Assumed Density Filtering Q-learning

Heejin Jeong[1*], **Clark Zhang**[1], **George J. Pappas**[1] and **Daniel D. Lee**[2]
[1]University of Pennsylvania, Philadelphia, PA 19104
[2]Cornell Tech, New York, NY 10044
{heejinj, clarkz, pappasg}@seas.upenn.edu, dd146@cornell.edu,

Abstract

While off-policy temporal difference (TD) methods have widely been used in reinforcement learning due to their efficiency and simple implementation, their Bayesian counterparts have not been utilized as frequently. One reason is that the non-linear max operation in the Bellman optimality equation makes it difficult to define conjugate distributions over the value functions. In this paper, we introduce a novel Bayesian approach to off-policy TD methods, called as ADFQ, which updates beliefs on state-action values, Q, through an online Bayesian inference method known as *Assumed Density Filtering*. We formulate an efficient closed-form solution for the value update by approximately estimating analytic parameters of the posterior of the Q-beliefs. Uncertainty measures in the beliefs not only are used in exploration but also provide a natural regularization for the value update considering all next available actions. ADFQ converges to Q-learning as the uncertainty measures of the Q-beliefs decrease and improves common drawbacks of other Bayesian RL algorithms such as computational complexity. We extend ADFQ with a neural network. Our empirical results demonstrate that ADFQ outperforms comparable algorithms on various Atari 2600 games, with drastic improvements in highly stochastic domains or domains with a large action space.

1 Introduction

Bayesian reinforcement learning is a classic reinforcement learning (RL) technique that utilizes Bayesian inference to integrate new experiences with prior information about the problem in a probabilistic distribution. It explicitly quantifies the uncertainty of the learning parameters unlike standard RL approaches in which uncertainty is unaccounted for. Explicit quantification of the uncertainty can help guide policies that consider the exploration-exploitation trade-off by exploring actions with higher uncertainty more often [Osband *et al.*, 2013; Osband *et al.*, 2014]. Moreover, it can also regularize posterior updates by properly accounting for uncertainty.

Motivated by these advantages, a number of algorithms have been proposed in both model-based [Dearden *et al.*, 1999; Duff, 2002; Guez *et al.*, 2012; Poupart *et al.*, 2006] and model-free Bayesian RL [Dearden *et al.*, 1998; Engel *et al.*, 2003; Engel *et al.*, 2005; Geist and Pietquin, 2010; Chowdhary *et al.*, 2014]. However, Bayesian approaches to *off-policy temporal difference (TD) learning* have been less studied compared to alternative methods due to difficulty in handling the max non-linearity in the Bellman optimality equation. Previous studies such as Dearden's Bayesian Q-learning [1998] and Kalman Temporal Difference Q-learning (KTD-Q) [Geist and Pietquin, 2010] suffer from their computational complexity and scalability. Yet off-policy TD methods such as Q-learning [Watkins and Dayan, 1992] have been widely used in standard RL, including extensions integrating neural network function approximations such as Deep Q-Networks (DQN) [Mnih *et al.*, 2013].

In this paper, we introduce a novel approximate Bayesian Q-learning algorithm, denoted as ADFQ, which updates belief distributions of Q (action-value function) and approximates their posteriors using an online Bayesian inference algorithm known as assumed density filtering (ADF). In order to reduce the computational burden of estimating parameters of the approximated posterior, we propose a method to analytically estimate the parameters. Unlike Q-learning, ADFQ executes a non-greedy update by considering all possible actions for the next state and returns a soft-max behavior and regularization determined by the uncertainty measures of the Q-beliefs. This alleviates overoptimism and instability issues from the greedy update of Q-learning which have been discussed in a number of papers [Tsitsiklis, 2002; Harutyunyan *et al.*, 2016; Hasselt, 2010; Hasselt *et al.*, 2016]. We prove the convergence of ADFQ to the optimal Q-values by showing that ADFQ becomes identical to Q-learning as all state and action pairs are visited infinitely often.

ADFQ is computationally efficient and is extended to complex environments with a neural network. There are previous works that implement Bayesian approaches to Deep RL by using uncertainty in the neural network weights and show promising performance in several Atari games [Azizzadenesheli *et al.*, 2018; O'Donoghue *et al.*, 2017; Osband *et al.*, 2016]. However, these approaches only focus on exploration and uncertainty information does not directly applied to updating RL parameters. Our method differs from these approaches as it ex-

*Contact Author

plicitly computes the variances of the Q-beliefs and uses them both for exploration and in the value update. Bellemare et al. [2017] proposed a gradient-based categorical DQN algorithm using a distributional perspective. The value distribution in their work represents the inherent randomness of the agent's interactions with its environment. In contrast, the Q-belief defined in ADFQ is a belief distribution of a learning agent on a certain state-action pair. Therefore, only ϵ-greedy is used in their experiments. We evaluate ADFQ with Thompson sampling (TS) [Thompson, 1933] as well as ϵ-greedy methods in various Atari games and they outperform DQN and Double DQN [Hasselt, 2010]. Particularly, the non-greedy update in ADFQ dramatically improves the performance in domains with a large number of actions and higher stochasticity. Example source code is available online[1].

2 Background

2.1 Assumed Density Filtering

Assumed Density Filtering (ADF) is a general technique for approximating the true posterior with a tractable parametric distribution in Bayesian networks. It has been independently rediscovered for a number of applications and is also known as *moment matching*, *online Bayesian learning*, and *weak marginalization* [Opper, 1999; Boyen and Koller, 1998; Maybeck, 1982]. Suppose that a hidden variable $\mathbf{x}$ follows a tractable parametric distribution $p(\mathbf{x}|\theta_t)$ where θ_t is a set of parameters at time t. In the Bayesian framework, the distribution can be updated after observing some new data (D_t) using Bayes' rule, $\hat{p}(\mathbf{x}|\theta_t, D_t) \propto p(D_t|\mathbf{x}, \theta_t)p(\mathbf{x}|\theta_t)$. In online settings, a Bayesian update is typically performed after a new data point is observed, and the updated posterior is then used as a prior for the following iteration.

When the posterior computed by Bayes' rule does not belong to the original parametric family, it can be approximated by a distribution belonging to the parametric family. In ADF, the posterior is projected onto the closest distribution in the family chosen by minimizing the reverse *Kullback-Leibler* divergence denoted as $KL(\hat{p}||p)$ where $\hat{p}$ is the original posterior distribution and p is a distribution in a parametric family of interest. Thus, for online Bayesian filtering, the parameters for the ADF estimate is given by $\theta_{t+1} = \text{argmin}_\theta \, KL(\hat{p}(\cdot|\theta_t, D_t)||p(\cdot|\theta))$.

2.2 Q-learning

RL problems can be formulated in terms of an MDP described by the tuple, $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability kernel, $R : \mathcal{S} \times \mathcal{A} \to I\!R$ is a reward function, and $\gamma \in [0, 1)$ is a discount factor. The value function is defined as $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|s_0 = s]$ for all $s \in \mathcal{S}$, the expected value of cumulative future rewards starting at a state s and following a policy π thereafter. The state-action value (Q) function is defined as the value for a state-action pair, $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|s_0 = s, a_0 = a]$ for all $s \in \mathcal{S}, a \in \mathcal{A}$. The objective of a learning agent in RL is to find an optimal policy $\pi^* = \text{argmax}_\pi \, V^\pi$. Finding the optimal values, $V^*(\cdot)$ and $Q^*(\cdot, \cdot)$, requires solving the Bellman optimality equation:

$$Q^*(s, a) = \mathbb{E}_{s' \sim P(\cdot|s,a)}[R(s, a) + \gamma \max_{b \in \mathcal{A}} Q^*(s', b)] \quad (1)$$

and $V^*(s) = \max_{a \in \mathcal{A}(s)} Q^*(s, a) \; \forall s \in \mathcal{S}$ where s' is the subsequent state after executing the action a at the state s. *Q-learning* is the most popular off-policy TD learning technique due to its relatively easy implementation and guarantee of convergence to an optimal policy [Watkins and Dayan, 1992; Kaelbling *et al.*, 1996]. At time step t, it updates $Q(s_t, a_t)$ after observing a reward r_t and the next state s_{t+1} (one-step TD learning). The update is based on the *TD error* – a difference between the *TD target*, $r_t + \gamma \max_b Q(s_{t+1}, b)$, and the current estimate on $Q(s_t, a_t)$ with a learning rate $\alpha \in (0, 1]$:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_b Q(s_{t+1}, b) - Q(s_t, a_t) \right)$$

3 Bayesian Q-learning with ADF

3.1 Belief Updates on Q

We define $Q_{s,a}$ as a Gaussian random variable with mean $\mu_{s,a}$ and variance $\sigma^2_{s,a}$ corresponding to the action value function $Q(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. We assume that the random variables for different states and actions are independent and have different means and variances, $Q_{s,a} \sim \mathcal{N}(\mu_{s,a}, \sigma^2_{s,a})$ where $\mu_{s,a} \neq \mu_{s',a'}$ if $s \neq s'$ or $a \neq a' \; \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$. According to the Bellman optimality equation in Eq.1, we can define a random variable for $V(s)$ as $V_s = \max_a Q_{s,a}$. In general, the probability density function for the maximum of Gaussian random variables, $M = \max_{1 \leq k \leq N} X_k$ where $X_k \sim \mathcal{N}(\mu_k, \sigma^2_k)$, is no longer Gaussian (see the appendix).

For one-step Bayesian TD learning, the beliefs on $\mathbf{Q} = \{Q_{s,a}\}_{\forall s \in \mathcal{S}, \forall a \in \mathcal{A}}$ can be updated at time t after observing r_t and s_{t+1} using Bayes' rule. In order to reduce notation, we drop the dependency on t denoting $s_t = s$, $a_t = a$, $s_{t+1} = s'$, $r_t = r$, yielding the causally related 4-tuple $\tau =< s, a, r, s' >$. We use the one-step TD target with a small Gaussian white noise, $r + \gamma V_{s'} + W$ where $W \sim \mathcal{N}(0, \sigma^2_w)$, as the likelihood for $Q_{s,a}$. The noise parameter, σ_w, reflects stochasticity of an MDP. We will first derive the belief updates on Q-values with $\sigma_w = 0$ for simplicity and then extend the result to the general case. The likelihood distribution can be represented as a distribution over $V_{s'}$ as $p(r + \gamma V_{s'}|q, \theta) = p_{V_{s'}}((q - r)/\gamma|s', \theta)$ where q is a value corresponding to $Q_{s,a}$ and θ is a set of mean and variance of $\mathbf{Q}$. From the independence assumptions on $\mathbf{Q}$, the posterior update is reduced to an update for the belief on $Q_{s,a}$:

$$\hat{p}_{Q_{s,a}}(q|\theta, r, s') \propto p_{V_{s'}}\left(\frac{q - r}{\gamma} \middle| s', \theta \right) p_{Q_{s,a}}(q|\theta)$$

The resulting posterior distribution is given as follows where $\phi(\cdot)$ is the standard Gaussian probability density function (PDF) and $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function (CDF) (derivation details in the appendix):

$$\hat{p}_{Q_{s,a}}(q|\theta, r, s') = \frac{1}{Z} \sum_{b \in \mathcal{A}} \frac{c_{\tau,b}}{\bar{\sigma}_{\tau,b}} \phi\left(\frac{q - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}} \right) \prod_{\substack{b' \in \mathcal{A} \\ b' \neq b}} \Phi\left(\frac{q - (r + \gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}} \right) \quad (2)$$

[1] https://github.com/coco66/ADFQ

Z is a normalization constant and

$$c_{\tau,b} = \frac{1}{\sqrt{\sigma^2_{s,a} + \gamma^2\sigma^2_{s',b}}}\phi\left(\frac{(r+\gamma\mu_{s',b}) - \mu_{s,a}}{\sqrt{\sigma^2_{s,a} + \gamma^2\sigma^2_{s',b}}}\right) \quad (3)$$

$$\bar{\mu}_{\tau,b} = \bar{\sigma}^2_{\tau,b}\left(\frac{\mu_{s,a}}{\sigma^2_{s,a}} + \frac{r+\gamma\mu_{s',b}}{\gamma^2\sigma^2_{s',b}}\right) \quad (4)$$

$$\frac{1}{\bar{\sigma}^2_{\tau,b}} = \frac{1}{\sigma^2_{s,a}} + \frac{1}{\gamma^2\sigma^2_{s',b}} \quad (5)$$

Note that all next actions are considered in Eq.2 unlike the conventional Q-learning update which only considers the subsequent action resulting in the maximum Q-value at the next step ($\max_b Q(s',b)$). This can lead to a more stable update rule as updating with only the maximum Q-value has inherent instability. The Bayesian update considers the scenario where the true maximum Q-value may not be the one with the highest estimated mean, and weights each subsequent Q-value accordingly. Each term for action b inside the summation in Eq.2 has three important features. First of all, $\bar{\mu}_{\tau,b}$ is an inverse-variance weighted (IVW) average of the prior mean and the TD target mean. Therefore, the Gaussian PDF part becomes closer to the TD target distribution if it has a lower uncertainty than the prior, and vice versa as compared in the first row of Fig.1. Next, the TD error, $\delta_{\tau,b} = (r+\gamma\mu_{s',b}) - \mu_{s,a}$, is naturally incorporated in the posterior distribution with the form of a Gaussian PDF in the weight $c_{\tau,b}$. Thus, a subsequent action which results in a smaller TD error contributes more to the update. The sensitivity of a weight value is determined by the prior and target uncertainties. An example case is described in the second row of Fig.1 where $\delta_{\tau,0} < \delta_{\tau,1}$ and $\sigma_{s',0} > \sigma_{s',1}$. Finally, the product of Gaussian CDFs provides a soft-max operation. The red curve with dots in the third row of Fig.1 represents $\prod_{b'\neq b}\Phi(q|r+\gamma\mu_{\tau,b'}, \gamma\sigma_{\tau,b'})$ for each b. For a certain q value (x-axis), the term returns a larger value for a larger $\mu_{s',b}$ as seen in the black circles. This results has a similarity with the soft Bellman equation [Ziebart, 2010], but the degree of softness in this case is determined by the uncertainty measures rather than a hyperparameter.

3.2 ADF on Q-Belief Updates

The posterior distribution in Eq.2, however, is no longer Gaussian. In order to continue the online Bayesian update, we approximate the posterior with a Gaussian distribution using ADF. When the parametric family of interest is spherical Gaussian, it is shown that the ADF parameters are obtained by matching moments. Thus, the mean and variance of the approximate posterior are given by those of the true posterior, $\mathbb{E}_{\hat{p}_{Q_{s,a}}}[q]$ and $\mathrm{Var}_{\hat{p}_{Q_{s,a}}}[q]$, respectively. It is fairly easy to derive the mean and variance when $|\mathcal{A}| = 2$. The derivation is presented in the appendix. However, to our knowledge, there is no analytically tractable solution for $|\mathcal{A}| > 2$.

When $\sigma_w > 0$, the expected likelihood is obtained by solving $\int_{\mathbb{R}} p(r+\gamma V_{s'}+w|q,\theta)p_W(w)dw$ which is an integral of a similar form with the posterior in Eq.2. Therefore, a closed-form expression is also not available in general except when $|\mathcal{A}| = 2$ (see the appendix).

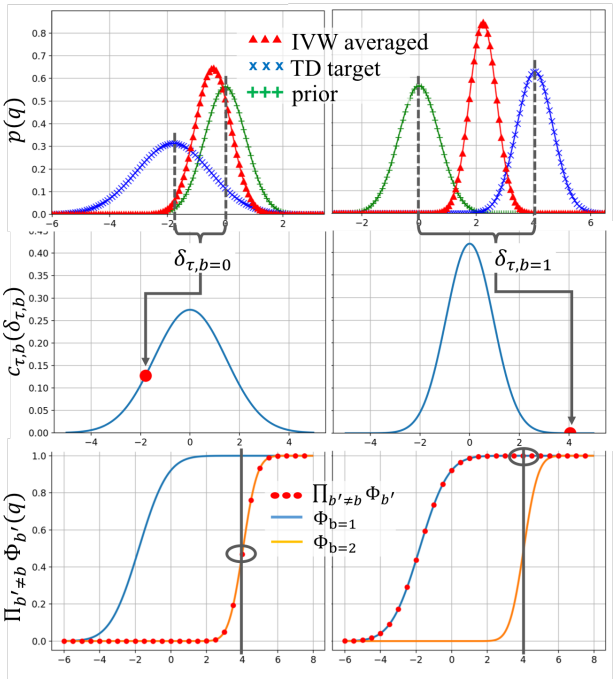

Figure 1: An example of the belief update in Eq.2 when $|\mathcal{A}| = 2, r = 0.0, \gamma = 0.9$ and prior (+ green) has $\mu_{s,a} = 0.0, \sigma^2_{s,a} = 0.5$. Each column corresponds to a subsequent state and action pair, (Left) $b = 0$: $\mu_{s',b} = 1.0, \sigma^2_b = 2.0$, (Right) $b = 1$: $\mu_{s',b} = 4.5, \sigma^2_b = 0.1$.

In the next sections, we prove the convergence of the means to the optimal Q-values for the case $|\mathcal{A}| = 2$ with the exact solutions for the ADF parameters. Then, we show how to derive an analytic approximation for the ADF parameters which becomes exact in the small variance limit.

3.3 Convergence to Optimal Q-values

The convergence theorem of the Q-learning algorithm has previously been proven [Watkins and Dayan, 1992]. We, therefore, show that the online Bayesian update using ADF with the posterior in Eq.2 converges to Q-learning when $|\mathcal{A}| = 2$. We apply an approximation from Lemma 1 in order to prove Theorem 1. Proofs are presented in the appendix.

Lemma 1. *Let X be a random variable following a normal distribution, $\mathcal{N}(\mu,\sigma^2)$. Then we have:*

$$\lim_{\sigma\to 0}\left[\Phi\left(\frac{x-\mu}{\sigma}\right) - \exp\left\{-\frac{1}{2}\left[-\frac{x-\mu}{\sigma}\right]^2_+\right\}\right] = 0 \quad (6)$$

where $[x]_+ = \max(0,x)$ is the ReLU *nonlinearity.*

Theorem 1. *Suppose that the mean and variance of $Q_{s,a}$ $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$ are iteratively updated by the mean and variance of $\hat{p}_{Q_{s,a}}$ after observing r and s' at every step. When $|\mathcal{A}| = 2$, the update rule of the means is equivalent to the Q-learning update if all state-action pairs are visited infinitely often and the variances approach 0. In other words, at the kth update on $\mu_{s,a}$:*

$$\lim_{\substack{k\to\infty\\ \{\sigma\}\to 0}} \mu_{s,a;k+1} = (1-\alpha_{\tau;k})\,\mu_{s,a;k} + \alpha_{\tau;k}\left(r + \gamma \max_{b\in\mathcal{A}} \mu_{s',b;k}\right)$$

where $\alpha_{\tau;k} = \sigma^2_{s,a;k} / \left(\sigma^2_{s,a;k} + \gamma^2 \sigma^2_{s',b^+;k} + \sigma^2_w\right)$ *and* $b^+ = \mathrm{argmax}_{b\in\mathcal{A}}\, \mu_{s',b}$.

Interestingly, α_τ approaches 1 when $\sigma_{s,a}/\sigma_{s',b^+} \to \infty$ and 0 when $\sigma_{s,a}/\sigma_{s',b^+} \to 0$ for $\sigma_w = 0$. Such behavior remains when $\sigma_w > 0$ but α_τ eventually approaches 0 as the number of visits to (s, a) goes to infinity. This not only satisfies the convergence condition of Q-learning but also provides a natural learning rate – the smaller the variance of the TD target (the higher the confidence), the more $Q_{s,a}$ is updated from the target information rather than the current belief. We show empirical evidence that the contraction condition on variance in Theorem 1 holds in the appendix.

4 Analytic ADF Parameter Estimates

When $|\mathcal{A}| > 2$, the update can be solved by numerical approximation of the true posterior mean and variance using a number of samples. However, its computation becomes unwieldy due to the large number of samples needed for accurate estimates. This becomes especially problematic with small variances as the number of visits to corresponding state-action pairs grows. In this section, we show how to accurately estimate the ADF parameters using an analytic approximation. This estimate becomes exact in the small variance limit.

4.1 Analytic Approximation of Posterior

Applying Lemma 1 to the Gaussian CDF terms in Eq.2, the posterior is approximated to the summation over $b \in \mathcal{A}$ of the following term:

$$\frac{c_{\tau,b}}{\sqrt{2\pi}\bar{\sigma}_{\tau,b}} \exp\left\{ -\frac{(q-\bar{\mu}_{\tau,b})^2}{2\bar{\sigma}^2_{\tau,b}} - \sum_{b'\neq b} \frac{[r+\gamma\mu_{s',b'}-q]^2_+}{2\gamma^2\sigma^2_{s',b'}} \right\}$$

Similar to Laplace's method, we approximate each term as a Gaussian distribution by matching the maximum values as well as the curvature at the peak of the distribution. In other words, the maximum of the distribution is modeled locally near its peak by the quadratic concave function:

$$-\frac{(q-\bar{\mu}_{\tau,b})^2}{2\bar{\sigma}^2_{\tau,b}} - \sum_{b'\neq b} \frac{[r+\gamma\mu_{s',b}-q]^2_+}{2\gamma^2\sigma^2_{s',b}} \approx -\frac{(q-\mu^*_{\tau,b})^2}{2{\sigma^*_{\tau,b}}^2}$$

We find $\mu^*_{\tau,b}$ and $\sigma^*_{\tau,b}$ by matching the first and the second derivatives, respectively:

$$\frac{1}{{\sigma^*_{\tau,b}}^2} = \frac{1}{\bar{\sigma}^2_{\tau,b}} + \sum_{b'\neq b} \frac{H\left(r+\gamma\mu_{s',b'}-\mu^*_{\tau,b}\right)}{\gamma^2\sigma^2_{s',b'}} \tag{7}$$

$$\frac{\mu^*_{\tau,b}}{{\sigma^*_{\tau,b}}^2} = \frac{\bar{\mu}_{\tau,b}}{\bar{\sigma}^2_{\tau,b}} + \sum_{b'\neq b} \frac{r+\gamma\mu_{s',b'}}{\gamma^2\sigma^2_{s',b'}} H(r+\gamma\mu_{s',b'}-\mu^*_{\tau,b}) \tag{8}$$

where $H(\cdot)$ is a Heaviside step function. The RHS of the self-consistent piece-wise linear equation Eq.8 is an IVW average mean of the prior, the TD target distribution of b, and other TD target distributions whose means are larger than $\mu^*_{\tau,b}$. The height of the peak is computed as,

$$k^*_{\tau,b} = \frac{c_{\tau,b}\sigma^*_{\tau,b}}{\bar{\sigma}_{\tau,b}} \exp(-Y) \tag{9}$$

Algorithm 1 ADFQ algorithm

```
Initialize randomly μ_{s,a}, σ_{s,a} ∀s ∈ S and ∀a ∈ A
for each episode do
    Initialize s_0
    for each time step t do
        Choose an action, a_t ~ π^{action}(s_t; θ_t)
        Perform the action and observe r_t and s_{t+1}
        for each b ∈ A do
            Compute μ*_{τ,b}, σ*_{τ,b}, k*_{τ,b} using Eq.7-9
        end for
        Update μ_{s_t,a_t} and σ_{s_t,a_t} using Eq.10 and Eq.11
    end for
end for
```

$$Y \equiv -\frac{\left(\mu^*_{\tau,b}-\bar{\mu}_{\tau,b}\right)^2}{2\bar{\sigma}^2_{\tau,b}} - \sum_{b'\neq b} \frac{\left[r+\gamma\mu_{s',b'}-\mu^*_{\tau,b}\right]^2_+}{2\gamma^2\sigma^2_{s',b'}}$$

The final approximated distribution is a Gaussian mixture model with $\mu^*_{\tau,b}, \sigma^*_{\tau,b}, w^*_{\tau,b}$ as mean, variance, and weight, respectively, for all $b \in \mathcal{A}$ where $w^*_{\tau,b} = k^*_{\tau,b} / \sum_{b'} k^*_{\tau,b'}$. Therefore, we update the belief distribution over $Q_{s,a}$ with the mean and variance of the Gaussian mixture model:

$$\mathbb{E}_{\tilde{p}}[q] = \sum_{b\in\mathcal{A}} w^*_{\tau,b}\mu^*_{\tau,b} \tag{10}$$

$$\mathrm{Var}_{\tilde{p}}[q] = \sum_{b\in\mathcal{A}} w^*_{\tau,b}{\sigma^*_{\tau,b}}^2 + \sum_{b\in\mathcal{A}} w^*_{\tau,b}{\mu^*_{\tau,b}}^2 - \left(\mathbb{E}_{\tilde{p}}[q]\right)^2 \tag{11}$$

The final mean is the weighted sum of each individual mean with a weight from $k^*_{\tau,b}$ and the final variance is the weighted sum of each individual variance added to a non-negative term accounting for the dispersion of the means. As shown in Eq.9, the weights are determined by TD errors, variances, and relative distances to larger TD targets. Each weight includes the TD error penalizing term, $c_{\tau,b}$, and also decreases as the number of TD targets larger than $\mu^*_{\tau,b}$ increases. Therefore, the weight provides a softened maximum property over b. The algorithm is summarized in Algorithm 1. Its space complexity is $O(|\mathcal{S}||\mathcal{A}|)$. The computational complexity of each update is $O(|\mathcal{A}|^2)$ which is higher than Q-learning but only by a factor of $|\mathcal{A}|$ and constant in the number of states.

4.2 Approximate Likelihood for Stochastic MDPs

In an asymptotic limit of $\sigma_w/\sigma_{s',b} \to 0$, $\forall b \in \mathcal{A}$ and $|\mathcal{A}| = 2$, the expected likelihood distribution for $\sigma_w > 0$ is similar to $p(r+\gamma V_{s'}|q,\theta)$ but the variance of its Gaussian PDF term is $\gamma^2\sigma^2_{s',b} + \sigma^2_w$ instead of $\gamma^2\sigma^2_{s',b}$ (see the appendix for details). Extending this result to the general case ($|\mathcal{A}| = n$ for $n \in \mathbb{N}$), the posterior distribution for $\sigma_w > 0$ is same with Eq.2 but $\gamma^2\sigma^2_{s',b}$ is replaced by $\gamma^2\sigma^2_{s',b}+\sigma^2_w$ in $c_{\tau,b}$, $\bar{\mu}_{\tau,b}$, and $\bar{\sigma}_{\tau,b}$ (Eq.3-5). Therefore, $\mu^*_{\tau,b}$, $\sigma^*_{\tau,b}$, and $k^*_{\tau,b}$ in the ADFQ algorithm (Table.1) are also changed accordingly. In practice, the non-zero noise parameter is needed when an MDP is stochastic.

4.3 Convergence of ADFQ

Theorem 1 extends to the ADFQ algorithm (Proof in the appendix). The contraction behavior of the variances in the case of Theorem 1 is also empirically observed in ADFQ.

Theorem 2. *The ADFQ update on the mean $\mu_{s,a}$ $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$ for $|\mathcal{A}| = 2$ is equivalent to the Q-learning update if the variances approach 0 and if all state-action pairs are visited infinitely often. In other words, we have :*

$$\lim_{\substack{k\to\infty \\ \{\sigma\}\to 0}} \mu_{s,a;k+1} = (1-\alpha_{\tau;k})\mu_{s,a;k} + \alpha_{\tau;k}(r + \gamma \max_{b\in\mathcal{A}} \mu_{s',b;k})$$

where $\alpha_{\tau;k} = \sigma^2_{s,a;k} / \left(\sigma^2_{s,a;k} + \gamma^2\sigma^2_{s',b^+;k} + \sigma^2_w\right)$ *and* $b^+ = \text{argmax}_{b\in\mathcal{A}}\, \mu_{s',b}$.

As we have observed the behavior of α_τ in Theorem 1, the learning rate α_τ again provides a natural learning rate with the ADFQ update. We can therefore think of Q-learning as a special case of ADFQ.

5 Demonstration in Discrete MDPs

To demonstrate the behavior of the ADFQ update, we look at the simple MDP ($\gamma = 0.9$) in Fig.2 at a specific iteration. An episode starts at s_0 and terminates at either s_2 or s_3. At s_1, each action returns a stochastic reward with $p = 0.2$. The optimal deterministic policy at s_1 is a_1. Suppose an RL learner has already visited (s_1, a_1) 3 times and obtained a reward of $r = 5$ every time. Now it is on the t-th iteration with (s_1, a_1) The plots in Fig.2 show the ADFQ update for Q_{s_0,a_0} at $t+1$ when $r_t = +5$ (left) and $r_t = -5$ (right). When it receives a less expected reward, -5, at t, $\sigma_{s_1,a_1;t}$ is updated to a larger value than the one in the $r_t = +5$ case. Then, the episode is terminated and the next episode starts at $s_{t+1} = s_0, a_{t+1} = a_0$. ADFQ considers both Q_{s_1,a_0} and Q_{s_1,a_1} for updating Q_{s_0,a_0}. Due to the relatively large TD error and variance of Q_{s_1,a_1}, a lower value is assigned to $w^*_{\tau,b=1}$. In this same scenario, Q-learning would update $Q(s_0, a_0)$ only from $Q(s_1, a_0)$ and regulate the update amount with the learning rate which is usually fixed or determined by the number of visits.

In order to show the benefits of the update rule, we examined Q-learning, ADFQ, and a numerical approximation of the mean and variance of Eq.2 (denoted as ADFQ-Numeric) for the convergence to the optimal Q-values in the presented MDP and a similar MDP but with 10 terminating states and 10 actions. Random exploration is used in order to evaluate

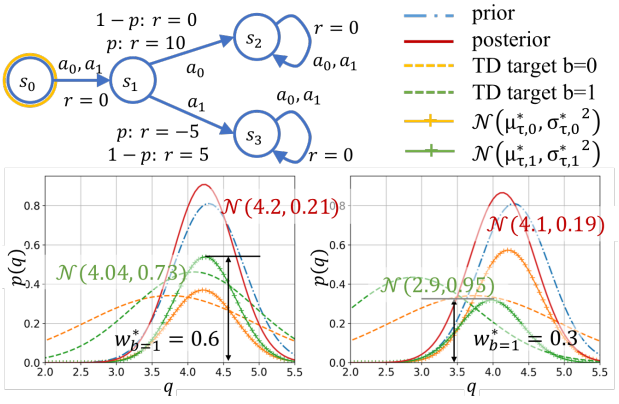

Figure 2: A simple MDP with stochastic rewards and ADFQ update example for $s_{t+1} = s_0, a_{t+1} = a_0$, (left) $r_t = 5$, (right) $r_t = -5$

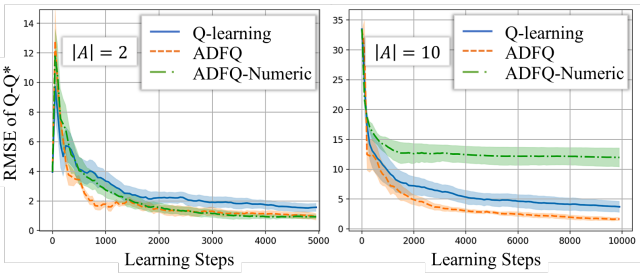

Figure 3: Convergence to Q^* in an MDP with $|\mathcal{A}| = 2$ (left) and an MDP with $|\mathcal{A}| = 10$ (right)

only the update part of each algorithm. During learning, we computed the root mean square error (RMSE) between the estimated Q-values (or means) and Q^*, and plotted the averaged results over 5 trials in Fig.3. As shown, ADFQ converged to the optimal Q-values quicker than Q-learning in both cases and showed more stable performance. ADFQ-Numeric suffers from correctly estimating the parameters when its variances become small as it is previously pointed out, and resulted a poor convergence result in the large MDP.

6 ADFQ with Neural Networks

In this section, we extend our algorithm to complex environments with neural networks similar to Deep Q-Networks (DQN) proposed in [Mnih *et al.*, 2013]. In the Deep ADFQ model with network parameters ξ, the output of the network is mean $\mu(s, a; \xi)$ and variance $\sigma^2(s, a; \xi)$ of each action for a given state s as shown in Fig.4. In practice, we use $-\log(\sigma_{s,a})$ instead of $\sigma^2_{s,a}$ for the output to ensure positive values for the variance. As in DQN, we have a train network (ξ) and a target network (ξ'). Mean and variance for s and s' from the target network are used as inputs into the ADFQ algorithm to compute the desired mean, μ^{ADFQ}, and standard deviation, σ^{ADFQ} for the train network. We used prioritized experience replay [Schaul *et al.*, 2015] and a combined Huber loss functions of mean and variance.

In order to demonstrate the effectiveness of our algorithm, we tested on six Atari games, Enduro ($|\mathcal{A}| = 9$), Boxing ($|\mathcal{A}| = 18$), Pong ($|\mathcal{A}| = 6$), Asterix ($|\mathcal{A}| = 9$), Kung-Fu Master ($|\mathcal{A}| = 14$), and Breakout ($|\mathcal{A}| = 4$), from the OpenAI gym simulator [Brockman *et al.*, 2016]. For baselines, we used DQN and Double DQN with prioritized experience replay

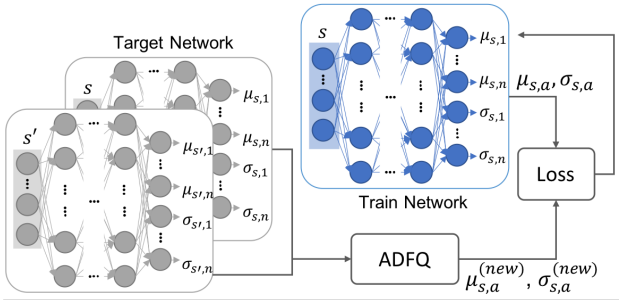

Figure 4: A neural network model for ADFQ

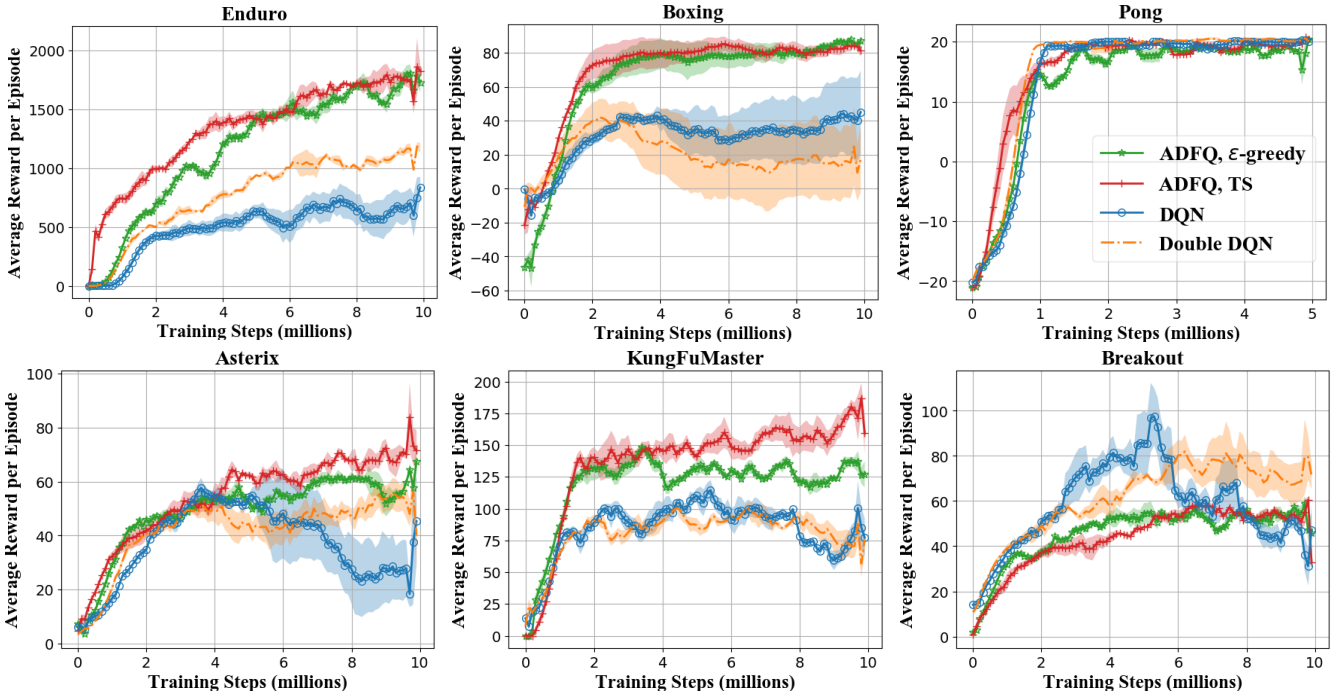

Figure 5: Performance of ADFQ, DQN, and Double DQN during learning smoothed by a moving average with window 6.

implemented in OpenAI baselines[2] with their default hyperparameters for all games. We used ϵ-greedy action policy with ϵ annealed from 1.0 to 0.01 for the baselines as well as ADFQ. In ADFQ, the greedy selection is performed on the mean values instead of Q-values. Additionally, we examined *Thompsing Sampling* (TS) for ADFQ which selects $a_t = \text{argmax}_a\, q_{s_t,a}$ where $q_{s_t,a} \sim p_{Q_{s_t,a}}(\cdot|\theta_t)$.The algorithms were evaluated for $T_H = 10M$ training steps (5M for Pong). Each learning was greedily evaluated at every epoch ($= T_H/100$) for 3 times, and their averaged results are presented in Fig.5. The entire experiment was repeated for 3 random seeds. Rewards were normalized to $\{-1, 0, 1\}$ and different from raw scores of the games. Both ADFQ with TS and with ϵ-greedy notably surpassed DQN and Double DQN in Enduro, Boxing, Asterix, and Kung-Fu Master and showed similar results in Pong. The performance of ADFQ in Breakout is explained as Breakout is the only tested domain where there is no dynamic object interrupting the learning agent. As the demonstration in Sec.5 and the additional experiments in the appendix show, improvements of ADFQ from Q-learning is more significant when an experimental domain has high stochasticity and its action space is large due to the non-greedy update with uncertainty measures. Additionally, ADFQ showed more stable performance in all tested domains overcoming DQN's instability. ADFQ with TS achieved slightly higher performance than the ϵ-greedy method utilizing the uncertainty in exploration.

7 Discussion

We proposed an approach to Bayesian off-policy TD method called ADFQ. ADFQ demonstrated that it could improve some of the issues from the greedy update of Q-learning by showing the quicker convergence to Q^* than Q-learning and surpassing DQN and Double DQN in various Atari games. The presented ADFQ algorithm demonstrates several intriguing results.

First, unlike the conventional Q-learning algorithm, ADFQ incorporates the information of all available actions for the subsequent state in the Q-value update. Each subsequent state-action pair contributes to the update based on its TD target mean and variance as well as its TD error. Particularly, we make use of our uncertainty measures not only in exploration but also in the value update as natural regularization. The advantages of this non-greedy update are noticeable in highly stochastic domains and domains with a large action space in the experiment. Next, we prove that ADFQ converges to Q-learning as the variances decrease and can be seen as a more general form of Q-learning. Last, one of the major drawbacks of Baysian RL approaches is their high computational complexity and poor scalability. ADFQ is computationally efficient and is extended to Deep ADFQ with a neural network.

We would like to highlight the fact that ADFQ is a Bayesian counterpart of Q-learning and is orthogonal to most other advancements made in Deep RL. Deep ADFQ merely changes the loss function and we compare with basic architectures here to provide insight as to how it may improve the performance. ADFQ can be used in conjunction with other extensions and techniques applied to Q-learning and DQN.

[2]https://github.com/openai/baselines

Acknowledgments

This work was supported by funding from the National Institutes of Health U19 program.

References

[Azizzadenesheli *et al.*, 2018] Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. *arXiv preprint arXiv:1802.04412*, 2018.

[Bellemare *et al.*, 2017] Marc G. Bellemare, Will Dabney, , and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[Boyen and Koller, 1998] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*, Berkeley, CA, 1998.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[Chowdhary *et al.*, 2014] Girish Chowdhary, Miao Liu, Robert Grande, Thomas Walsh, Jonathan How, and Lawrence Carin. Off-policy reinforcement learning with gaussian process. *IEEE/CAA Journal of Automatica Sinica*, 1(3):227–238, 2014.

[Dearden *et al.*, 1998] Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian q-learning. In *AAAI/IAAI*, pages 761–768, 1998.

[Dearden *et al.*, 1999] Richard Dearden, Nir Friedman, and David Andre. Model based bayesian exploration. In *Proceedings of the 15th conference on uncertainty in artificial intelligence*, pages 150–159, 1999.

[Duff, 2002] Michael Duff. Optimal learning: Computational procedures for bayes-adaptive markov decision processes. *PhD diss., University of Massachusetts at Amherst*, 2002.

[Engel *et al.*, 2003] Yaakov Engel, Shie Mannor, and Ron Meir. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

[Engel *et al.*, 2005] Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 201–208, 2005.

[Geist and Pietquin, 2010] Matthieu Geist and Olivier Pietquin. Kalman temporal differences. *Journal of Artificial Intelligence Research*, 39:483–532, 2010.

[Guez *et al.*, 2012] Arthur Guez, David Silver, and Peter Dayan. Efficient bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems*, pages 1071–1079, 2012.

[Harutyunyan *et al.*, 2016] Anna Harutyunyan, Marc G. Bellemare, Tom Stepleton, and Rémi Munos. Q (λ) with off-policy corrections. In *International Conference on Algorithmic Learning Theory*, pages 305–320. Springer, 2016.

[Hasselt *et al.*, 2016] Hado V. Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *13th AAAI Conference on Artificial Intelligence*, 2016.

[Hasselt, 2010] Hado V. Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, 2010.

[Kaelbling *et al.*, 1996] Leslie P. Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[Maybeck, 1982] Peter S. Maybeck. Stochastic models, estimation and control. *Academic Press*, chapter 12.7, 1982.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *Advances in Neural Information Processing Systems, Deep Learning Workshop*, 2013.

[O'Donoghue *et al.*, 2017] Brendan O'Donoghue, Ian Osband, Rémi Munos, and Volodymyr Mnih. The uncertainty bellman equation and exploration. *arXiv preprint arXiv:1709.05380*, 2017.

[Opper, 1999] Manfred Opper. A bayesian approach to online learning. *On-Line Learning in Neural Networks*, 1999.

[Osband *et al.*, 2013] Ian Osband, Daniel Russo, and Benjamin V. Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.

[Osband *et al.*, 2014] Ian Osband, Benjamin V. Roy, and Zheng Wen. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0631*, 2014.

[Osband *et al.*, 2016] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin V. Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pages 4026–4034, 2016.

[Poupart *et al.*, 2006] Pascal Poupart, Nikos Vlassis, Jesse Hoey, and Kevin Regan. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[Schaul *et al.*, 2015] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[Thompson, 1933] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.

[Tsitsiklis, 2002] John N. Tsitsiklis. On the convergence of optimistic policy iteration. *Journal of Machine Learning Research*, 3:59–72, 2002.

[Watkins and Dayan, 1992] Christopher J.C.H Watkins and Peter Dayan. Q-learning. In *Machine Learning*, pages 279–292, 1992.

[Ziebart, 2010] Brian D. Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. 2010.