Reinforcement Learning with Bounded Information Loss

Jan Peters, Katharina Mülling, Yevgeny Seldin, Yasemin Altun

Max Planck Institute for Biological Cybernetics
Spemannstr. 38, 72076 Tübingen, Germany

Abstract. Policy search is a successful approach to reinforcement learning. However, policy improvements often result in the loss of information. Hence, it has been marred by premature convergence and implausible solutions. As first suggested in the context of covariant or natural policy gradients, many of these problems may be addressed by constraining the information loss. In this paper, we continue this path of reasoning and suggest two reinforcement learning methods, i.e., a model-based and a model free algorithm that bound the loss in relative entropy while maximizing their return. The resulting methods differ significantly from previous policy gradient approaches and yields an exact update step. It works well on typical reinforcement learning benchmark problems as well as novel evaluations in robotics. We also show a Bayesian bound motivation of this new approach [8].

Keywords: relative entropy, policy search

INTRODUCTION

Policy search is a reinforcement learning approach that attempts to learn improved policies based on information observed in past trials or from observations of another agent's actions [1, 2]. However, policy search, as most reinforcement learning approaches, is usually phrased in an optimal control framework where it directly optimizes the expected return. As there is no notion of the sampled data or a sampling policy in this problem statement, there is no connection between finding an optimal policy and staying close to the observed data. In an online setting, many methods can deal with this problem by staying close to the previous policy (e.g., policy gradient methods allow only small incremental policy updates). Hence, approaches that allow stepping further away from the data are problematic, particularly, off-policy approaches. Directly optimizing a policy will automatically result in a loss of data as an improved policy needs *to forget experience* to avoid the mistakes of the past and to aim on the observed successes. However, choosing an improved policy purely based on its return favors biased solutions that eliminate states in which only bad actions have been tried out. This problem is known as *optimization bias* [3]. Optimization biases may appear in most on- and off-policy reinforcement learning methods due to undersampling (e.g., if we cannot sample all state-actions pairs prescribed by a policy, we will overfit the taken actions), model errors or even the policy update step itself.

Policy updates may often result in a loss of essential information due to the policy improvement step. For example, a policy update that eliminates most exploration by taking the best observed action often yields fast but premature convergence to a suboptimal policy. This problem was observed by Kakade [4] in the context of policy gradients.

There, it can be attributed to the fact that the policy parameter update $\delta\theta$ was maximizing its collinearity $\delta\theta^T\nabla_\theta J$ to the policy gradient while only regularized by fixing the Euclidian length of the parameter update $\delta\theta^T\delta\theta=\varepsilon$ to a step-size ε. Inspired by Amari's work [5] in supervised learning, Kakade [4] concluded that the identity metric of the distance measure was the problem, and that the usage of the Fisher information metric $F(\theta)$ in a constraint $\delta\theta^T F(\theta)\delta\theta=\varepsilon$ leads to a better, more natural gradient. Bagnell and Schneider [6] clarified that the constraint introduced in [4] can be seen as a Taylor expansion of the loss of information or *relative entropy* between the path distributions generated by the original and the updated policy. Bagnell and Schneider's [6] clarification serves as a key insight to this paper.

In this paper, we propose a new method based on this insight, that allows us to estimate new policies given a data distribution both for off-policy or on-policy reinforcement learning. We start from the optimal control problem statement subject to the constraint that the loss in information is bounded by a maximal step size. Note that the methods proposed in [6, 4, 2] used a small fixed step size instead. As we do not work in a parametrized policy gradient framework, we can directly compute a policy update based on all information observed from previous policies or exploratory sampling distributions. All sufficient statistics can be determined by optimizing the dual function that yields the equivalent of a value function of a policy for a data set. We show that the method outperforms the previous policy gradient algorithms [2] as well as SARSA [1].

Background & Notation

We consider the regular reinforcememt learning setting [1, 7] of a stationary Markov decision process (MDP) with n states s and m actions a. When an agent is in state s, he draws an action $a\sim\pi(a|s)$ from a stochastic policy π. Subsequently, the agent transfers from state s to s' with transition probability $p(s'|s,a)=\mathscr{P}^a_{ss'}$, and receives a reward $r(s,a)=\mathscr{R}^a_s\in\Re$. As a result from these state transfers, the agent may converge to a stationary state distribution $\mu^\pi(s)$ for which

$$\forall s' : \textstyle\sum_{s,a}\mu^\pi(s)\pi(a|s)p(s'|s,a)=\mu^\pi(s') \tag{1}$$

holds under mild conditions, see [7]. The goal of the agent is to find a policy π that maximizes the expected return

$$J(\pi)=\textstyle\sum_{s,a}\mu^\pi(s)\pi(a|s)r(s,a), \tag{2}$$

subject to the constraints of Eq.(1) and that both μ^π and π are probability distributions. This problem is called the optimal control problem; however, it does not include any notion of data as discussed in the previous section. In some cases, only some features of the full state s are relevant for the agent. In this case, we only require *stationary feature* vectors

$$\textstyle\sum_{s,a,s'}\mu^\pi(s)\pi(a|s)p(s'|s,a)\phi_{s'}=\sum_{s'}\mu^\pi(s')\phi_{s'}. \tag{3}$$

Note that when using Cartesian unit vectors $u_{s'}$ of length n as features $\phi_{s'}=u_{s'}$, Eq.(3) will become Eq.(1). Using features instead of states relaxes the stationarity condition considerably and often allows a significant speed-up while only resulting in approximate solutions and being highly dependable on the choice of the features. Good features may be RBF features and tile codes, see [1].

BOUNDING RELATIVE ENTROPY LOSS IN REINFORCEMENT LEARNING

We will show both a model-based and a model-free approach here.

Model-based RL with Bounded Relative Entropy Loss

Our method aims at finding the optimal policy that maximizes the expected return based on all observed series of states, actions and rewards. At the same time, we intend to bound the loss of information measured using relative entropy between the observed data distribution $q(s,a)$ and the data distribution $p^{\pi}(s,a) = \mu^{\pi}(s)\pi(a|s)$ generated by the new policy π. Ideally, we want to make use of every sample (s,a,s',r) independently, hence, we express the information loss bound as

$$D(p^{\pi}||q) = \sum\nolimits_{s,a}\mu^{\pi}(s)\pi(a|s)\log\frac{\mu^{\pi}(s)\pi(a|s)}{q(s,a)} \leq \varepsilon, \tag{4}$$

where $D(p^{\pi}||q)$ denotes the Kullback-Leibler divergence, $q(s,a)$ denotes the observed state-action distribution, and ε is our maximal information loss. The problem can hence be stated as follows:

$$\max_{\pi,\mu^{\pi}} J(\pi) = \sum\nolimits_{s,a}\mu^{\pi}(s)\pi(a|s)\mathscr{R}_s^a, \tag{5}$$

$$\text{s.t. } \varepsilon \geq \sum\nolimits_{s,a}\mu^{\pi}(s)\pi(a|s)\log\frac{\mu^{\pi}(s)\pi(a|s)}{q(s,a)}, \tag{6}$$

$$\sum\nolimits_{s'}\mu^{\pi}(s')\phi_{s'} = \sum\nolimits_{s,a,s'}\mu^{\pi}(s)\pi(a|s)\mathscr{P}_{ss'}^a\phi_{s'}, \tag{7}$$

$$1 = \sum\nolimits_{s,a}\mu^{\pi}(s)\pi(a|s). \tag{8}$$

Both μ^{π} and π are probability distributions and the features $\phi_{s'}$ of the MDP are stationary under policy π.

Without the information loss bound constraint in Eq.(6), there is no notion of sampled data and we obtain the stochastic control problem where differentiation of the Langrangian also yields the classical Bellman equation $\phi_s^T\theta = \mathscr{R}_s^a - \lambda + \sum_{s'}\mathscr{P}_{ss'}^a\phi_{s'}^T\theta$. In this equation, $\phi_s^T\theta = V_{\theta}(s)$ is known today as value function while the Langrangian multipliers θ become parameters and λ the average return. While such MDPs may be solved by linear programming [9], approaches that employ sampled experience cannot be derived properly from these equations. The key difference to past optimal control approaches lies in the addition of the constraint in Eq. (6). As discussed in the introduction, natural policy gradient may be derived from a similar problem statement. However, the natural policy gradient requires that ε is small, it can only be properly derived for the path space formulation and it can only be derived from a local, second order Taylor approximation of the problem. Stepping away further from the sampling distribution q will violate these assumptions and, hence, natural policy gradients are inevitably on-policy[1].

[1] Note that there exist sample re-use strategies for larger step away from q using importance sampling, see [1, 10, 11], or off-policy approaches such as Q-Learning (which is known to have problems in approximate, feature-based learning).

The ε can be chosen freely where larger values lead to bigger steps while excessively large values can destroy the policy. Its size depends on the problem as well as on the amount of available samples. As shown in the appendix, we can obtain the optimal policy

$$\pi(a|s) = \frac{q(s,a)\exp\left(\frac{1}{\eta}\delta_\theta(s,a)\right)}{\sum_b q(s,b)\exp\left(\frac{1}{\eta}\delta_\theta(s,b)\right)}, \tag{9}$$

where $\delta_\theta(s,a) = \mathscr{R}_s^a + \sum_{s'} \mathscr{P}_{ss'}^a V_\theta(s') - V_\theta(s)$ denotes the Bellman error. Here, the value function $V_s(\theta) = \theta^T \phi_s$ is determined by minimizing

$$g(\theta,\eta) = \eta \log\left(\sum\nolimits_{s,a} q(s,a)\exp\left(\varepsilon + \frac{1}{\eta}\delta_\theta(s,a)\right)\right), \tag{10}$$

with respect to θ and η. The value function $V_\theta(s) = \phi_s^T \theta$ appears naturally in the derivation of this formulation (see Appendix). The new error function Eq.(10) for obtaining the value functions's parameters θ differs substantially from traditional temporal difference errors, residual gradient errors and monte-carlo rollout fittings [1, 7]. The presented solution is derived for arbitrary stationary features and is therefore sound with function approximation. The derived policy is similar to the Gibbs policy used in policy gradient approaches [7] and in SARSA [1]. In order to turn proposed solution into algorithms, we need to efficiently determine the solution (θ^*, η^*) of the dual function g. Eq. (10) can be rewritten as

$$\min_{\theta,\tilde{\eta}} g(\theta,\tilde{\eta}) = \tilde{\eta}^{-1} \log \sum_{s,a} \exp\left(\log q(s,a) + \varepsilon + \tilde{\eta}\,\delta_\theta(s,a)\right),$$

which is known to be convex [12] as $\delta_\theta(s,a)$ is linear in θ. Given that g is convex and smoothly differentiable, we can determine the optimal solution $g(\theta^*, \eta^*)$ efficiently with any standard optimizer such as Broyden–Fletcher–Goldfarb–Shannon (BFGS) method (denoted in this paper by `fmin_BFGS(`g`,` ∂g`,` $[\theta_0, \eta_0]$`)` with $\partial g = [\partial_\theta g, \partial_\eta g]$).

Model-free RL with Bounded Relative Entropy Loss

Obviously, the algorithm as presented in the previous section would be handicapped by maintaining a high accuracy model of the Markov decision problem $(\mathscr{R}_s^a, \mathscr{P}_{ss'}^a)$. Model estimation would require covering prohibitively many states and actions, and it is hard to obtain an error-free model from data [13, 1]. Furtheremore, in most interesting control problems, we do not intend to visit all states and take all actions — hence, the number of samples N may often be smaller than the number of all state-action pairs mn. Thus, in order to become model-free, we need to rephrase the algorithm. To our great surprise, this part turned out to be simpler than expected. We introduced zero mean state action features ϕ_{sa} in addition to our state feature ϕ_s. Hence, we can change Eqns.(6,7) to

$$\varepsilon \geq \sum_{s,a,s'} \mu^\pi(s)\pi(a|s)\mathscr{P}_{ss'}^a \log \frac{\mu^\pi(s)\pi(a|s)\mathscr{P}_{ss'}^a}{q(s,a,s')},$$

and $\sum_{a',s'} \mu^\pi(s')\pi(a'|s')\left(\phi_{s'a'} + \phi_{s'}\right) = \sum_{s,a,s'} \mu^\pi(s)\pi(a|s)\mathscr{P}_{ss'}^a \phi_{s'}$. Hence, we also obtain the constraint $\sum_{a,s} \mu^\pi(s)\pi(a|s)\phi_{sa} = 0$. It is straightforward to realize that these zero mean features have to realize in an Advantage Function. We obtain a policy

$$\pi(a|s) \approx \frac{\sum_{t=0}^{T} \mathbb{I}_{s_t=s,a_t=a} \exp\left(\frac{1}{\eta}\left(r(s_t,a_t)+V_{s_{t+1}}-(A_{s_t a_t}+V_{s_t})\right)\right)}{\sum_{t=0}^{T} \mathbb{I}_{s_t=s} \exp\left(\frac{1}{\eta}\left(r(s_t,a_t)+V_{s_{t+1}}-(A_{s_t a_t}+V_{s_t})\right)\right)},$$

and have a critic of

$$g \approx -\eta \log\left(\left(\sum_{t=0}^{T} \exp\left(\varepsilon+\frac{1}{\eta}\left(\bar{r}(s_t,a_t)+V_{s_{t+1}}-(A_{s_t a_t}+V_{s_t})\right)\right)\right)^{-1}\right).$$

These methods are clearly model-free.

A Motivation for using Relative Entropy for Regularization

As shown below, relative entropy $D(\mu^{\pi}(s)\pi(a|s)||q(s,a))$ counts possible configurations (or "free energy") of the system and the weighted difference $J(\pi) - \eta D(\mu^{\pi}(s)\pi(a|s)||q(s,a))$ (the "total energy") is bounded. Inspired by [8], we can derive

$$\begin{aligned}
J(\pi) &= \sum_{s,a} \mu^{\pi}(s)\pi(a|s)\mathscr{R}_s^a \\
&= \eta \sum_{s,a} \mu^{\pi}(s)\pi(a|s) \log\left(\frac{\mu^{\pi}(s)\pi(a|s)}{q(s,a)}\left(e^{\frac{1}{\eta}\mathscr{R}_s^a}\right)\frac{q(s,a)}{\mu^{\pi}(s)\pi(a|s)}\right) \\
&= \eta D(\mu^{\pi}(s)\pi(a|s)||q(s,a)) + \eta \sum_{s,a} \mu^{\pi}(s)\pi(a|s) \log\left(\left(e^{\frac{1}{\eta}\mathscr{R}_s^a}\right)\frac{q(s,a)}{\mu^{\pi}(s)\pi(a|s)}\right) \\
&\leq \eta D(\mu^{\pi}(s)\pi(a|s)||q(s,a)) + \eta \log\left(\sum_{s,a} \mu^{\pi}(s)\pi(a|s)\left(e^{\frac{1}{\eta}\mathscr{R}_s^a}\right)\frac{q(s,a)}{\mu^{\pi}(s)\pi(a|s)}\right) \qquad (11) \\
&= \eta D(\mu^{\pi}(s)\pi(a|s)||q(s,a)) + \eta \log\left(\sum_{s,a} q(s,a) e^{\frac{1}{\eta}\mathscr{R}_s^a}\right), \qquad (12)
\end{aligned}$$

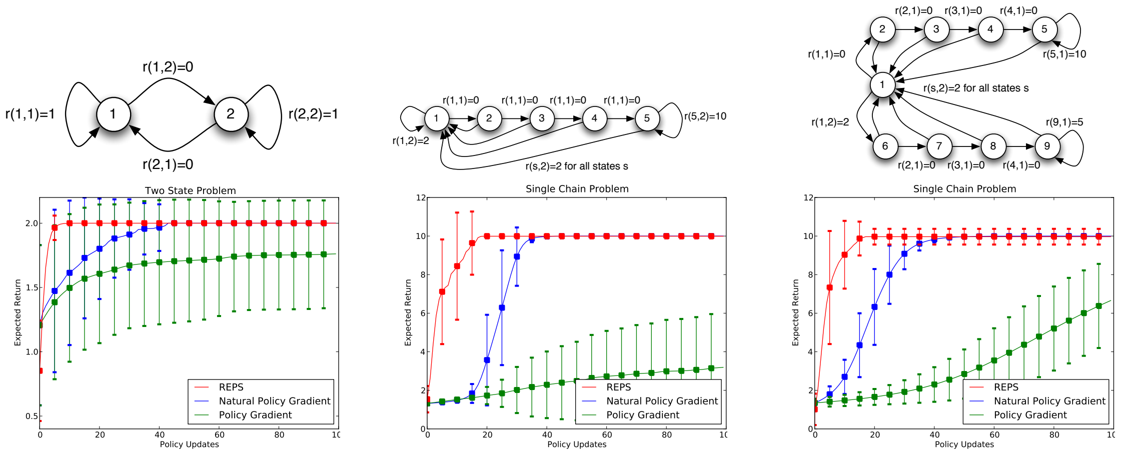
where Eq.(11) is by Jensen's inequality. From the last equation we obtain that:

$$J(\pi) - \eta D(\mu^{\pi}(s)\pi(a|s)||q(s,a)) \leq \eta \log\left(\sum_{s,a} q(s,a) e^{\frac{1}{\eta}\mathscr{R}_s^a}\right).$$

The right hand side of the last equation is fixed, whereas the left hand side depends on $\pi(a|s)$. This motivates maximization of the left hand size (as a function of $\pi(a|s)$) to achieve an equality, which corresponds to proper utilization of degrees of freedom. Note, the stationarity constraint is not considered here.

EXPERIMENTS

In the following section, we test our *Sample-based Policy Iteration with Relative Entropy Policy Search* approach using first several example problems from the literature and, subsequently, on the Mountain Car standard evaluation. Subsequently, we show first steps towards a robot application currently under development.



(a) *Two State Problem* (Bagnell and Schneider 2003)

(b) *Single Chain Problem* (Furmston & Barbar 2010)

(c) *Double Chain Problem* (Furmston & Barbar 2010)

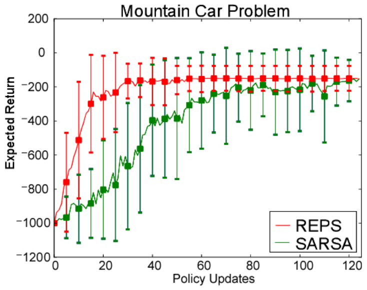
FIGURE 1. Three different methods are compared on three toy examples. The vanilla policy gradients are significantly outperformed due to their slow convergence as already discussed by Bagnell and Schneider (2003) for the Two State Problem. Policy iteration based on Relative Entropy Policy Search (REPS) exhibited the best performance.

Example Problems

We compare our approach both to 'vanilla' policy gradient methods and natural policy gradients [6, 2] using several toy problems. As such, we have chosen (i) the Two-State Problem [6], (ii) the Single Chain Problem [14], and (iii) the Double Chain Problem [14]. In all of these problems, the optimal policy can be observed straightforwardly by a human observer but they pose a major challenge for 'vanilla' policy gradient approaches.We used unit features for all methods. For the two policy gradient approaches a Gibbs policy was employed [7, 6]. On all three problems, we let our policy run until the state distribution has converged to the stationary distribution. For small problems like the presented ones, this usually takes less than 200 steps. Subsequently, we update the policy and resample. We take highly optimized vanilla policy gradients with minimum-variance baselines [2] and the Natural Actor-Critic with unit basis functions as additional function approximation [2]. Instead of a small fixed learning rate, we use an additional momentum term in order to improve the performance. We tuned all meta-parameters of the gradient methods to maximum performance. We start with the same random initial policies for all algorithms and average over 150 learning runs. Nevertheless, similar as in [6, 2], we directly observe that natural gradient outperforms the vanilla policy gradient. Furthermore, we also observe that our REPS policy iteration yields a significantly higher performance. The performance of all three methods for all three problems is shown in Fig. 1 (a-c).

Mountain-Car Problem

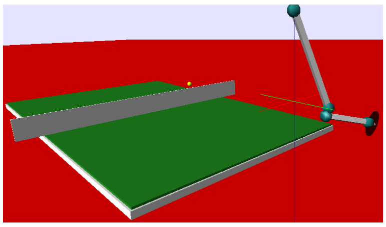
The mountain car problem [1] is a well-known problem in reinforcement learning. We adapt the code from [15] and employ the same tile-coding features for both SARSA and REPS. We implement our algorithm in the same settings and are able to show that REPS policy iteration also outperforms SARSA. While SARSA is superficially quite similar to the presented method, it differs significantly in two parts, i.e., the critic of SARSA converges slower, and the additional multiplication by the previous policy results in a faster pruning of taken bad actions in the REPS approach. As a result, REPS is significantly faster than SARSA as can be observed in Fig. 2.



2: Performance on the mountain-car problem.

Primitive Selection in Robot Table Tennis

Table tennis is a hard benchmark problem for robot learning that includes most difficulties of complex skills. The setup is shown in Fig. 3. A key problem in a skill learning system with multiple motor primitives (e.g., many different forehands, backhands, smashes, etc.) is the selection of task-appropriate primitives triggered by an external stimulus. Here, we have generated a large set of motor primitives that are triggered by a gating network that selects and generalizes among them similar to a mixture of experts. REPS improves the gating network by reinforcement learning where any successful hit results as a reward of +1 and for failures no reward is given. REPS appears to be sensitive to good initial sampling policies. The results vary considerably with initial policy performance. When the system starts with an initial policy that has a success rate of ∼24%, it may quickly converge prematurely yielding a success rate of ∼39%. If provided a better initialization, it can reach success rates of up to ∼59%.



3: Simulated setup for learning robot table tennis.

DISCUSSION & CONCLUSION

In this paper, we have introduced a new reinforcement learning method called Relative Entropy Policy Search. It is derived from a principle as previous covariant policy gradient methods [6], i.e., attaining maximal expected reward while bounding the amount of information loss. Unlike parametric gradient method, it allows an exact policy update and may use data generated while following an unknown policy to generate a new, better policy. It resembles the well-known reinforcement learning method SARSA to an extent; however, it can be shown to outperform it as the critic operates on a different, more sound cost function than traditional temporal difference learning, and as its weighted “soft-max” policy update will promote successful actions faster than the standard soft-max. We have shown that the method performs efficiently when used in a policy iteration setup. REPS is sound with function approximation and can be kernelized straightforwardly which offers interesting possibilities for new algorithms. Application of REPS for reinforcement learning of motor primitive selection for robot table tennis has been successful in simulation.

REFERENCES

1. Sutton, R., and Barto, A. 1998. *Reinforcement Learning*. MIT Press.
2. Peters, J., and Schaal, S. 2008. Natural actor critic. *Neurocomputing* 71(7-9):1180–1190.
3. Mannor, S.; Simester, D.; Sun, P.; and Tsitsiklis, J. N. 2007. Biases and variance in value function estimates. *Management Science* 53(2):308–322.
4. Kakade, S. A. 2002. Natural policy gradient. In *NIPS 14*.
5. Amari, S. 1998. Natural gradient works efficiently in learning. *Neural Computation* 10(2):251–276.
6. Bagnell, J., and Schneider, J. 2003. Covariant policy search. In *International Joint Conference on Artificial Intelligence*.
7. Sutton, R.; McAllester, D.; Singh, S.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NIPS 12*.
8. Amari, S. 2006. On Bayesian bounds. *ICML*.
9. Puterman, M. L. 2005. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley and Sons.
10. Peshkin, L., and Shelton, C. R. 2002. Learning from scarce experience. In *ICML*, 498–505.
11. Hachiya, H.; Akiyama, T.; Sugiyama, M.; Peters, J.; 2008. Adaptive importance sampling with automatic model selection in value function approximation. In *AAAI*, 1351–1356.
12. Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.
13. Deisenroth, M. 2009. *Efficient Reinforcement Learning using Gaussian Processes*. Ph.D. thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany.
14. Furmston, T., and Barber, D. 2010. Variational methods for reinforcement learning. In *AISTATS*.
15. Hernandez, J. 2010. *http://www.dia.fi.upm.es/ jamartin/download.htm*.

DERIVATION OF MODEL-BASED RL WITH BOUNDED RELATIVE ENTROPY LOSS

We denote $p_{sa} = \mu^{\pi}(s)\pi(a|s)$ and $\mu^{\pi}(s) = \sum_a p_{sa}$ for brevity of the derivations, and give the Lagrangian for the program in Eqs.(5-8) by

$$\begin{aligned} L = & \left(\sum_{s,a} p_{sa}\mathscr{R}_s^a\right) + \eta\left(\varepsilon - \sum_{s,a} p_{sa}\log\frac{p_{sa}}{q_{sa}}\right) + \sum_{s'}\theta^T\left(\sum_{s,a} p_{sa}\mathscr{P}_{ss'}^a\phi_{s'} - \sum_{a'} p_{s'a'}\phi_{s'}\right) \qquad (13) \\ & + \lambda\left(1 - \sum_{s,a} p_{sa}\right) = \sum_{s,a} p_{sa}\left(\mathscr{R}_s^a - \eta\log\frac{p_{sa}}{q_{sa}} - \lambda - \theta_s^T\phi_s + \sum_{s'}\mathscr{P}_{ss'}^a\theta_{s'}^T\phi_{s'}\right) + \eta\varepsilon + \lambda, \end{aligned}$$

where η, θ and λ denote the Lagrangian multipliers.We substitute $V_s = \theta^T\phi_s$. We differentiate $\partial_{p_{sa}}L = \mathscr{R}_s^a - \eta\log(p_{sa}/q_{sa}) + \eta - \lambda + \sum_{s'}\mathscr{P}_{ss'}^a V_{s'} - V_s = 0$, and obtain $p_{sa} = q_{sa}\exp(\eta^{-1}\left(\mathscr{R}_s^a + \sum_{s'}\mathscr{P}_{ss'}^a V_{s'} - V_s\right))\exp(1-\lambda/\eta)$. Given that we require $\sum_{s,a} p_{sa} = 1$, it is necessary that $\exp(1-\lambda/\eta)^{-1} = \sum_{s,a} q_{sa}\exp(\eta^{-1}\left(\mathscr{R}_s^a + \sum_{s'}\mathscr{P}_{ss'}^a V_{s'} - V_s\right)$, (hence, λ depends on θ), and we can compute p_{sa} as a direct function of η and θ. We can extract a policy using $\pi(a|s) = p_{sa}/\sum_a p_{sa}$, and hence optain Eq. (9). Reinserting these results into Eq.(13), we obtain the dual function $g(\theta,\eta,\lambda) = -\eta + \eta\varepsilon + \lambda = -\eta\log(\exp(1-\lambda/\eta)\exp(-\varepsilon))$, which can be rewritten as Eq.(10) by inserting $\exp(1-\lambda/\eta)^{-1}$. The derivation of the Model-free RL Algorithm with Bounded Relative Entropy Loss follows analogously.