Bayesian Model Uncertainty in Reinforcement Learning

Elliot Nelson

March 2019

Consider the problem of learning a posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ for the parameters $\boldsymbol{\theta}$ of a neural network, given some dataset $\mathcal{D}$ (which may be growing over time). For instance, we may wish to learn a distribution over a reinforcement learning agent's model of its environment, given the transition and reward data it has observed. The posterior,

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta}), \tag{1}$$

depends on a prior distribution p_0, and on the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ of the data.

The true posterior distribution is generally intractable (especially with a growing dataset $\mathcal{D}$), so we aim to approximate it with a network $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$, parametrized by $\boldsymbol{\phi}$. We will seek to minimize a distance between q_ϕ and the true posterior, which we take to be the KL divergence. It is straightforward to show that

$$-\nabla_{\boldsymbol{\phi}} D_{KL}[p||q_{\boldsymbol{\phi}}] = -\int d\boldsymbol{\theta} q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \left[-\log p(\mathcal{D}|\boldsymbol{\theta}) + \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{\theta})}{p_0(\boldsymbol{\theta})}\right] \tag{2}$$

The first term pushes $\boldsymbol{\phi}$ in a direction such that parameters $\boldsymbol{\theta}$ with a high negative log-likelihood (low likelihood) have a lower probability $q_{\boldsymbol{\phi}}$. That is, $\boldsymbol{\theta}$'s which do not fit the data become less likely. The second term pushes $\boldsymbol{\phi}$ in a direction that keeps $q_{\boldsymbol{\phi}}$ close to the prior p_0. If p_0 is uniform, this is the gradient of the entropy of $q_{\boldsymbol{\phi}}$, and maintains uncertainty in the approximate posterior. We can rewrite the gradient as

$$\nabla_{\boldsymbol{\phi}} D_{KL}[p||q_{\boldsymbol{\phi}}] = \nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi}), \tag{3}$$

where the cost function is (assuming uniform p_0)

$$J(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{\phi}}[-\log p(\mathcal{D}|\boldsymbol{\theta})] - H[q_{\boldsymbol{\phi}}]. \tag{4}$$

Here, $\mathbb{E}_{\boldsymbol{\phi}}$ denotes the expectation value with respect to $q_{\boldsymbol{\phi}}$ and $H[\cdot]$ is the entropy.

We would like to choose an architecture for q which allows for efficient sampling, so that we can estimate the gradient as a sum over sample network parameters $\boldsymbol{\theta}_i$,

$$\int d\boldsymbol{\theta} q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) j(\boldsymbol{\theta}) \to \frac{1}{N_{\text{samples}}} \sum_i j(\boldsymbol{\theta}_i), \tag{5}$$

for the function $j(\boldsymbol{\theta})$ given implicitly in Eq. (4).

In reinforcement learning (RL), the dataset is a list of tuples of observations, actions, successor observations, and rewards: $\mathcal{D} = \{(o_i, a_i, o'_i, r_i\}$. In the case where $\boldsymbol{\theta}$ parametrizes a predictive model of the environment, the likelihood is the output of this predictive model. Given a minibatch $\mathcal{D}_{\text{batch}}$ of transitions – perhaps from a replay buffer – and a model parameters $\boldsymbol{\theta}_{\boldsymbol{\phi}}$ sampled from $q_{\boldsymbol{\phi}}$, we can compute an unbiased estimator of $\nabla_{\boldsymbol{\phi}} J$ which gives the following update to the variational posterior:

$$\Delta\boldsymbol{\phi} \propto \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}_{\boldsymbol{\phi}}) \times \left[\sum_{i}^{|\mathcal{D}_{\text{batch}}|} \log p(o'_i, r_i | o_i, a_i; \boldsymbol{\theta}_{\boldsymbol{\phi}}) - \frac{|\mathcal{D}_{\text{batch}}|}{|\mathcal{D}|} \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}_{\boldsymbol{\phi}}) \right]. \tag{6}$$

Note that as the dataset grows in size and $|\mathcal{D}_{\text{batch}}|/|\mathcal{D}| \to 0$, the uncertainty-maintaining entropy term decays away, which corresponds to annealing a regularization coefficient to zero.[1] Annealing the entropy term to zero with a schedule scaling differently than $1/|\mathcal{D}|$ opens up a space of generalizations of Bayesian learning.

To summarize, Eq. (6) defines a step of stochastic gradient descent with the loss function $D_{KL}[p||q_{\boldsymbol{\phi}}]$ (difference between the approximate and true posterior over parameters of an environment model), given a sample from the approximate posterior and a batch of transition data.

Bayesian learning of a value function in model-free RL could be carried out similarly.

Eq. (6) can also be generalized to the case of a non-uniform prior, for example, to a prior which imposes maximal uncertainty over next-state predictions.

[1]This is because the log-prior contributes relatively less to the true posterior as the log-likelihood becomes a sum over many datapoints – reasonable priors should all be able to converge to the same posterior, given the same evidence.