})$, particularly at those values of N when $\|\hat{V}^*_{\mathcal{P}^*} - \hat{V}^*_{\mathcal{P}_{\text{MLE}}}\|_2$ is larger. This basically means that wronger models happened to make better policies. We do not believe this is a general pattern beyond this particular problem, but further investigation might be interesting.

5.3 Model Learning in a Continuous State Space

We empirically compare the quality of a model learned by VAML (9) with the model learned by MLE in a continuous state space problem and study the effect of having model approximation error. We also study the effect of the number of samples used for estimation of the expectations in the gradient evaluation of VAML (cf. Theorem 1).

We consider $\mathcal{X} = [0, 1]^d$ with $d = 10$, and we ignore the actions. We consider two possible cases for the true model

$\mathcal{P}^*$: It is either a Gaussian distribution or an exponential distribution. In both cases the mean of distributions is specified by a function of the current state. The model class $\mathcal{M}$, however, only consists of Gaussian distributions. So whenever the true model is exponential, we have model approximation error.

To be concrete, let $\phi'(x) = \text{diag}(\cos(x_1), \dots, \cos(x_d)) \in \mathbb{R}^{d\times d}$ and $w^* \in \mathbb{R}^d$. For the Gaussian model case, $\mathcal{P}^*(\cdot|x)$ generates the next states according to $X' \sim \mathcal{N}(\phi'(x)w^*, \sigma^2\mathbf{I}_{d\times d})$. For the exponential case, $\beta = \phi'(x)w^* \in \mathbb{R}^d$ would be the scale parameter, so the probability density function for the jth dimension is $\mathcal{P}^*(\mathrm{d}x'_j|x) = \frac{1}{\beta_j}\exp\left(-\frac{x'_j}{\beta_j}\right)$. We choose the model class $\mathcal{M}$ to be the same as the Gaussian case, that is $\mathcal{M} = \left\{\hat{\mathcal{P}} : x \mapsto \mathcal{N}(\phi'(x)w, \sigma^2\mathbf{I}_{d\times d}) : w \in \mathbb{R}^d\right\}$. The features $\phi(x) = [x_1, x_1^2, x_2, x_2^2, \dots, x_d, x_d^2] \in \mathbb{R}^{2d}$ of $\mathcal{F}$ are selected to be "incompatible" with ϕ'.

Each experiment consists of randomly choosing a model, specified by w^*, generating data from it, and then comparing the behaviour of MLE and VAML. The vector w^* is drawn uniformly randomly in the range of $[0, 2]^d$. We generate n data points in the form of $\mathcal{D}_n = \{(X_i, X'_i)\}_{i=1}^n$. The distribution of X_i is uniform on $\mathcal{X}$ and the samples are selected independently, and $X'_i \sim \mathcal{P}^*(\cdot|X_i; w^*)$. The value of σ is 0.5 in our experiments, and we use 50 independent runs.

The task of model learning would be to find a $\hat{w}$ such that $\mathcal{N}(\phi'(x)\hat{w}, \sigma^2\mathbf{I}_{d\times d})$ is as close as possible to the true model. The MLE minimizes the empirical KL distance, i.e., the conditional version of (1) and VAML uses (9). We use gradient descent to optimize the loss functions for both cases, in particular using ADAM by Kingma and Ba [2015] to update the weights.[9]

To compute the gradients of an exponential family, either for MLE or VAML, we require to compute certain expectations as is evident in the statement of Theorem 1 and (14). Nonetheless, for the Gaussian model, the involved expectations can be computed in closed-form for MLE. But for VAML, we still require the numerical computation of the expectations. Fortunately sampling from a Gaussian distribution is easy, so we simply obtain m independent samples from the current estimated model to compute the expectations (and covariance matrix, using independent samples). In this experiment, we only use a small number of samples ($m = 5$) for each expectation evaluation. The result is a noisy, but unbiased, estimate of the gradient.

Figure 4 depicts the results for both cases when the true model belongs to $\mathcal{M}$ (Gaussian case) and when it does not (exponential case). The figure shows the evolution of two types of error functions as a function of the number of samples n used in the training. The first error function is $\|\hat{w} - w^*\|_2$, and the second one is $c_{2,\nu}(\hat{\mathcal{P}}, \mathcal{P}^*) = \|(\hat{\mathcal{P}} - \mathcal{P}^*)\phi\|_{2,\nu}$.

When there is no model approximation error, both errors of MLE and VAML gradually go to zero. They also perform comparably the same. The reason that MLE performs well is that when $\mathcal{P}^* \in \mathcal{M}$, the solution $\hat{w}$ by MLE converges to w^*. In that case, $\hat{\mathcal{P}} \to \mathcal{P}^*$ (e.g., in KL sense), so as discussed around (4) and in Section 4, the associated V-weighted cost goes to zero too.

When there is mismatch between the true model (exponential) and the models in our class $\mathcal{M}$, we have model approximation error. In this case, we observe that even though MLE has a smaller ℓ_2-error in the weight space, its $c_{2,\nu}(\hat{\mathcal{P}}, \mathcal{P}^*)$ is significantly larger, and saturates at a much larger model approximation error. Here it pays off to try to directly minimize the cost function that really matters, and not the one that minimizes a probabilistic notion of distance, such as the KL divergence.

We also study the effect of number of samples used in the estimation of expectations in the gradient computations (cf. Theorem 1). Here we set the number of training samples to $n = 250$, but change the number of Monte Carlo samples m used to estimate the expectations in computation of gradient.

Figure 5 shows the result for when there is no model approximation error, and Figure 6 shows it when there is. We observe that increasing m leads to better solutions, but even small m is sufficient to provide reasonably good solutions. The effect of smaller values of m is more prominent in the case that we do not have model approximation error (Figure 5). In that case, the error due to the Monte Carlo estimate becomes comparable to the already small absolute error.

[9] We choose $\beta_1 = 0.8$, $\beta_2 = 0.99$, $\alpha = 0.03$, and $\varepsilon = 10^{-8}$ according to the notation of Kingma and Ba [2015]. We slightly modified the procedure by gradually decreasing α through iterations with a geometric rate of 0.995 per iteration. The experiments are done with 500 iterations.

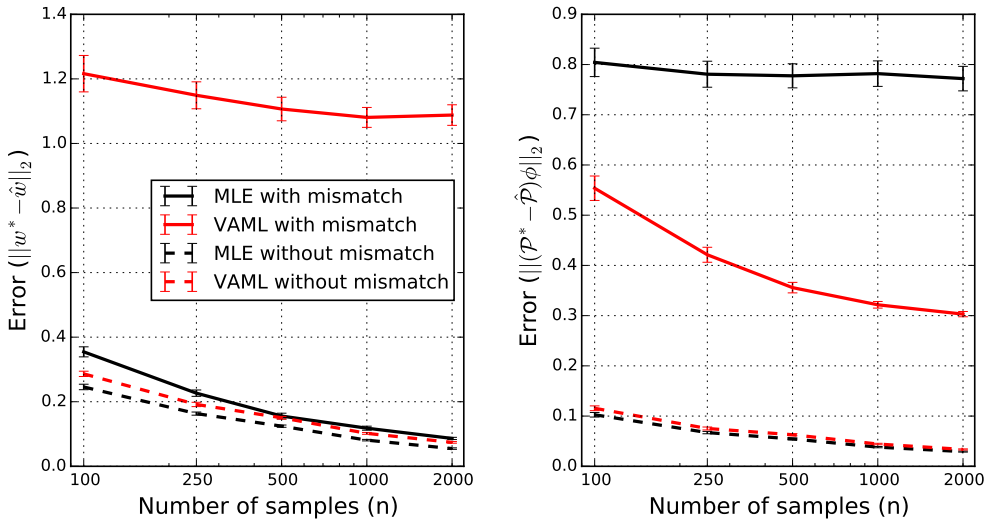


Figure 4: Effect of the number of samples n on MLE and VAML. The left figure shows the ℓ_2-error of the estimated parameters and the right figure shows the error in the value approximation within the value function space $\mathcal{F}$ as a function of the number of samples n used for training. The red curves show the behaviour of VAML while the black ones show the behaviour of MLE. The dashed curves corresponds to the case when there is no model approximation error while the bold curves are for the case of model approximation error. The error bars depict one standard error with the number of independent runs equal 50.

6 DISCUSSION AND FUTURE WORK

We presented a loss function to learn the probability transition model to be used by a model-based reinforcement learning algorithm. In contrast with the conventional approaches, we take some aspects of the decision problem, particularly the knowledge about the value function approximator, into account.

There are several methods for learning the transition probability kernel or quantities related to it. For example, the method of Ormoneit and Sen [2002] approximates $\mathcal{P}$ by a particular finite approximation that is obtained by smoothing kernel. The method then uses the estimated finite MDP to find the approximate value function. Similarly, the method of Barreto et al. [2011] finds a smaller finite approximation of the transition probability kernel than the work of Ormoneit and Sen [2002] using the stochastic factorization trick. These methods effectively learn a transition model that does not benefit from the structure of the value function beyond the required condition that the value function should be Lipschitz continuous (cf. Lemma 2 of Ormoneit and Sen [2002]). This is in contrast with the method of this work that explicitly takes the value function space $\mathcal{F}$ into account.

In a different line of work, some methods estimate auxiliary operators that are different from $\mathcal{P}$, but can be used to compute the effect of Bellman operator on a value function. One such example is the method by Grünewälder et al. [2012], which directly estimates the conditional mean embedding operator, that is, the mapping $V \mapsto \int \mathcal{P}(\mathrm{d}x'|x,a)V(x')$ for all V in an RKHS. Lever et al. [2016] suggest a method to improve the computational cost of Grünewälder et al. [2012]. Yao et al. [2014] introduce the concept of pseudo-MDP, which relaxes the constraint that $\mathcal{P}$ should be a probability kernel. Their work, however, is different from the method suggested in this section as here the estimated $\hat{\mathcal{P}}$ is by construction a probability kernel, as opposed to the outcome of their estimator, which is not. Comparing VAML, which learns a generative model, and these other approaches is an interesting research problem.

We empirically studied the behaviour of VAML and MLE when they are used within a model-based RL algorithm. The results showed that minimizing the loss suggested by VAML translates into having a better value function approximation error, as well as smaller performance loss when the learned model is used for planning. We also studied model approximation properties of VAML vs. MLE through some examples.

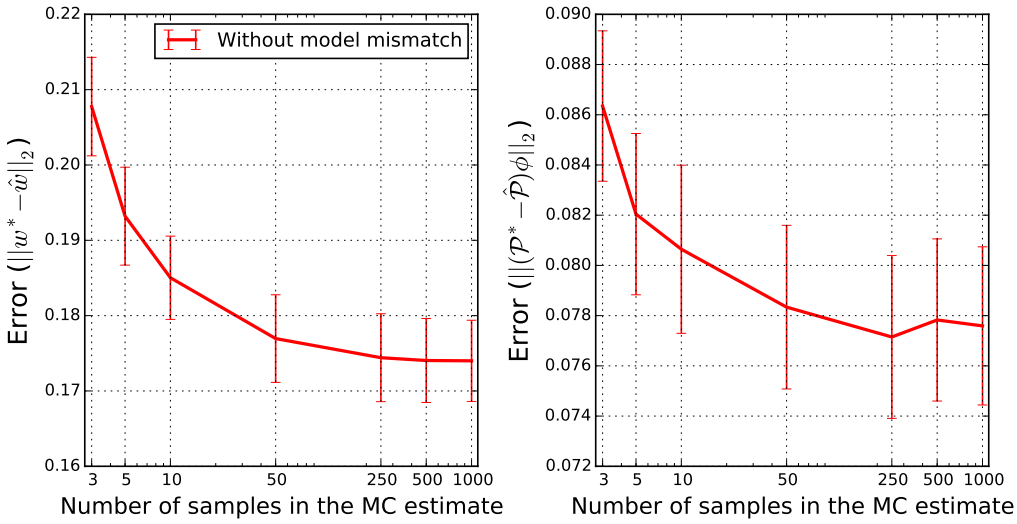


Figure 5: Effect of the number of samples m used for the Monte Carlo estimation of expectations in VAML. The left figure shows the ℓ_2-error of the estimated parameters and the right figure shows the error in the value approximation within the value function space $\mathcal{F}$ as a function of the number of Monte Carlo samples m used for estimating each expectation in the gradient. Both figures are for when there **is no** model approximation error. The error bars depict one standard error with the number of independent runs equal 50.

We would like to mention that exponential family is not the only class of probability distributions for modeling of the environment. Another possibility, which deserves further study, is the adoption of the generative adversarial network to VAML's loss function [Goodfellow et al., 2014]. Finally, incorporating the structure of policy space into model learning is another interesting research topic along the research program of this work.

A ALTERNATIVE FORMULATION: OUTSIDE SUPREMUM

We briefly mentioned in Section 2 that we might define the cost function as

$$c_{2,\nu}^2(\hat{\mathcal{P}}, \mathcal{P}^*) = \sup_{V \in \mathcal{F}} c_\nu(\hat{\mathcal{P}}, \mathcal{P}^*; V) = \sup_{V \in \mathcal{F}} \int \mathrm{d}\nu(x,a) \left| \int \left[\mathcal{P}^*(\mathrm{d}x'|x,a) - \hat{\mathcal{P}}(\mathrm{d}x'|x,a) \right] V(x') \right|^2,$$

which has the supremum over V outside the integral over state-actions, instead of within the integral of (7). Having the supremum outside the integral amounts to selecting only a *single* value function from $\mathcal{F}$, whereas having the supremum inside means that for each choice of state-action (x,a), we allow the value function evaluated over the next-state distribution to be different. The "inside" supremum formulation is more conservative than the "outside" formulation, but it is still tighter than upper bounds such as (4), which completely ignore the structure of the value function space $\mathcal{F}$. In this section, we derive how this alternative formulation can be written as an optimization problem.

Consider the function space $\mathcal{F} = \mathcal{F}_1 = \left\{ V_\theta(x) = \phi^\top(x)\theta \,:\, \theta \in \mathbb{R}^p, \|\theta\|_2 \leq 1 \right\}$ with $\phi : \mathcal{X} \to \mathbb{R}^p$, as before. Define

$$\Delta_\phi(x,a;\hat{\mathcal{P}}) = \int \Big(\mathcal{P}^*(\mathrm{d}x'|x,a) - \hat{\mathcal{P}}(\mathrm{d}x'|x,a) \Big) \phi(x').$$

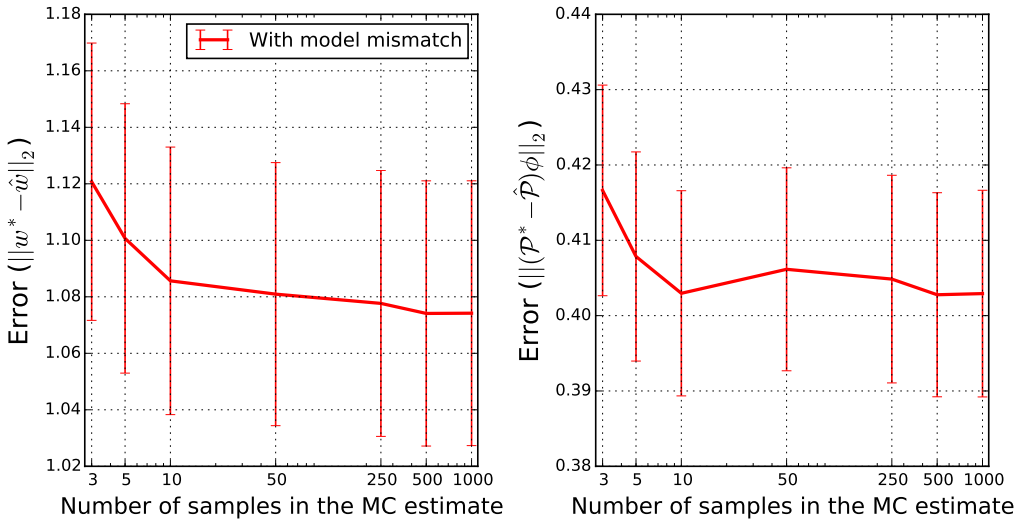


Figure 6: Effect of the number of samples m used for the Monte Carlo estimation of expectations in VAML. The left figure shows the ℓ_2-error of the estimated parameters and the right figure shows the error in the value approximation within the value function space $\mathcal{F}$ as a function of the number of Monte Carlo samples m used for estimating each expectation in the gradient. Both figures are for when there **is** model approximation error. The error bars depict one standard error with the number of independent runs equal 50.

We then have

$$\sup_{V \in \mathcal{F}} \int \mathrm{d}\nu(x,a) \left| \int \left[\mathcal{P}^*(\mathrm{d}x'|x,a) - \hat{\mathcal{P}}(\mathrm{d}x'|x,a) \right] V(x') \right|^2 =$$
$$\sup_{\|\theta\|_2 \le 1} \int \mathrm{d}\nu(x,a) \left| \Delta_\phi^\top(x,a;\hat{\mathcal{P}})\theta \right|^2 =$$
$$\sup_{\|\theta\|_2 \le 1} \theta^\top \underbrace{\left[\int \mathrm{d}\nu(x,a) \Delta_\phi^\top(x,a;\hat{\mathcal{P}}) \Delta_\phi(x,a;\hat{\mathcal{P}}) \right]}_{\triangleq \Lambda(\hat{\mathcal{P}})} \theta$$

Since $\Lambda(\hat{\mathcal{P}})$ is symmetric and positive semidefinite, it can be decomposed as $\Lambda(\hat{\mathcal{P}}) = L^\top(\hat{\mathcal{P}}) L(\hat{\mathcal{P}})$. So we have

$$\sup_{\|\theta\|_2 \le 1} \left\langle \Lambda(\hat{\mathcal{P}})\theta \,,\, \theta \right\rangle = \sup_{\|\theta\|_2 \le 1} \left\langle L(\hat{\mathcal{P}})\theta \,,\, L(\hat{\mathcal{P}})\theta \right\rangle$$
$$= \left\| L(\hat{\mathcal{P}}) \right\|_2^2 = \sigma_{\max}^2 \left(L(\hat{\mathcal{P}}) \right) = \lambda_{\max} \left(\Lambda(\hat{\mathcal{P}}) \right).$$

Therefore, the optimization problem for the population version of this formulation of VAML becomes

$$\hat{\mathcal{P}} \leftarrow \operatorname*{argmin}_{\mathcal{P} \in \mathcal{M}} \lambda_{\max} \left(\Lambda(\mathcal{P}) \right).$$

Deriving an algorithm to solve this optimization problem as well as further investigation of this formulation is postponed to a future work.

B COVERING NUMBERS FOR EXPONENTIAL FAMILY

Theorem 2 does not make any assumption on whether the model space is defined by an exponential family or not. Therefore, the capacity condition of Assumption A1 is stated in the form of an upper bound on the metric entropy

(or covering number) of the model space $\mathcal{M}$, without any reference to how the probability model is defined. Here we provide some covering number results for exponential family.

For simplicity of analysis, we assume that $\mathcal{X}$ is countable. Let us consider a space of functions $\mathcal{G}$ in the form of $g(x'|x,a) : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$. Define a corresponding exponential family

$$\hat{\mathcal{P}}_g(x'|x,a) = \frac{\exp\left(g(x'|x,a)\right)}{\sum_{x''} \exp\left(g(x''|x,a)\right)},$$

and the model space

$$\mathcal{M}_{\mathcal{G}} = \{\hat{\mathcal{P}}_g : g \in \mathcal{G}\}.$$

For the particular choice of $g(x'|x,a) = g_w(x'|x,a) = \phi'^{\top}(x'|x,a)w$ with $\phi'(x'|x,a) \in \mathbb{R}^{p'}$, parameterized by $w \in \mathbb{R}^{p'}$, we retrieve the model considered in Section 2.2. Define $\mathcal{M}_{\mathcal{W},B} = \left\{ \hat{\mathcal{P}}_w : \|w\|_2 \le B \right\}$ for some non-negative finite B.

We denote $\|g(\cdot|z)\|_2^2 = \sum_{x' \in \mathcal{X}} g^2(x'|z)$. For a sequence $z_{1:n} \triangleq z_1, \ldots, z_n \subset \mathcal{X} \times \mathcal{A}$, we define the empirical norm $L_2(P_{z_{1:n}})$ of $g \in \mathcal{G}$ by $\|g\|_{2,n}^2 = \frac{1}{n}\sum_{i=1}^n \|g(\cdot|z_i)\|_2^2 = \frac{1}{n}\sum_{i=1}^n \sum_{x' \in \mathcal{X}} g^2(x'|z_i)$. The ε-covering number of $\mathcal{G}$ w.r.t. this empirical norm is denoted by $\mathcal{N}(\varepsilon, \mathcal{G}, L_2(P_{z_{1:n}}))$. The covering number $\mathcal{N}(\varepsilon, \mathcal{M}_{\mathcal{G}}, L_2(P_{z_{1:n}}))$ is defined similarly. The following proposition is the main result of this section.

Proposition 6. *Consider the function space $\mathcal{G}$ and its corresponding model space $\mathcal{M}_{\mathcal{G}}$. For any sequence $z_1, \ldots z_n \subset \mathcal{X} \times \mathcal{A}$, we have*

$$\mathcal{N}(\varepsilon, \mathcal{M}_{\mathcal{G}}, L_2(P_{z_{1:n}})) \le \mathcal{N}(2\varepsilon, \mathcal{G}, L_2(P_{z_{1:n}})).$$

Furthermore, assume that $\sup_z \|\phi(\cdot|z)\|_2 \le C$. We have

$$\mathcal{N}(\varepsilon, \mathcal{M}_{\mathcal{W},B}, L_2(P_{z_{1:n}})) \le \left(\frac{2BC + \varepsilon}{\varepsilon}\right)^{p'}.$$

To prove this result, we use a lemma from Huang et al. [2015] (the extended version that has the proofs). The statement in that paper looks slightly different, as it is about π and Q, but that result is essentially an upper bound on the changes in probabilities in the exponential family as a function of the changes in the exponent, which matches our need here.

In the next lemma, the ℓ_2-norm is defined on $\mathcal{X}$: For a function $u : \mathcal{X} \to \mathbb{R}$ (or a vector on $\mathcal{X}$), $\|u\|_2^2 = \sum_{x \in \mathcal{X}} u^2(x)$.

Lemma 7 (Lemma 4 of Huang et al. [2015]—Extended Version)**.** *For a function $u : \mathcal{X} \to \mathbb{R}$, define*

$$\mathcal{P}_u(x) = \frac{\exp\left(u(x)\right)}{\sum_{x'} \exp\left(u(x')\right)}.$$

It holds that

$$\|\mathcal{P}_{u_1} - \mathcal{P}_{u_2}\|_2 \le \frac{1}{2}\|u_1 - u_2\|_2.$$

We are now ready to prove Proposition 6.

Proof of Proposition 6. Consider $g_1, g_2 \in \mathcal{G}$ and their corresponding $\hat{\mathcal{P}}_{g_1}, \hat{\mathcal{P}}_{g_2} \in \mathcal{M}_{\mathcal{G}}$. For any $z = (x,a)$, Lemma 7 with the choice of $u_1 = g_1(\cdot|z)$ (and similarly for u_2) shows that

$$\left\|\hat{\mathcal{P}}_{g_1}(\cdot|z) - \hat{\mathcal{P}}_{g_2}(\cdot|z)\right\|_2 \le \frac{1}{2}\|g_1(\cdot|z) - g_2(\cdot|z)\|_2.$$

Let us consider a sequence $z_1, \ldots, z_n \in \mathcal{X} \times \mathcal{A}$. We have

$$\frac{1}{n}\sum_{i=1}^n \left\|\hat{\mathcal{P}}_{g_1}(\cdot|z_i) - \hat{\mathcal{P}}_{g_2}(\cdot|z_i)\right\|_2^2 \le \frac{1}{4n}\sum_{i=1}^n \|g_1(\cdot|z_i) - g_2(\cdot|z_i)\|_2^2.$$

Therefore, a u-cover of $\mathcal{G}$ w.r.t. $L_2(\mathcal{P}_{z_{1:n}})$ defines a $\frac{u}{2}$-cover of $\mathcal{M}_{\mathcal{G}}$, as desired.

For the second part, notice that when $g_w(x'|z) = \phi'^{\top}(x'|z)w$,

$$\begin{aligned}\|g_{w_1}(\cdot|z) - g_{w_2}(\cdot|z)\|_2^2 &= \sum_{x'} \left|\phi'^{\top}(x'|z)(w_1 - w_2)\right|^2 \\ &\le \sum_{x'} \|\phi'(x'|z)\|_2^2 \, \|w_1 - w_2\|_2^2 \\ &= \|w_1 - w_2\|_2^2 \sum_{x'} \|\phi'(x'|z)\|_2^2 .\end{aligned}$$

With the same covering argument as before, we get that

$$\mathcal{N}\left(u, \mathcal{M}_{\mathcal{W},B}, L_2(P_{z_{1:n}})\right) \le \mathcal{N}\left(\frac{2u}{C}, \{\, w \,:\, \|w\|_2 \le B \,\}, \ell_2(\mathbb{R}^p)\right).$$

Finally we evoke Lemma 10, the covering number of a bounded ball in $\mathbb{R}^{p'}$, to obtain the result. □

C AUXILIARY RESULTS

C.1 Rademacher Complexity

We define Rademacher complexity and quote a result from Bartlett and Mendelson [2002]. For more information about Rademacher complexity, we refer the reader to Bartlett et al. [2005]; Bartlett and Mendelson [2002].

Let $\sigma_1, \ldots, \sigma_n$ be independent random variables with $\mathbb{P}\{\sigma_i = 1\} = \mathbb{P}\{\sigma_i = -1\} = 1/2$. For a function space $\mathcal{F} : \mathcal{X} \to \mathbb{R}$, define $R_n\mathcal{F} = \sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(X_i)$ with $X_i \sim \nu$. The Rademacher complexity (or average) of $\mathcal{F}$ is $\mathbb{E}[R_n\mathcal{G}]$, in which the expectation is w.r.t. both σ and X_i. Rademacher complexity appears in the analysis of the supremum of an empirical process right after we apply symmetrization . This makes it a notion of complexity closely related to the behaviour of the empirical process. One may interpret it as a complexity measure that quantifies the extent that a function from $\mathcal{F}$ can fit a noise sequence of length n [Bartlett and Mendelson, 2002].

The following result is a simplified (and slightly reworded) version of Theorem 2.1 of Bartlett et al. [2005].

Lemma 8. *Let $\mathcal{F} : \mathcal{X} \to \mathbb{R}$ be a measurable function space with B-bounded functions. Let $X_1, \ldots, X_n \in \mathcal{X}$ be independent random variables. Assume that for some $r > 0$, $\mathrm{Var}[f(X_i)] \le r$ for every $f \in \mathcal{F}$. Then for every $\delta > 0$, with probability at least $1 - \delta$,*

$$\sup_{f\in\mathcal{F}} \left| \mathbb{E}[f(X)] - \frac{1}{n}\sum_{i=1}^n f(X_i) \right| \le$$
$$\inf_{\alpha>0} \left\{ 2(1+\alpha)\mathbb{E}[R_n(\mathcal{F})] + \sqrt{\frac{2r\ln(2/\delta)}{n}} + 2B\left(\frac{1}{3} + \frac{1}{\alpha}\right)\frac{\log(2/\delta)}{n} \right\}.$$

C.2 Supremum of the Weighted Sums

We quote Theorem 19.1 of Györfi et al. [2002] for ease of reference. This result, in a more general form, appears as Lemma 3.2 of van de Geer [2000].

Lemma 9. *Let $L > 0$ and $W_1, \ldots, W_n$ be independent random variables with expectation zero and values in $[-L, L]$. Let $z_1, \ldots, z_n \in \mathbb{R}^d$, let $R > 0$, and let $\mathcal{F}$ be a class of functions $f : \mathbb{R}^d \to \mathbb{R}$ with the property*

$$\frac{1}{n}\sum_{i=1}^n |f(z_i)|^2 \le R^2,$$

for all $f \in \mathcal{F}$. Then

$$\sqrt{n}\delta \ge 48\sqrt{2}L \int_{\frac{\delta}{8L}}^{\frac{R}{2}} \sqrt{\log \mathcal{N}(u, \mathcal{F}, L_2(P_{z_{1:n}})}\mathrm{d}u$$

and

$$\sqrt{n}\delta \geq 36RL$$

imply

$$\mathbb{P}\left\{\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(z_i)W_i\right| > \delta\right\} \leq 5\exp\left(-\frac{n\delta^2}{2304L^2R^2}\right).$$

C.3 Covering Number for an Euclidean Ball

The following lemma, quoted from van de Geer [2000], upper bounds the covering number of a ball with radius B in $\mathbb{R}^p$.

Lemma 10 (Covering number of a ball in an Euclidean space – Lemma 2.5 of van de Geer 2000)**.** *A ball in $\mathbb{R}^p$ with radius B w.r.t. Euclidean metric (i.e., $\{\, w \in \mathbb{R}^p : \|w\|_2 \leq B \,\}$) can be covered by $\left(\frac{4B+\varepsilon}{\varepsilon}\right)^p$ balls with radius ε.*

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback.

References

Bernardo Ávila Pires and Csaba Szepesvári. Policy error bounds for model-based reinforcement learning with factored linear models. In *Conference on Learning Theory (COLT)*, 2016. 9

André M.S. Barreto, Doina Precup, and Joelle Pineau. Reinforcement learning using kernel-based stochastic factorization. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS - 24)*, pages 720–728. 2011. 23

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research (JMLR)*, 3:463–482, 2002. 27

Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Local Rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005. 9, 13, 27

Wendelin Böhmer, Steffen Grünewälder, Yun Shen, Marek Musial, and Klaus Obermayer. Construction of approximation spaces for reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 14:2067–2118, 2013. 7

Miguel A Carreira-Perpinan and Geoffrey Hinton. On contrastive divergence learning. In *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, volume 10, pages 33–40, 2005. 8

Marc Peter Deisenroth, Gerhard Neumann, and Jan Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013. 2

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015. 2

Amir-massoud Farahmand. Action-gap phenomenon in reinforcement learning. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS - 24)*, pages 172–180. Curran Associates, Inc., 2011. 19

Amir-massoud Farahmand and Doina Precup. Value pursuit iteration. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS - 25)*, pages 1349–1357. Curran Associates, Inc., 2012. 7

Amir-massoud Farahmand, Azad Shademan, Martin Jägersand, and Csaba Szepesvári. Model-based and model-free reinforcement learning for visual servoing. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2917–2924, May 2009. 2

Amir-massoud Farahmand, Rémi Munos, and Csaba Szepesvári. Error propagation for approximate policy and value iteration. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS - 23)*, pages 568–576. 2010. 5

Amir-massoud Farahmand, André M.S. Barreto, and Daniel N. Nikovski. Value-aware loss function for model learning in reinforcement learning. In *13th European Workshop on Reinforcement Learning (EWRL)*, December 2016. 2

Alborz Geramifard, Finale Doshi, Joshua Redding, Nicholas Roy, and Jonathan How. Online discovery of feature dependencies. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 881–888, New York, NY, USA, June 2011. ACM. 7

Evarist Giné and Richard Nickl. *Mathematical Foundations of Infinite-Dimensional Statistical Models*. Cambridge University Press, 2015. 8, 13

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS - 27)*, pages 2672–2680. 2014. 24

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. 8

Michael C. Grant and Stephen P. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, 2014. 18

Steffen Grünewälder, Guy Lever, Luca Baldassarre, Arthur Gretton, and Massimiliano Pontil. Modelling transition dynamics in MDPs with RKHS embeddings. In *International Conference on Machine Learning (ICML)*, pages 535–542. ACM, 2012. 23

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Verlag, New York, 2002. 8, 11, 27

Todd Hester and Peter Stone. TEXPLORE: Real-time sample-efficient reinforcement learning for robots. *Machine Learning*, 90(3), 2013. 2, 3

De-An Huang, Amir-massoud Farahmand, Kris M Kitani, and J. Andrew Bagnell. Approximate MaxEnt inverse optimal control and its application for mental simulation of human interactions. In *AAAI Conference on Artificial Intelligence*, January 2015. 26

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 20, 22

Guy Lever, John Shawe-Taylor, Ronnie Stafford, and Csaba Szepesvári. Compressed conditional mean embeddings for model-based reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2016. 23

Yitao Liang, Marlos C. Machado, E. Talvitie, and Michael Bowling. State of the art control of atari games using shallow reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 485–493, 2016. 7

David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. 8

Sridhar Mahadevan and Bo Liu. Basis construction from power series expansions of value functions. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS - 23)*, pages 1540–1548. 2010. 7

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research (JMLR)*, 8:2169–2231, 2007. 7

Mahdi Milani Fard, Yuri Grinberg, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup. Bellman error based feature generation using random projections on sparse spaces. In *Advances in Neural Information Processing Systems (NIPS - 26)*, pages 3030–3038, 2013. 7

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. 7

Rémi Munos. Performance bounds in L_p norm for approximate value iteration. *SIAM Journal on Control and Optimization*, pages 541–561, 2007. 5

Dirk Ormoneit and Saunak Sen. Kernel-based reinforcement learning. *Machine Learning*, 49:161–178, 2002. 23

Ronald Parr, Christopher Painter-Wakefield, Lihong Li, and Michael Littman. Analyzing feature generation for value-function approximation. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 737 – 744, New York, NY, USA, 2007. ACM. 7

Marek Petrik. An analysis of Laplacian methods for value function approximation in MDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2574–2579, 2007. 7

David Silver, Richard S. Sutton, and Martin Müller. Reinforcement learning of local shape in the game of go. In Manuela M. Veloso, editor, *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1053–1058, 2007. 7

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998. 2, 7

Richard S. Sutton, Csaba Szepesvári, Alborz Geramifard, and Michael Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008. 2

Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Morgan Claypool Publishers, 2010. 2, 3, 7

Sara A. van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, 2000. 8, 11, 27, 28

Hengshuai Yao, Csaba Szepesvári, Bernardo Ávila Pires, and Xinhua Zhang. Pseudo-MDPs and factored linear action models. In *IEEE Symposium on Adaptive Dynamic Programming And Reinforcement Learning (ADPRL)*, 2014. 23