Off-Policy Reinforcement Learning with Gaussian Processes

Girish Chowdhary*
School of Mechanical and Aerospace Engineering
Oklahoma State University
Stillwater, OK 74078
girish.chowdhary@okstate.edu

Miao Liu
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708
miao.liu@duke.edu

Robert C. Grande
Department of Aeronautics & Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139
rgrande@mit.edu

Thomas J. Walsh
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
twalsh@mit.edu

Jonathan P. How
Department of Aeronautics & Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139
jhow@mit.edu

Abstract
An off-policy Bayesian nonparametric approximate reinforcement learning framework, termed as GPQ, that employs a Gaussian Processes (GP) model of the value (Q) function is presented in both the batch and online settings. Sufficient conditions on GP hyperparameter selection are established to guarantee convergence of off-policy GPQ in the batch setting, and theoretical and practical extensions are provided for the online case. In particular, the convergence results in the batch case extend theoretical results on the Fitted Q-Iteration family of algorithms and the online results provide a theoretical grounding for the use of sparse, budgeted GP representations. These results reveal a reason for potential divergence of off-policy approximate reinforcement learning employing Gaussian kernels as well as hyperparameter selection conditions to eliminate this possibility. Empirical results demonstrate GPQ has competitive learning speeds in addition to its convergence guarantees and its ability to automatically choose its own basis locations.

Keywords: Reinforcement Learning, Gaussian Processes

Acknowledgements
This work was supported in part by the Aerojet Fellowship as well as the Office of Naval Research under MURI program award #N000141110688 and contract #N000140910625.

*A longer version of this work with complete proofs is currently in submission at the Conference on Neural Information Processing Systems (NIPS-13).
1 Introduction

Reinforcement learning (RL) [17] in continuous or large state spaces often relies on function approximation to maintain a compact model of the value (Q) function [4, 5, 10], but often requires a precise choice of features to achieve good performance. Gaussian Processes (GPs) [15] are Bayesian Nonparametric (BNP) models that are capable of automatically adjusting features based on the observed data and have been successfully employed in high-dimensional approximate RL domains [4], but several properties of RL with GPs, particularly convergence guarantees, off-policy learning, and exploration techniques have not been fully addressed.

More specifically, no convergence results for RL algorithms with GPs exist, and existing RL methods with GPs either require a planner [14, 3, 7] or are restricted to on-policy learning [4]. The latter approach is less general than off-policy RL, which enables learning the optimal value function using samples collected with a safe or exploratory policy. In addition, existing model-free GP RL methods (e.g. [4]) do not fully leverage the Bayesian notion of predictive variance inherent to a GP model in guiding the exploration strategy.

In this abstract, the three problems mentioned above for approximate RL using GPs are addressed: convergence, off-policy learning, and exploration. More specifically, a model-free off-policy approximate reinforcement learning technique, termed as GPQ, that uses a GP model to approximate the value function is presented. Sufficient conditions for convergence of GPQ to the best achievable optimal Q-function given the data (Q*) are presented in the batch and online settings, and it is shown that these properties hold even as features are added or removed to maintain computational feasibility. We also present an exploration scheme based on the GPs explicit confidence intervals over Q*. Unlike other recent papers on off-policy RL with fixed-parameter linear function approximation [18, 10, 8, 9], our approach allows the basis functions to be automatically identified from data. Furthermore, our condition for convergence reveals why GPQ or kernel base Fitted Q-Iteration [6] could diverge, and how this divergence can be prevented by tuning a regularization-like parameter of the GP. More broadly, this work helps to connect the fields of BNP modeling and RL by combining BNP strengths such as prior distributions and data-driven basis construction with RL architectures such as Fitted Q-Iteration [5] in a theoretically grounded manner.

2 Background

We consider domains modeled as a Markov Decision Process (MDP) [17] with standard notation: \( M = (S, A, \mathcal{P}, \mathcal{R}, \gamma) \), policy \( \pi : S \mapsto A \) and optimal value (Q) function \( Q^*(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r(s_t, a_t) + \max_{a'} \gamma Q^*(s_{t+1}, a')] \). Reinforcement learning is concerned with finding the optimal policy \( \pi^*(s) = \arg\max_a Q^*(s, a) \) when \( \mathcal{P} \) and \( \mathcal{R} \) are unknown. When an RL method learns the value function of the same policy with which samples were collected it is classified as on-policy. When the policy employed to obtain samples is different, it is termed off-policy. For continuous RL domains, linear value function approximation is often used to model the Q-function as a weighted combination of fixed bases \( \phi(s, a) \), that is \( Q(s_t, a_t) = \phi^T(s_t, a_t) \theta \). However, function approximation can cause divergence for off-policy RL methods [1, 19].

Gaussian Processes (GPs) [15] are BNP function approximation models: they do not specify a model structure and explicitly model noise and uncertainty. A GP is defined as a collection of random variables, any finite subset of which has a joint Gaussian distribution with mean (prediction) function \( m(z) \) and covariance kernel \( k(z', z) \), for input points \( z \) and \( z' \) (z = (s, a) in our case). Define \( K(Z, Z) \) as the kernel matrix with entries \( K_{l,m} = k(z_l, z_m) \). \( k(Z, z_{t+1}) \in \mathbb{R}^T \) denotes the kernel vector corresponding to the \( t + 1 \)th measurement, and \( \omega_n^2 \) represents the variance of the measurement uncertainty. The conditional probability can be calculated as a normal variable [15] with mean \( m(z_{t+1}) = \alpha^T k(Z, z_{t+1}) \), where \( \alpha = [K(Z, Z) + \omega_n^2 I]^{-1} y \) are the kernel weights, and covariance

\[
\Sigma(z_{t+1}) = k(z_{t+1}, z_{t+1}) + \omega_n^2 - k^T(Z, z_{t+1})[K(Z, Z) + \omega_n^2 I]^{-1} k(Z, z_{t+1})
\] (1)

In this paper, we use the sparsification method of [2], which allows for sequential updates. This sparsification algorithm works by building a dictionary of basis vector points. In order to determine when a new point should be added to the dictionary, a linear independence test is performed:

\[
\beta_{t+1} = k(z_{t+1}, z_{t+1}) - k(Z_d, z_{t+1})^T K(Z_d, Z_d)^{-1} k(Z_d, z_{t+1}).
\] (2)

When \( \beta_{t+1} \) is larger than a specified threshold \( \beta_{tol} \), then a new data point is added to the dictionary. Otherwise, the weights \( \alpha \) are updated, but the dimensionality of \( \alpha \) remains the same.

GPs have previously been used in on-policy model-free RL [4] and also model-based RL [14, 3, 7]. For linear fixed-basis (non-GP) value function approximation, several recent approaches ensure convergence in the online case by adding some form of regularization to TD algorithms, as done in GQ [10], LARS-TD [8], and RO-TD [9]. In contrast to these algorithms, GPQ attempts to directly minimize the Q-learning TD error and does not require a priori basis function placement. Other algorithms, such as Bellman Error Basis Functions [13], dynamically construct features for linear function approximators, but are primarily used in policy evaluation. In the batch case, several algorithms belonging to the Fitted Q-Iteration (FQI)
family of algorithms \cite{5} can guarantee off-policy convergence with specific function approximators, including averagers \cite{12} and certain forms of regularized least squares \cite{6}.

3 Off-Policy RL with a GP

Our goal is to perform posterior inference using available information so that the current estimate of the mean \( \hat{m} \) approaches the mean of \( \hat{Q} \). Let the current estimate of the mean of the Q-function be \( \hat{Q}(s,a) = \hat{m}(s,a) \). Since samples of \( \hat{Q} \) are not available, posterior inference needs to be performed using the best estimate of \( \hat{Q} \) at the current time as:

\[
\hat{Q}(s_t,a_t) = r(s_t,a_t) + \gamma \max_{a'}(\hat{Q}(s_{t+1}, a')).
\]

(3)

Whenever we update the model of \( \hat{Q}(s_t, a_t) \) with a new observation, the accuracy of the observation is dependent on the accuracy of the current model. Typically, the parameter \( \omega_n^2 \) is viewed as an uncorrelated, Gaussian measurement noise in GP literature. Here, we offer an alternative interpretation of \( \omega_n^2 \) as a regularization term, which accounts for the fact that current measurements are not necessarily drawn from the true model and therefore prevents our model from converging too quickly to an incorrect estimate of \( \hat{Q} \). As we show later, \( \omega_n^2 \) plays a pivotal role in preventing divergence as well.

3.1 Batch GP-Fitted Q-Iteration

Using GPs and the update rule in Equation \cite{3} in the batch setting gives us Algorithm 1, which we call GP-FQI because it is a member of the Fitted Q-Iteration \cite{5} family of algorithms. Below, we prove that GP-FQI can diverge if the regularization parameter is not properly set. However, we also prove that for any set of hyperparameters and desired density of data, a proper regularization constant can be determined to ensure convergence. We begin with a counter-example showing divergence in the batch setting if \( \omega_n^2 \) is insufficient, but show convergence when \( \omega_n^2 \) is large enough.

Consider a system with three nodes on the real line at locations \(-1, 0, \) and \(1\). At each time step, the agent can move deterministically to any node or remain at its current node. The reward associated with all actions is zero. All algorithms are initialized with \( \hat{Q}(z) = 1 \forall z, \gamma = 0.9999, \) and we use a RBF kernel with bandwidth \( \sigma = 1 \) in all cases. We consider two settings of the regularization parameter, \( \omega_n^2 = 0.1 \) and \( \omega_n^2 = 1 \). Figure 1 shows that when \( \omega_n^2 \) is set too low, the Bellman operator can produce divergence in the batch setting. If the regularization is set to the higher value, GP-FQI converges. In the following sections, we show that determining the sufficient regularization parameter \( \omega_n^2 \) depends only on the density of the data and the hyperparameters, not the initialization value of \( \hat{Q} \) or \( \gamma \).

In the following theorem, we show that in the case of finite data, a finite regularization term always exists which guarantees convergence.

**Theorem 1.** Given a GP with data \( Z \) of finite size \( N \), and Mercer kernel that is bounded above by \( k_{\text{max}} \), there exists a finite regularization parameter \( \omega_n^2 \) such that the Bellman operator \( T \) is a contraction in the batch setting. In particular, \( \omega_n^2 = 2(\|K(Z,Z)\|_\infty - k_{\text{max}}) \leq 2N \)

The crux of the proof is showing that the Bellman operator is a contraction if the term \( \|K(Z,Z)\|\|K(Z,Z) + \omega_n^2 I\|^{-1} \leq 1 \) when \( \omega_n^2 \) is set in this way. In the next theorem, we show that for a GP with infinite data that only adds data points which exceed the linear independence test \( \beta_{\text{tol}} \), a finite regularization term also exists.
Theorem 2. Given a GP with infinite data generated using a sparse approximation with acceptance tolerance $\beta_{tol}$, and given a Mercer kernel function that decays exponentially, there exists a finite regularization parameter $\omega_\beta^2$ such that the Bellman operator $T$ is a contraction in the batch setting.

The key to the proof is that $\|K(Z,Z)\| = \max_i \sum_j k(z_j, z_i)$, which is convergent for an infinite number of data points selected using the linear independence test in [2]. Theorem 2 provides a powerful insight into the convergence properties of GPs in the context of the Bellman operator. As the density of basis vectors increases or as the bandwidth of the kernel function grows, corresponding to decreasing $\beta_{tol}$, the basis vector weights $\alpha_i$ become increasingly correlated. As the weights become correlated, changing the weight at one basis vector also changes the weights of nearby basis vectors. It is this sharing of weights that can result in divergence, as seen in [1]. Theorem 2 shows that for a given $\beta_{tol}$ and kernel function, there exists a finite regularization parameter $\omega_\beta^2$ that will prevent divergence. In the next theorem, we bound the approximation error from using a sparse representation of a GP versus a full GP. The key to the proof is that the maximum error is linear in $\beta_{tol}$.

Theorem 3. If the sparse GP algorithm is used, the error $\|\mathbb{E}[^{Q}_t] - Q^*\|$ is uniformly, ultimately bounded for the Bellman operator.

3.2 Online Learning and Exploration with GPQ

In theory, GP-FQI provides a method for learning the Q-function online with provable convergence. Specifically, one could employ an $\epsilon$-greedy exploration policy to guarantee sufficient data collection and perform the batch updates after every step. However, because of the computational costs of such a “batch-sequential” algorithm, we now consider a different GPQ algorithm: Online GPQ (Algorithm 2). At each step of Online GPQ, an action is taken using a policy $\pi$ that ensures ergodicity of the induced Markov chain, and the value $y_t = r + \gamma \max_{b'} \bar{Q}_t(s', b)$ at location $z_t$ is calculated. The sparse online GPQ algorithm of [2] is used to determine whether or not to add a new basis vector to the active bases set $B^t$ and then the kernel weights are updated. Below we provide a set of sufficient conditions for convergence in the online case, where $C_t$ and $K_t^{-1}$ are positive definite matrices related to the posterior and the prior covariance. The proof uses similar techniques to Theorem 17 of [11], using an ODE representation of $\alpha$: $\dot{\alpha}(t) = \mathbb{E}_\pi [q_t S_t]$ and then showing that $\alpha \rightarrow \alpha^*$ for each active basis set.

Theorem 4. with an ergodic sampling policy $\pi$, for each active basis set, a sufficient condition for convergence of $\bar{m}(z_t) \rightarrow m^*(z_t)$ as $t \rightarrow \infty$ online GPQ is $\mathbb{E}_\pi \left[ C_t k_t k_t^T + K_t^{-1} k_t k_t^T \right] \geq \gamma \mathbb{E}_\pi \left[ C_t k_t k_t^T + K_t^{-1} k_t k_t^T \right]$, where $k_t^\alpha = \max\alpha' (k_t^T x_{t+1}, a') \alpha_t$.

While $\epsilon$-greedy exploration helps guarantee convergence, in practice it is often inefficient at gathering useful samples. A guiding principle of more sample-efficient RL algorithms (e.g. [15]) is “optimism in the face of uncertainty”, specifically using over-estimates of the value function and greedy actions to balance exploration/exploitation. We propose using the upper confidence tails from the GP as an optimistic value function. Specifically, for any point $\langle s, a \rangle$, the GP will report an upper confidence tail of $m(s) + 2\Sigma(s, s_{+1})$ where $m$ and $\Sigma$ are defined in Section 2. We can then modify Online GPQ to use greedy actions with respect to this upper tail and change the GP update to $\dot{Q}(s, a) = r(s) + \gamma \max_a[Q(s_{+1}, a) + 2\Sigma(s, a)]$.

That is, we use the upper tail of the next state’s Q-value in the Bellman update to overestimate the value function (reminiscent of Model-Based Interval Estimation [16]). We leave a detailed theoretical analysis of this technique to future work.

4 Empirical Results and Conclusions

We now present experiments with two variants of Online GPQ, one with the $\epsilon$-greedy exploration and the other using the optimistic values mentioned above. Comparisons are made to Q-learning with function approximation and fixed bases (QL-FB), the GQ [10] algorithm with fixed bases, and tabular Q-learning. Our experiments cover three domains, a discrete 5 x 5 Gridworld, a continuous state Inverted Pendulum, and continuous state Puddle-World. After parameter tuning and cross-validation, the policy learned from all these methods for the three domains are evaluated based on discounted cumulative reward averaged over 20 independent runs and are shown in Figure 2.

In the Gridworld experiments, while all of the algorithms find the optimal policy, the GPQ based methods converge much faster by quickly identifying important areas for basis points. We also see that optimistic exploration using the GP’s variance is advantageous, as the algorithm very quickly uncovers the optimal policy. In Inverted pendulum, GPQ quickly finds adequate bases and converges to a near optimal policy while GQ requires more samples. Beyond the “best parameter” results shown in this graph, we also observed GPQ methods are more resilient against small quantizations (budgets) because they are able to select their own bases, while GPQ and QL-FB are far more sensitive to the number and placement of bases. In Puddle-World, basis placement is more challenging and GPQ sometimes converges to a cautious (puddle adverse) policy. Multiple lines are shown for QL-FB and GQ, depicting their best and worst case in terms of parameter settings, as they were extremely sensitive to these settings in this domain. While the best case versions of GQ and QL-FB reached better policies than GPQ, in the worst case, their Q-values appear to diverge, as illustrated in the final graph showing the number of steps to the goal, where the worst-case QL-FB and GQ stay near the initial state for all 500 steps.
steps. While GQ has convergence guarantees when data comes from a fixed policy, those conditions are violated here, hence the potential for divergence. In summary, while very careful selection of parameters for QL-FB and GQ leads to slightly better performance, GPQ performs almost as well as their best case with less information (since it does not need the bases a priori) and far outperforms their worst-case results.

In this abstract, we presented a NPB framework (GPQ) that uses GPs to approximate the value function in off-policy RL. We presented algorithms using this framework in the batch and online case and provided sufficient conditions for their convergence. Our results show that GPQ's representational power allows it to perform as well or better than other off-policy RL algorithms and that a regularization-like term can help decouple parameters and avoid divergence.

References