PAC-BAYESIAN RANDOMIZED VALUE FUNCTION WITH INFORMATIVE PRIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Randomized value function has been shown as an effective exploration strategy for reinforcement learning (RL), which samples from a learned estimation of the distribution over the randomized Q-value function and then selects the optimal action. However, value function methods are known to suffer from value estimation error. Overfitting of value function is one of the main reasons to estimation error. To address this, in this paper, we propose a Bayesian linear regression with informative prior (IP-BLR) operator to leverage the data-dependent prior in the learning process of randomized value function, which can leverage the statistics of training results from previous iterations. We theoretically derive a generalization error bound for the proposed IP-BLR operation at each learning iteration based on PAC-Bayesian theory, showing a trade-off between the distribution obtained by IP-BLR and the informative prior. Since the optimal posterior that minimizes this generalization error bound is intractable, we alternatively develop an adaptive noise parameter update algorithm to balance this trade-off. The performance of the proposed IP-BLR deep Q-network (DQN) with adaptive noise parameter update is validated through some classical control tasks. It demonstrates that compared to existing methods using non-informative prior, the proposed IP-BLR DQN can achieve higher accumulated rewards in fewer interactions with the environment, due to the capabilities of more accurate value function approximation and better generalization.

1 INTRODUCTION

Deep reinforcement learning (RL) has demonstrated outstanding performance over many challenging tasks (Mnih et al., 2015; Silver et al., 2018). However, the design of an efficient exploration strategy for the intractably large state and action spaces still remains a main challenge. Noise injection is a commonly used exploration strategy in deep RL. For example, the ϵ-greedy strategy that takes action uniformly with probability ϵ is effective for tabular and linear function approximation-based Q-learning, and has also succeeded in deep Q-network-based methods (Mnih et al., 2015). By adding random perturbation into the coefficients of value function, randomized value function method has been introduced for exploration. In standard RL setting, the value function is deterministic w.r.t. the state and action, which explilcitly indicates how valuable a certain state or action is. Policy built on randomized value function, on the other hand, can behave stochastically and be induced to a more efficient exploration. Using linear function approximation, randomized least-squares value iteration (RLSVI) (Osband et al., 2019) leverages Bayesian linear regression (BLR) (Bishop, 2006) to learn an approximation to the posterior distribution of the randomized Q-value function. In deep RL, randomized noise is injected into the parameters of neural network, while the noise parameter is trained together with the network (Fortunato et al., 2017; Plappert et al., 2017).

RL algorithms are known to suffer from value estimation error, which leads to suboptimal policy. It has been proved by Hasselt et al. (2016) that estimation error of any kind can lead to overestimation of action's value in Q-learning. In actor-critic methods, the overestimation bias exists in the approximation of critic (Fujimoto et al., 2018). In randomized value function method, the value estimation error is also inevitable. One of the main reasons for the occurrence of estimation error in function approximation is overfitting on a finite training set by minimizing the empirical error (risk). If we can learn a regressor by minimizing the expected error (risk) of the true distribution, e.g., ideally in the setting of supervised learning, then the regressor can be generalized to the unseen

data (Bishop, 2006). The distance between the empirical risk and true risk is called the generalization error (Germain et al., 2016). Theoretically, the generalization error can be reduced to zero by minimizing directly on the true risk, which, however, is usually infeasible since the true distribution is unfortunately not available in practice.

Fortunately, by assuming a distribution over coefficients, we can bound the generalization error by PAC-Bayes. In classification tasks, algorithms derived from PAC-Bayesian theory have been proved to guarantee a tight generalization error bound (Germain et al., 2009). PAC-Bayes provides an upper bound for generalization error in terms of the empirical risk and Kullback-Leibler (KL) divergence between the prior and posterior distributions, which prevents the posterior of coefficients from being updated to overfitting by minimizing the empirical risk. In previous works, the assumption of prior is often non-informative, such as fixed Gaussian distribution (Germain et al., 2016). Through a careful design, however, the prior distribution can also be optimized for a better generalization. For example, previous works that use data-dependent priors have achieved a tighter bound and better performance on classification tasks (Parrado-Hernández et al., 2012), where the data-dependent prior can be obtained by pre-learning on a part of the whole training dataset. This motivates us to incorporate PAC Bayesian theory with informative prior into the batch RL algorithms, since iterative training does provide rich data-dependent prior information that can be exploited to reduce the generalization error incurred by function approximation.

In this paper, we propose to use informative prior for the posterior update in randomized value function exploration. To further reduce the generalization error incurred by function approximation, we propose to optimize an upper bound provided by employing PAC-Bayesian theory during the coefficient update. Our main contributions are as follows.

- We propose to use the Bayesian linear regression with an informative prior (IP-BLR) distribution to learn the distribution estimation of the randomized Q-value function. For each IP-BLR iteration, we construct the informative prior by using posterior from the previous iteration.
- We derive an upper bound for the generalization error in each IP-BLR iteration based on the PAC-Bayesian theory. This generalization error bound shows a trade-off between the posterior of IP-BLR operation and the informative prior distribution. We further provide an analogy to the optimal Gibbs posterior (Alquier et al., 2016) that minimizes this generalization error bound, which is unfortunately intractable and not piratically feasible in the training process.
- Alternatively, we aim to manipulate the noise parameter β to balance the IP-BLR posterior and the prior in the final posterior distribution, based on which propose a practical deep RL algorithm called IP-BLR DQN.
- Through extensive experiments, we demonstrate that the proposed IP-BLR DQN presents a quicker jumpstart for the learning process and significantly improves the asymptotic performance than the randomized value function baselines, which mainly benefits from the proposed informative prior and adaptive noise parameter update.

2 BACKGROUND

Reinforcement Learning: In RL setting, the environment is modeled as an MDP, denoted by a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P} >$ with state space $\mathcal{S}$, action space $\mathcal{A}$, reward space $\mathcal{R}$ and environment dynamic $\mathcal{P}$. An agent improves its policy π by interacting with the environment. At each time step t, the agent observes the state $s_t \in \mathcal{S}$ and takes an action $a_t \in \mathcal{A}$ according to its current policy $\pi(a_t|s_t) : \mathcal{S} \to \mathcal{A}$. Afterwards, the agent will receive a reward $r(s_t, a_t) : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$. Then the state of the environment changes to the next state s_{t+1} with probability $P(s_{t+1}|s_t, a_t) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, which is determined by the environment dynamic $\mathcal{P}$. The RL problem aims to find an optimal policy that maximizes the accumulated discounted reward $G_t^\pi = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k)$, where $\gamma \in [0, 1]$ is a discount factor. The state-action value function Q is defined as the expectation of return $Q(s_t, a_t) = \mathbb{E}\left[\sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k)|s_t, a_t\right]$, with the following Bellman equation specifying the relationship between the current state-action value and its successor:

$$Q(s_t, a_t) = \mathbb{E}\left[r(s_t, a_t))\right] + \gamma \mathbb{E}_{P,\pi}[Q(s_{t+1}, a_{t+1})]. \tag{1}$$

The optimal state-action value function Q^* is given under the optimal policy π^* that solves the RL problem. Then the Bellman optimality operator $\mathcal{T}$ is given by:

$$\mathcal{T}^*Q(s_t, a_t) = \mathbb{E}\left[r(s_t, a_t)\right] + \gamma \mathbb{E}_P\left[\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})\right],$$

which has been proved to be a contraction mapping with a unique fixed point at Q^*. It hence guarantees the convergence of Q-learning based approaches.

PAC-Bayesian Theory: Suppose that we have M training samples $\{(X_i, Y_i)\}_{i=1}^M \in (\mathcal{X} \times \mathcal{Y})^M$, where $X_i = (X_i^1, X_i^2, ..., X_i^p)^{\mathrm{T}}$ and Y_i is a one-dimensional label. In the context of Q-learning, $\mathcal{X}$ corresponds to $\mathcal{S} \times \mathcal{A}$ and Y is the temporal difference (TD)-target. The goal of supervised learning is to learn a predictor $f \in \mathcal{F}$, where $\mathcal{F}$ is the function space : $\mathcal{X} \to \mathcal{Y}$. To quantify the prediction performance, a loss function $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is given. Hence the empirical risk $\hat{R}_\ell(f)$ is defined as the average loss of these M samples $\hat{R}_\ell(f) = \frac{1}{M}\sum_{i=1}^M \ell(f(X_i), Y_i)$. Suppose that we have the underlying data distribution $\mathcal{D}$ over $(\mathcal{X}, \mathcal{Y})$, the true risk is given by $R_\ell(f) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}\left[\ell(f(X), Y)\right]$, which measures the predictor f's generalization performance. During the training process, we only have access to the empirical risk that is used to train this predictor, which may cause overfiting. PAC-Bayesian theory provides an appealing oracle inequality that bounds the true risk with empirical risk, given in the following (Alquier et al., 2016).

Theorem 1. *For a given distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, any prior distribution J_f and posterior distribution P_f over the function space $\mathcal{F}$, a non-negative parameter λ, and a loss function $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, with a probability of at least $1 - \delta$ we have:*

$$\mathbb{E}_{P_f}[R_\ell(f)] \leq \mathbb{E}_{P_f}[\hat{R}_\ell(f)] + \frac{1}{\lambda}\left[D_{\mathrm{KL}}(P_f||J_f) + \ln\frac{C}{\delta}\right], \tag{2}$$

where $D_{\mathrm{KL}}(P_f||J_f)$ denotes the Kullback-Leibler (KL) divergence between P_f and J_f, and $C = \mathbb{E}_{(X',Y')\sim\mathcal{D}}\mathbb{E}_{J_f}\exp\left\{\lambda[R_\ell(f) - \hat{R}_\ell(f)]\right\}$.

3 Deep BLR-Q Learning with Informative Prior

In randomized least-squares value iteration (RLSVI) (Osband et al., 2019), following the basic assumption of Bayesian linear regression (BLR) (Bishop, 2006), the true Q-value $q_{s,a}$ is considered as the regression from a linear regression model with Gaussian noise $\epsilon \sim \mathcal{N}(0, \beta^{-1})$:

$$q_{s,a} = Q(s, a) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\theta}}(s, a) + \epsilon, \tag{3}$$

where $\boldsymbol{\phi}_{\boldsymbol{\theta}}(s, a)$ is the feature vector extracted by the deep neural network with weights $\boldsymbol{\theta}$, and $\boldsymbol{w}$ denotes the model parameter vector with dimensionality of Ω. For the ease of notation, we interchangeably use $\boldsymbol{\phi}(s, a)$ and $\boldsymbol{\phi}_\theta(s, a)$ throughout this paper. Suppose that we have a batch of transitions $\{(s_i, a_i, r_i, s_i')\}_{i=1}^M$ for current training iteration. The TD-target value $y_i \in \mathcal{Y}$ is

$$y_i = r_i + \gamma \max_{a'} \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\theta}}(s_i', a'). \tag{4}$$

Considering statistical independence of different transitions in this batch, which holds by randomly sampling from the replay buffer (Mnih et al., 2015), the likelihood function can be expressed as

$$p\left(\boldsymbol{y}|\boldsymbol{w}, \beta\right) = \prod_{i=1}^M \mathcal{N}\left(y_i|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\theta}}(s, a), \beta^{-1}\right). \tag{5}$$

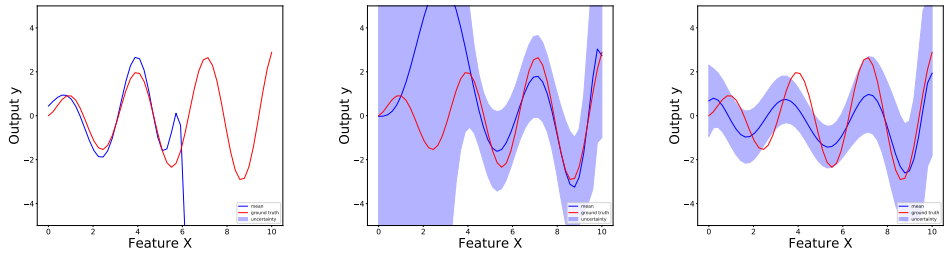
Assuming the prior distribution as $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{S})$ and referring to the Bayes rule, we can obtain the posterior distribution of $\boldsymbol{w}$, which is still a multivariate Gaussian distribution:

$$p(\boldsymbol{w}|\boldsymbol{y}, \beta) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m'}, \boldsymbol{S'}), \tag{6}$$

with the distribution parameters given by (Bishop, 2006):

$$\boldsymbol{m'} = \boldsymbol{S'}\left(\boldsymbol{S}^{-1}\boldsymbol{m} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y}\right), \boldsymbol{S'} = \left(\boldsymbol{S}^{-1} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}, \tag{7}$$

where $\boldsymbol{\Phi} = (\boldsymbol{\phi}_{\boldsymbol{\theta}}(s_1, a_1), \cdots, \boldsymbol{\phi}_{\boldsymbol{\theta}}(s_M, a_M))^{\mathrm{T}}$. Refer to Appendix A.1 for more details.



(a) (b) (c)

Figure 1: Example to illustrate the motivation of introducing informative prior to BLR: BLR with the non-informative prior trained on (a) $\mathcal{T}_{[0,6]}$ and (b) $\mathcal{T}_{[4,10]}$, respectively, and (c) BLR with the informative prior trained on $\mathcal{T}_{[4,10]}$.

The training of RL is an iterative process. In each iteration, previous randomized value function-based algorithms assume a non-informative prior distribution to learn the model parameters, which however is constrained on the current batch. In fact, the posterior distribution of model parameters $\boldsymbol{w}$ from the last iteration can be utilized as an informative prior for the current training iteration, able to provide richer information including the estimation of mean and variance parameters from the previous batches. The advantage of using informative prior comes from the nature of Bayesian approach, which can be verified by a simple regression task as shown in Figure 1. Here, we aim to fit function $f(x) = \sqrt{x}\sin(\frac{3}{2}x)$ with the test set $\mathcal{T}_{[0,10]} \triangleq \{(x_i, y_i)|x_i \in [0, 10]\}$. Under a non-informative prior assumption on the Gaussian distribution of zero mean and fixed variance, we first run BLR on training set $\mathcal{T}_{[0,6]}$. The test result for BLR in Figure 1(a) shows huge uncertainty in unseen area $\mathcal{T}_{[6,10]}$. Then we conduct BLR by treating the posterior distribution of model parameters from Figure 1(a) as informative prior in training set $\mathcal{T}_{[4,10]}$. The result in Figure 1(c) demonstrates that the learned estimation for $\mathcal{T}_{[0,4]}$ has been retained even though it is not contained in the current training set $\mathcal{T}_{[4,10]}$. In comparison, with the non-informative prior assumption, Figure 1(b) shows that the test result for BLR trained on the training set $\mathcal{T}_{[4,10]}$ holds huge uncertainty in $\mathcal{T}_{[0,4]}$.

Therefore, we propose to utilize the informative prior in BLR to learn a randomized Q-value function. Note that in RL training, due to the slow update for policy π, the training set may change slowly and IP-BLR may quickly lose uncertainty, which makes the agent prone to get stuck in a suboptimal policy. To keep the exploration of uncertainty, we use $\sigma^2\boldsymbol{I}$ as the prior variance instead of the posterior covariance matrix $\boldsymbol{S}'$ in (7) from the last iteration. We then give the convergence proof for the BLR Q-learning with informative prior. Analogous to Bellman optimality operator, we formally define the IP-BLR Bellman optimality operator, as follows.

Definition 1. *(IP-BLR operator) In each iteration, we set $\sigma^2\boldsymbol{I}$ and the posterior mean $\boldsymbol{m}'_p$ from the pervious iteration as the informative prior $\mathcal{N}(\boldsymbol{m}'_p, \sigma^2\boldsymbol{I})$ for the distribution of $\boldsymbol{w}$, based on which the prior predictive distribution for Q-value $q_{s,a}$ can be derived as J_q: $p(q_{s,a}) \sim \mathcal{N}(\boldsymbol{m}_p'^{\mathrm{T}}\phi(s,a), \beta^{-1}\boldsymbol{I} + \sigma^2\phi_{\boldsymbol{\theta}}(s,a)^{\mathrm{T}}\phi_{\boldsymbol{\theta}}(s,a))$. We define the following IP-BLR Bellman optimality operator H^* that changes the estimation of Q-value distribution based on BLR with the above informative prior.*

$$H^* q_{s,a} \sim \mathcal{N}(\boldsymbol{m}'^{\mathrm{T}}\phi(s,a), \beta^{-1}\boldsymbol{I} + \phi(s,a)^{\mathrm{T}}\boldsymbol{S}'\phi(s,a)), \tag{8}$$

where we have $\boldsymbol{m}' = (\boldsymbol{I} + \sigma^2\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}(\boldsymbol{m}'_p + \sigma^2\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y})$ based on (7). The proof follows eq.(3.59) in Bishop (2006).

We update the posterior of $\mathbf{w}$ by IP-BLR and optimize $\boldsymbol{\theta}$ together to the direction of the TD-target. Thus we can have:

$$\|\mathbb{E}q - q^*\|_\infty \leq \frac{1}{1-\gamma}\|\mathbb{E}q - \mathbb{E}H^*q\|_\infty, \tag{9}$$

where the equality holds when $\mathbb{E}q = q^*$.

Proof. See Appendix A.2 for details. □

4 PAC-BAYESIAN GENERALIZATION BOUND FOR RANDOMIZED VALUE FUNCTION APPROXIMATION

Randomized value function $Q \in \mathcal{Q} : \mathcal{S} \times \mathcal{A} \to \mathcal{Y}$ in (3) provides an efficient method for exploration. Iteratively applying IP-BLR Bellman operator and update $\boldsymbol{\theta}$ to the TD-target can lead the mean of the estimated distribution for Q-value $q_{s,a}$ to the optimal state-action value $q^*_{s,a}$, which theoretically requires the exhausting visit of state-action space. However, in practical RL training, especially for deep RL algorithms, the true distribution $\mathcal{D}^\pi$ under current policy π over $\mathcal{S} \times \mathcal{A} \times \mathcal{Y}$ is not available. Since we cannot get samples for the whole state-action space, a commonly used alternative is to sample a batch of transitions for training. In this section, we derive an oracle generalization inequality for each IP-BLR training process based on the PAC-Bayesian theory.

Theorem 2. *Given the true distribution $\mathcal{D}^\pi$ over $\mathcal{S} \times \mathcal{A} \times \mathcal{Q}^{target}$ under the current policy π, in each IP-BLR training iteration with fixed $\boldsymbol{\phi}(s, a)$, for all prior distributions J_w, posterior distributions P_w over model parameters w and corresponding predictive distribution J_q and P_q over Q-value $q_{s,a}$, a non-negative λ, with a probability of at least $1 - \delta$ we have:*

$$\|\mathbb{E}_{P_q} q_{s,a} - q^*_{s,a}\|_\infty \leq \frac{\mathbb{E}_{P_q}\left(\|q_{s,a} - \mathbb{E}H^* q_{s,a}\|_2\right)}{1-\gamma} + \frac{1}{\lambda}\left[D_{\mathrm{KL}}\left(P_q||J_q\right) + \ln \frac{\mathbb{E}_{\mathcal{D}^\pi}\left[\psi_J\left(s,a,y\right)\right]}{\delta}\right], \tag{10}$$

where $\psi_J\left(s,a,y\right) = \mathbb{E}_{J_q} \exp\left[\lambda\varphi(q_{s,a})\right]$, $\varphi\left(q_{s,a}\right) = \|q_{s,a} - q^*_{s,a}\|_\infty - \frac{1}{1-\gamma}\|q_{s,a} - \mathbb{E}H^* q_{s,a}\|_2$.

Proof. See Appendix A.3 for details. □

In Theorem 2, we provide an upper bound for the proposed IP-BLR optimality Bellman operator. The LHS of (10) provides a oracle distance between optimal Q-function and current mean of randomized Q-function. On the RHS, the first term indicates that the posterior P_q should match with the mean of IP-BLR, while the second term specifies a divergence between P_q and J_q. The third term measures the distance between the optimal Q-function and the mean of randomized value function after IP-BLR operation. When δ and prior J_w are selected, the model parameters $\boldsymbol{w}$' posterior distribution P_w is the unique one that can be optimized. Therefore the bound in (10) suggests a trade-off redbetween the IP-BLR operator $H^* q_{s,a}$ and the prior distribution J_q, resulting in the following optimization problem:

$$\min_{P_q} \mathbb{E}_{P_q} \frac{1}{1-\gamma}\|q_{s,a} - \mathbb{E}H^* q_{s,a}\|_2 + D_{\mathrm{KL}}(P_q||J_q). \tag{11}$$

This objective function is convex over P_q. By applying the KKT condition, we can get the optimal Gibbs posterior (Alquier et al., 2016) as

$$P_q : p(q_{s,a}) = \frac{1}{Z} J_q(q_{s,a}) \exp\{-\frac{\lambda}{1-\gamma}\|q_{s,a} - \mathbb{E}H^* q_{s,a}\|_2\} \tag{12}$$

where $Z = \int J_q(q') \exp\{-\frac{\lambda}{1-\gamma}\|q' - \mathbb{E}H^* q_{s,a})\|_2\} dq'$ is the normalization term. This Gibbs posterior is hard to compute and sample due to the intractable term Z. Markov Chain Monte-Carlo (MCMC) method is a commonly used method to sample the Gibbs posterior (Alquier & Biau, 2013), but it is too slow for big datasets. Another method is to approximate Gibbs posterior by a set of Gaussian distribution based on the variantional inference (VI) (Alquier et al., 2016). However, solving the variantional lower bound is still costly in our iterative setting. Instead, in the next section, we alternatively propose to manipulate noise parameter β.

5 ADAPTIVE NOISE PARAMETER UPDATE

Reviewing (7), the relationship between the informative prior $\boldsymbol{m}$, $\boldsymbol{S}$ and the current information $\boldsymbol{\Phi}$, $\boldsymbol{y}$ is controlled by the noise distribution parameter β. In BLR, β is a preset hyper-parameter (Bishop, 2006). Recalling the suggestion of Theorem 2, we should solve the trade-off between the IP-BLR and prior. In this section, we assume that the prior of β follows uniform distribution and propose an adaptive noise parameter update.

From Bayes rule, we have

$$p(\boldsymbol{y}|\boldsymbol{w},\beta)p(\boldsymbol{w}) = p(\boldsymbol{w}|\boldsymbol{y},\beta)p(\boldsymbol{y}|\beta). \tag{13}$$

Substituting (5), (6) and prior distribution $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{m}, \sigma^2\boldsymbol{I})$ into (13), we obtain

$$p(\boldsymbol{y}|\beta) \propto |\boldsymbol{\Sigma_y}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma_y}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right\}, \tag{14}$$

where $\boldsymbol{\mu}^T = \beta\sigma^{-2}\boldsymbol{m}^T\boldsymbol{S'}\boldsymbol{\Phi}\boldsymbol{\Sigma_y}$ and $\boldsymbol{\Sigma_y}^{-1} = \beta\mathbf{I} + \beta^2\boldsymbol{\Phi}\boldsymbol{S'}\boldsymbol{\Phi}^T$. According to the maximum likelihood estimation, we want to maximize β's posterior $p(\beta|\boldsymbol{y}) \propto p(\boldsymbol{y}|\beta)p(\beta)$ (Tipping, 2001), which is equivalent to the following optimization problem:

$$\max_{\beta} -\frac{1}{2}\log|\boldsymbol{\Sigma_y}| - \frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma_y}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}). \tag{15}$$

The objective function is concave over β. By setting derivatives w.r.t. β of (15) to zero, we get the update for the noise parameter β:

$$\beta_{new} = 2M/\left[\mathrm{Tr}(\boldsymbol{\Sigma_y}) + \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{m'}\|^2\right] \tag{16}$$

The first term in the denominator is the trace of the covariance matrix of the Q-value's posterior distribution conditioned on β, which reflects the uncertainty for the estimation of Q-value. The second term is the mean square error (MSE) with respect to the BLR's posterior mean. The value of these two terms consistently reflect the performance of the posterior parameter in the IP-BLR operator. A larger trace of posterior covariance matrix or a larger MSE will incur a worse performance and a smaller β, which in turn will make the posterior distribution closer to the prior distribution. As shown in (7) and (15), the update of β will result in the change of the posterior parameters $(\boldsymbol{m'}, \boldsymbol{S'})$ and vice versa. Hence we alternately perform the IP-BLR operation and β update as shown in Algorithm 2 to find an appropriate update for $(\boldsymbol{m'}, \boldsymbol{S'}, \beta)$.

To summarize, we propose IP-BLR DQN with adaptive noise parameter in Algorithm 1 in Appendix A.4. The agent explores the environment guided by the variance of Q-value estimation instead of randomly injected noise. The parameter $\boldsymbol{\theta}$ from feature extraction layer is learned by vanilla DQN backpropagation with $\boldsymbol{w}$ fixed. Besides, we periodically update the distribution parameters of $\boldsymbol{w}$ using Algorithm 2 in Appendix A.4.

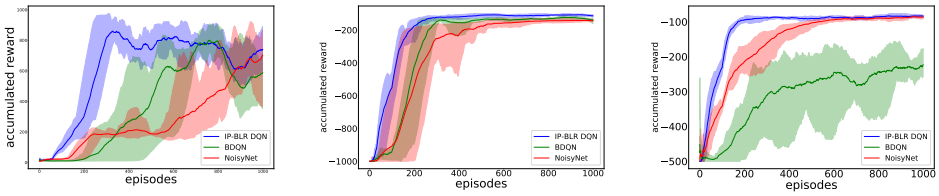
6 Experiments

In this section, we conduct several empirical experiments to validate the effectiveness of our proposed IP-BLR methods. We compare our method with Bayesian DQN (BDQN) (Azizzadenesheli et al., 2018) and NoisyNet DQN (Fortunato et al., 2017), which all are randomized value function baselines. In NoisyNet, noise is added to the weights of deep RL to introduce randomized value function for exploration. Besides, we empirically show the outcome of the adaptive noise parameter update. We also conduct experiments to study how the vital hyper-parameters (i.e., feature dimension and prior variance σ) affect the overall performance.

6.1 Comparison on Classical Control Tasks

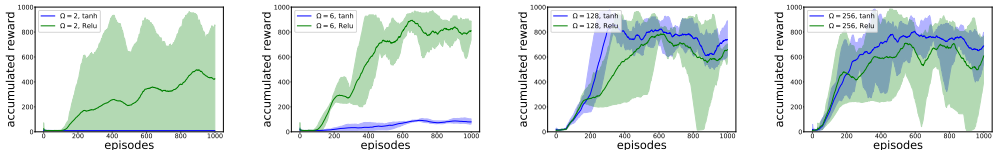
We choose classical control tasks: MountainCar, CartPole and Acrobot provide by OpenAI (Brockman et al., 2016) and use the basic reward function setting. In MountainCar and Acrobot, the agent will get reward -1 for each time step, until the episode ends. In CartPole, the agent will get reward $+1$ for keeping the pole upright until the pole and cart deviate too much from the perpendicular and the center, respectively. We set the maximum time steps for one episode to 1000 for CartPole and MountainCar, and to 500 for Acrobot.

We use a neural network to represent Q-value function in NoisyNet and feature extraction in BDQN and IP-BLR DQN, which has the same dense layers setting in all the three algorithms. Besides, in BDQN and our IP-BLR DQN, we use the same feature dimension and σ^2. In our IP-BLR DQN, we perform 20 iterations in Algorithm 2.



(a) CartPole (b) MountainCar (c) Acrobot

Figure 2: Comparison between IP-BLR DQN, BDQN and NoisyNet on CartPole, MountainCar and Acrobot.



(a) $\Omega = 2$ (b) $\Omega = 6$ (c) $\Omega = 128$ (d) $\Omega = 256$

Figure 3: IP-BLR DQN with Tanh and ReLU after the last dense layer with the feature dimension Ω=[2, 6, 128, 256] on CartPole.

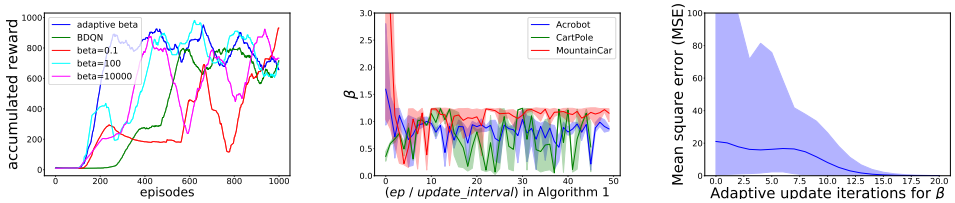
We run each training process for 1000 episodes and plot the learning curves on three test environments in Figure 2. The results are obtained by running each algorithms with five different random seeds. The accumulated reward averages on the last 100 episodes. We can see that IP-BLR DQN has the best asymptotic performance on all the three environments, while BDQN is mostly better than NoisyNet. IP-BLR DQN has less performance degradation during the training process compared to the other two comparison methods, and presents a quicker jumpstart on the training process.

6.2 Hyper-parameters

In order to demonstrate the effectiveness of the proposed IP-BLR operator and the adaptive noise parameter update, we conduct several experiments to study and provide an empirical analysis of the hyper-parameter selection for our IP-BLR DQN.

The feature dimension Ω is a vital hyper-parameter for both BDQN and IP-BLR DQN. A small dimension may not be representative for the value function of a task, while a large dimension may exceed the capability of BLR in the high-dimensional space. We plot the learning curves of IP-BLR DQN with Tanh and ReLU activation after the last dense layer in MountainCar, with feature dimension set to $[2, 6, 128, 256]$ in Figure 3. The results show that the feature dimension affects the performance of IP-BLR DQN, and the training process suffers from a too small or too large feature dimension, which is mainly dependent on the task and the activation function used. With Tanh as the activation function, we suggest that the feature dimension should be comparable to the range of Q-value in tasks, since Tanh ranges from -1 to 1. In comparison, with ReLU as the activation function, the performance is not that sensitive to Ω. However, as shown in Figure 3(a) and 3(d), a too large or too small Ω still will degrade the performance.

In addition, IP-BLR DQN reduces to BDQN without the proposed informative prior distribution and adaptive noise parameter update. To validate the effectiveness of the utilization of informative prior distribution, we further compare IP-BLR DQN with BDQN and IP-BLR DQN without the adaptive β update. Referring to Theorem 2, we propose the adaptive update for β to balance the trade-off between the IP-BLR operation and informative prior. We compare the performance on Cartpole achieved by BDQN with $\beta = 100$, IP-BLR DQN with adaptive noise parameter update, and IP-BLR DQN with fixed $\beta \in [0.1, 1, 100, 10000]$ in Figure 4(a). It shows that the training process of IP-BLR DQN is more stable and has a quick jumpstart, which mainly benefits from the adaptive noise parameter update. In addition, under the same preset $\beta = 100$, IP-BLR DQN has faster convergence than BDQN, which demonstrates the effectiveness of utilizing the informative prior. The curves of β during the training process in three environments are shown in Figure 4(b),



(a) $\Omega = 6$ (b) $\Omega = 2$ (c) $\Omega = 12$

Figure 4: (a) Comparison between IP-BLR DQN with/without adaptive β update and BDQN, (b) dynamic trend of β during training, (c) trend of the MSE reduction.

presenting a dynamic trend. The trend of MSE reduction that is averaged on all the update based on Algorithm 2 is shown in Figure 4(c), demonstrating that our iterative update for β can effectively reduce the MSE loss during training in DQN.

6.3 Experiment Setup

In all the experiments, we use Adam optimizer for learning the neural network with two hidden layers of dimensions 50 and 256, respectively, by using the learning rate 3×10^{-4}. For RL hyper-parameters, we set the discounted factor to $\gamma = 0.99$, replay buffer size to 1×10^5 and mini-batch size to 64. For BDQN and IP-BLR DQN, we set the training set size as 1000.

7 Related Work

Previous attempts to extend randomized value function strategy to deep RL include the direct Q-function approximation by DNN (Azizzadenesheli et al., 2018) and duplicates of Q-networks to approximate the posterior distribution (Osband et al., 2016). In statistic learning, PAC-Bayesian theory firstly provides a probably approximately correct (PAC) guarantee on generalization error for Bayesian algorithms (McAllester, 1999). For regression setting, a PAC-Bayesian bound with unbounded loss function is given by (Germain et al., 2016) and a practical Gaussian process regression algorithm is studied in (Reeb et al., 2018). In RL, a PAC-Bayes generalization bound on reparameterized RL is provided with randomization on policy parameters by Wang et al. (2019), while our work is done on standard RL with randomization on value function parameters. Fard & Pineau (2010) propose a PAC-Bayes bound for model-free RL on discrete state spaces without introducing function approximation, while our bound and algorithm are derived for the complex continuous state space and randomized value function.

PAC learning has also been introduced to RL for the measure of exploration efficiency, which is known as probably approximately correct in Markov decision processes (PAC-MDP). PAC-MDP requires RL algorithms to behave non-optimal compared to optimal performance within a small range in a certain number of time steps that is polynomial w.r.t. the size of state space and action space. RL algorithms such as E^3 (Kearns & Singh, 2002) and R-max (Brafman & Tennenholtz, 2003) that satisfy PAC-MDP all show efficient exploration since near-optimal policy can be achieved in polynomial time. Recent studies have advanced PAC-MDP to being capable of dealing with rich observations (Dann et al., 2018; Du et al., 2019).

8 Conclusion

In this paper, we have proposed for the randomized value function-based RL an IP-BLR operator by leveraging the data-dependent prior in training to reduce the estimation error of value function. Based on PAC-Bayesian theory, we theoretically derived a generalization error bound for the proposed IP-BLR operation at each training iteration, and developed an practically feasible algorithm for adaptive noise parameter update to balance the trade-off between the posterior of IP-BLR operation and the informative prior distribution. Experiments on classical control tasks have shown that due to the more accurate value function approximation and better generalization, the proposed IP-BLR DQN could achieve higher accumulated rewards in fewer interactions with the environment.

REFERENCES

Pierre Alquier and Gérard Biau. Sparse single-index model. *J. Mach. Learn. Res.*, 14(1):243–280, January 2013.

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of gibbs posteriors. *J. Mach. Learn. Res.*, 17(1):8374–8414, January 2016.

K. Azizzadenesheli, E. Brunskill, and A. Anandkumar. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9, Feb 2018.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

Ronen I. Brafman and Moshe Tennenholtz. R-max–a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(2):213, 2003.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016.

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient pac rl with rich observations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 1422–1432. Curran Associates, Inc., 2018.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient RL with rich observations via latent state decoding. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1665–1674, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Mahdi M Fard and Joelle Pineau. Pac-bayesian model selection for reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1624–1632, 2010.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. *CoRR*, abs/1706.10295, 2017.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596, 2018.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. Pac-bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 353–360, New York, NY, USA, 2009. Association for Computing Machinery.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. Pac-bayesian theory meets bayesian inference. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 1884–1892, Red Hook, NY, USA, 2016. Curran Associates Inc.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2094–2100. AAAI Press, 2016.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2):209–232, 2002.

David A. McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

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, 2015.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4026–4034. Curran Associates, Inc., 2016.

Ian Osband, Benjamin Van Roy, Daniel J. Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. Pac-bayes bounds with data dependent priors. *J. Mach. Learn. Res.*, 13(1):3507–3531, December 2012.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *CoRR*, abs/1706.01905, 2017.

David Reeb, Andreas Doerr, Sebastian Gerwinn, and Barbara Rakitsch. Learning gaussian processes by minimizing pac-bayesian generalization bounds. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 3341–3351, Red Hook, NY, USA, 2018. Curran Associates Inc.

Jean-Francis Roy, Mario Marchand, and François Laviolette. A column generation bound minimization approach with pac-bayesian generalization guarantees. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 1241–1249, Cadiz, Spain, 09–11 May 2016. PMLR.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

Huan Wang, Stephan Zheng, Caiming Xiong, and Richard Socher. On the generalization gap in reparameterizable reinforcement learning. In *International Conference on Machine Learning*, pp. 6648–6658, 2019.

A APPENDIX

A.1 PROOF OF (7)

Proof. Assuming Gaussian prior for $\mathbf{w}$ as $\mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}, \beta^{-1}\boldsymbol{I})$, for a batch of training data $\{(s_i, a_i, y_i)\}_{i=1}^{M}$, the likelihood function can be expressed as

$$\begin{aligned} p\left(\boldsymbol{y}|\boldsymbol{w}, \beta\right) &= \prod_{i=1}^{M} \mathcal{N}\left(y_i; \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}_{\boldsymbol{\theta}}(s_i, a_i), \beta^{-1}\right). \\ &= (\frac{2\pi}{\beta})^{-\frac{M}{2}} \exp\left\{-\frac{\beta}{2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})\right\} \end{aligned} \tag{17}$$

Then the posterior $p\left(\boldsymbol{y}|\boldsymbol{w}, \beta\right)$ is also Gaussian distribution. By reformulating (17) as a standard Gaussian distribution, we can get the posterior mean $\boldsymbol{m'}$ and covariance $\boldsymbol{S'}$:

$$\boldsymbol{m'} = \boldsymbol{S'}\left(\boldsymbol{S}^{-1}\boldsymbol{m} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y}\right), \boldsymbol{S'} = \left(\boldsymbol{S}^{-1} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}$$

□

A.2 PROOF OF (9)

Proof.

$$\begin{aligned} \|\mathbb{E}q - q^*\|_\infty =& \|\mathbb{E}q - \mathbb{E}H^*q + \mathbb{E}H^*q - q^*\|_\infty \\ \leq& \|\mathbb{E}q - \mathbb{E}H^*q\|_\infty + \|\mathcal{T}^*\mathbb{E}q - \mathcal{T}^*q^*\|_\infty \\ \leq& \|\mathbb{E}q - \mathbb{E}H^*q\|_\infty + \gamma\|\mathbb{E}q - q^*\|_\infty \end{aligned}$$

Then we have:

$$\|\mathbb{E}q - q^*\|_\infty \leq \frac{1}{1-\gamma}\|\mathbb{E}q - \mathbb{E}H^*q\|_\infty$$

□

A.3 PROOF OF THEOREM 2

Proof. Referring to the change of measure inequality (Roy et al., 2016), we obtain:

$$\lambda\mathbb{E}_{P_q}\varphi(q_{s,a}) \leq D_{\mathrm{KL}}(P_q||J_q) + \ln\psi_J \tag{18}$$

Hence,

$$\lambda\mathbb{E}_{P_q}\|q_{s,a} - q^*_{s,a}\|_\infty \leq \lambda\mathbb{E}_{P_q}\frac{1}{1-\gamma}_{s,a} - \mathbb{E}H^*q_{s,a}\|_2 + D_{\mathrm{KL}}(P_q||J_q) + \ln\psi_J.$$

From Jensen's inequality, we have

$$\|\mathbb{E}_{P_q}q_{s,a} - q^*_{s,a}\|_\infty \leq \mathbb{E}_{P_q}\|q_{s,a} - q^*_{s,a}\|_\infty. \tag{19}$$

Applying Markov's inequality, we have

$$p\left(\psi_J(s,a,y)\right) \leq \alpha) \geq 1 - \frac{\mathbb{E}_{s,a,y}\left[\psi_J\left(s,a,y\right)\right]}{\alpha}. \tag{20}$$

By letting $\delta = \frac{\mathbb{E}_{s,a,y}[\psi_J(s,a,y)]}{\alpha}$ and integrating (19) and (20) into (18), the inequality in (10) holds. □

A.4 ALGORITHMS

Algorithm 1 IP-BLR DQN

Initialize feature extraction parameters $\boldsymbol{\theta}$
Initialize target network's feature extraction parameters $\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$
Initialize IP-BLR parameters $\boldsymbol{m}$, σ, $\boldsymbol{S}$, β
Initialize replay buffer $\mathcal{B}$
for episode $ep = 1, \ldots, N$ **do**
  for step $t = 1, \ldots, T$ **do**
    Draw $\boldsymbol{w}$ from $\mathcal{N}(\boldsymbol{m}, \boldsymbol{S})$
    Select an action $a_t = argmax_a \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi_\theta}(s_t, a)$
    Execute action a_t and receive r_t and s'_t
    Store transition (s_t, a_t, r_t, s'_t) in $\mathcal{B}$
    Sample a random batch of transitions $\{(s_i, a_i, r_i, s'_i)\}_{i=1}^{M}$ from $\mathcal{B}$
    $y_i = \begin{cases} r_i, & \text{for terminal } s'_i \\ \text{caculated from (4)}, & \text{otherwise} \end{cases}$
    Update θ through one step gradient descent on $(y_i - \boldsymbol{m}^{\mathrm{T}} \boldsymbol{\phi_\theta}(s_t, a))^2$
  end for
  if $ep \ mod \ update\_interval == 0$ **then**
    $(\boldsymbol{m}, \boldsymbol{S}) \leftarrow$Algorithm 2$(\boldsymbol{m}, \sigma, \{(\phi_\theta(s_i, a_i), y_i)\}_{i=1}^{M})$
  end if
  update target network: $\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$
end for

Algorithm 2 IP-BLR with adaptive noise parameter

Input: prior parameters $\boldsymbol{m}$, σ and training set $\{(\phi_\theta(s_i, a_i), y_i)\}_{i=1}^{M}$
Output: posterior parameters $\boldsymbol{m}'$, $\boldsymbol{S}'$

for $each \ iteration$ **do**
  Update $(\boldsymbol{m}', \boldsymbol{S}')$ through (7) with $\boldsymbol{S} = \sigma^2 \boldsymbol{I}$
  Update β through (16)
end for