Dynamical resource allocation in reinforcement learning as inference

Luigi Acerbi

March 2022

Our goal is to cast Dynamical Resource Allocation (DRA; Patel et al. [2020]) as inference following the framework presented in Levine [2018].

1 The inference framework

We augment the state-action space in DRA assuming that for any (s_t, a_t) pair there is an augmented state-action pair $(s_t, \boldsymbol{\alpha}_t)$, with extended action $\boldsymbol{\alpha}_t \equiv (\boldsymbol{\eta}_t(s_t), a_t)$. In this extended action space, the a_t are the usual actions performed by the agent (e.g., move up, down, etc.), whereas the $\boldsymbol{\eta}_t(s_t) \in \mathbb{R}^K$ correspond to the *memory noise vector* added to the Q values, where K is the number of available actions from state s_t. We denote with $\eta_t = \eta_t(s_t, a_t)$ the memory noise associated with action a_t from state s_t. That is, in this formulation we assume that deciding the noise being added to the memories is an action the agent takes at each time step.[1] In the following, to remove clutter we omit the dependence of $\boldsymbol{\eta}_t$ on s_t.

According to standard rules of probability, the action probability is given by

$$p(\boldsymbol{\alpha}_t|s_t) = p(a_t, \boldsymbol{\eta}_t|s_t) = p(a_t|\boldsymbol{\eta}_t, s_t)p(\boldsymbol{\eta}_t|s_t).$$

We recall that $p(\boldsymbol{\eta}_t|s_t)$ and $p(a_t|\boldsymbol{\eta}_t, s_t)$ represent our *priors* over actions. Here we can encode the cost associated with choosing different amounts of noise. For example, the choice taken in DRA is

$$p(\boldsymbol{\eta}_t|s_t, \theta) = \prod_{k=1}^{K} \mathcal{N}\left(\eta_t^{(k)}; 0, \sigma^2_{\text{base}}\right) \tag{1}$$

where $\mathcal{N}\left(x; \mu, \sigma^2\right)$ denotes a normal pdf with mean μ and variance σ^2, and σ^2_{base} in DRA represents the base memory variance. For $p(a_t|\boldsymbol{\eta}_t, s_t)$, instead, we use a default uniform prior (as in [Levine, 2018]).

Finally, we assume that the transition probabilities $p(s_{t+1}|s_t, a_t)$ do not depend on $\boldsymbol{\eta}_t$, and that the reward $r(s_t, a_t)$ depends only on s_t and a_t.

[1] In practice, this choice will become part of the policy.

2 Variational inference

We cast reinforcement learning and DRA as an approximate inference process using variational inference, applying the framework of [Levine, 2018] to our setup.

2.1 Defining the target posterior distribution

Following [Levine, 2018], we derive a posterior distribution over extended trajectories $\tau \equiv (s_1, \boldsymbol{\alpha}_1, \ldots, s_T, \boldsymbol{\alpha}_T)$ where T is the horizon (considered fixed and finite for simplicity). After conditioning on the optimality of the trajectory at each time step, the posterior distribution reads [Levine, 2018]:

$$p(\tau) \propto \left[p(s_1) \prod_{t=1}^{T} p(s_{t+1}|s_t, \boldsymbol{\alpha}_t) p(\boldsymbol{\alpha}_t|s_t) \right] \exp \left(\sum_{t=1}^{T} r(s_t, \boldsymbol{\alpha}_t) \right) \tag{2}$$

where $p(\boldsymbol{\alpha}_t|s_t)$ is the prior over actions (generally omitted in [Levine, 2018], because assumed to be flat there), and we omitted the conditioning over the optimality observations $\mathcal{O}_1, \ldots, \mathcal{O}_T$ to remove clutter.

Under the assumptions of our model, we can rewrite Eq. 2 as:

$$p(\tau) \propto \left[p(s_1) \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t) p(\boldsymbol{\eta}_t|s_t) \right] \exp \left(\sum_{t=1}^{T} r(s_t, a_t) \right). \tag{3}$$

Eq. 3 represents the target unnormalized posterior of our approximate inference process.

2.2 Defining the variational distribution

At this point, $p(\tau)$ is an arbitrary (unnormalized) distribution that we can approximate in many different ways. A natural choice is to use *variational inference*, whereby an unnormalized target density $p(\tau)$ is approximated by another distribution $q_\theta(\tau)$ belonging to a given family of distributions parameterized by θ. Variational inference casts inference into an optimization problem for the variational parameters θ, where the objective function $\mathcal{F}(\theta)$ is given by the *evidence lower bound* (ELBO):

$$\theta^\star = \arg\max_\theta \mathcal{F}(\theta) = \arg\max_\theta \left\{ \mathbb{E}_{q_\theta(\tau)} \left[\log p(\tau) \right] + \mathcal{H} \left[q_\theta(\tau) \right] \right\}, \tag{4}$$

where $\mathcal{H}[q] = -\mathbb{E}_q[\log q]$ is the entropy of $q(\cdot)$. Maximizing the ELBO in Eq. 4 is exactly equivalent to minimizing the Kullback-Leibler divergence between the variational distribution $q_\theta(\tau)$ and the target $p(\tau)$.

In the context of 'reinforcement learning as inference', the variational distribution $q_\theta(\tau)$ corresponds to a stochastic policy parameterized by θ. For example, DRA uses a tabular representation in which the parameters of the policy, $\theta = \left(\bar{Q}(s_t, a_t), \sigma^2(s_t, a_t) \right)$, represent the mean Q-values and their memory variance for each (s_t, a_t) pair [Patel et al., 2020].

As a structured family of distributions for $q_\theta(\tau)$ we choose:

$$q_\theta(\tau) = p(s_1) \prod_{t=1}^{T} \left[p(s_{t+1}|s_t, a_t) q(a_t|\boldsymbol{\eta}_t, s_t, \theta) q(\boldsymbol{\eta}_t|s_t, \theta) \right], \tag{5}$$

where we explain the various terms below:

- $p(s_1)$ and $p(s_{t+1}|s_t, a_t)$ are the true prior and transition probabilities of the task (assumed to be known).
- $q(\boldsymbol{\eta}_t|s_t, \theta) = \prod_{k=1}^{K} \mathcal{N}\left(\eta_t^{(k)}; 0, \sigma^2(s_t, a_t)\right)$ is the memory distribution with allocated memory variance $\sigma^2(s_t, a_t)$ associated with action a_t at state s_t.
- In DRA, the probability of choosing action a_t from state s_t follows a soft Thompson sampling policy, which is a softmax function of the mean Q values and of the memory noise $\boldsymbol{\eta}_t$ [Patel et al., 2020],

  $$q(a_t|\boldsymbol{\eta}_t, s_t, \theta) = \mathrm{softmax}_{a_t}\left(\bar{Q}_t(s_t, a_t) + \eta_t(s_t, a_t); \beta\right), \tag{6}$$

  where $\beta > 0$ is a given inverse temperature hyperparameter.

2.3 The ELBO

The entropy of the variational distribution from Eq. 5 is

$$\begin{aligned}
\mathcal{H}\left[q_\theta(\tau)\right] &= -\,\mathbb{E}_{q_\theta(\tau)}\left[\log p(s_1) + \sum_{t=1}^{T}\{\log p(s_{t+1}|s_t, a_t) + \log q(a_t|\boldsymbol{\eta}_t, s_t, \theta) + \log q(\boldsymbol{\eta}_t|s_t, \theta)\}\right] \\
&= -\sum_{t=1}^{T} \mathbb{E}_{q(s_t|\theta)q(a_t,\boldsymbol{\eta}_t|s_t,\theta)} \log q(a_t|\boldsymbol{\eta}_t, s_t, \theta) - \sum_{t=1}^{T} \mathbb{E}_{q(s_t|\theta)q(\boldsymbol{\eta}_t|s_t,\theta)} \log q(\boldsymbol{\eta}_t|s_t, \theta) + \text{const} \\
&= \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{\eta}_t, s_t|\theta)} \mathcal{H}\left[q(a_t|\boldsymbol{\eta}_t, s_t, \theta)\right] + \sum_{t=1}^{T} \mathbb{E}_{q(s_t|\theta)} \mathcal{H}\left[q(\boldsymbol{\eta}_t|s_t, \theta)\right] + \text{const}
\end{aligned} \tag{7}$$

where in the second step we clumped together all terms constant in θ (not needed for subsequent calculations), and we denoted with $q(s_t|\theta)$ the marginal probability of state s_t according to the policy.

We can now rewrite the optimization objective, the ELBO from Eq. 4, for the chosen variational distribution in Eq. 5 as follows:

$$\begin{aligned}
\mathcal{F}(\theta) &= \mathbb{E}_{q_\theta(\tau)}\left[\log p(\tau)\right] + \mathcal{H}\left[q_\theta(\tau)\right] \\
&= \mathbb{E}_{q_\theta(\tau)}\left[\sum_{t=1}^{T} r(s_t, a_t)\right] - \sum_{t=1}^{T} \mathbb{E}_{q(s_t|\theta)}\left[D_{\mathrm{KL}}\left(q(\boldsymbol{\eta}_t|s_t, \theta) || p(\boldsymbol{\eta}_t|s_t)\right)\right] \\
&\quad + \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{\eta}_t, s_t|\theta)} \mathcal{H}\left[q(a_t|\boldsymbol{\eta}_t, s_t, \theta)\right],
\end{aligned} \tag{8}$$

where we discarded all the terms which do not depend on θ. The KL divergence term in Eq. 8 is the difference between the memory noise distribution for the actions in state s_t (a zero-mean multivariate normal with diagonal covariance with $\sigma^2(s_t, a_t)$ on the diagonal for each action) and the base distribution (a zero-mean multivariate normal with diagonal covariance $\sigma^2_{\text{base}} I$).

Eq. 8 bears a close resemblance to the DRA objective (Eq. 1 in [Patel et al., 2020]), with some cosmetic differences and two substantial differences. The cosmetic differences are:

- The lack of rescaling factor λ in front of the KL divergence, which can be simply obtained with an appropriate rescaling of the rewards.
- The lack of discounting factor γ in the expected reward, which could be included in Eq. 8 with an 'absorbing state' as described in [Levine, 2018].

The *substantial* differences are:

- The KL divergence term in Eq. 8 (the memory cost) is computed *along the trajectory*, whereas in the original DRA formulation the cost of memories is computed once and for all.
- There is an additional entropy term due to the entropy over actions, as in maximum entropy reinforcement learning (see [Levine, 2018]).

3 Recovering the original DRA memory cost

The original DRA formulation of the memory cost term (the term with D_{KL}) is arguably the one closest to the intended meaning of resource allocation, as it is assumed that the agent pays a cost to *encode* (and store) memories with a certain precision, while Eq. 8 is akin to paying the cost *each time a memory is accessed*.

It turns out that we can recover the original DRA memory cost term with a small change to our probabilistic setup. In Section 1, we specified that at each time step the agent allocates the memory noise vector $\boldsymbol{\eta}_t(s_t)$ only for the current state s_t. However, we can have the agent allocate at each time step the memory noise vector $\boldsymbol{\eta}_t$ *for all states* (and actions), *independently of the current state*. The rationale is that the agent is keeping all memories, even when they are not being used. We then impose the same independence structure on the variational distribution.[2]

With this change, the ELBO takes the form:

$$\begin{aligned}\mathcal{F}(\theta) &= \mathbb{E}_{q_\theta(\tau)}\left[\log p(\tau)\right] + \mathcal{H}\left[q_\theta(\tau)\right] \\ &= \mathbb{E}_{q_\theta(\tau)}\left[\sum_{t=1}^{T} r(s_t, a_t)\right] - T \cdot D_{\text{KL}}\left(q(\boldsymbol{\eta}|\theta) || p(\boldsymbol{\eta})\right) \\ &\quad + \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{\eta}_t, s_t|\theta)} \mathcal{H}\left[q(a_t|\boldsymbol{\eta}_t, s_t, \theta)\right],\end{aligned} \tag{9}$$

[2]We need to check details and be careful with the notation, but I think this should work.

which we can rescale by $\frac{\lambda}{T}$, yielding

$$\begin{aligned}\tilde{\mathcal{F}}(\theta) = \mathbb{E}_{q_\theta(\tau)}\left[\sum_{t=1}^{T}\tilde{r}(s_t, a_t)\right] - \lambda \cdot D_{\mathrm{KL}}\left(q(\boldsymbol{\eta}|\theta)||p(\boldsymbol{\eta})\right) \\ + \frac{\lambda}{T}\sum_{t=1}^{T}\mathbb{E}_{q(\boldsymbol{\eta}_t, s_t|\theta)}\mathcal{H}\left[q(a_t|\boldsymbol{\eta}_t, s_t, \theta)\right],\end{aligned} \tag{10}$$

where $\tilde{r}(s_t, a_t) \equiv \frac{\lambda}{T}r(s_t, a_t)$ and $\lambda > 0$ is the cost hyperparameter [Patel et al., 2020]. For a fixed horizon T, Eq. 10 is equivalent to the original DRA objective plus an additional action entropy term.[3]

References

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Nisheet Patel, Luigi Acerbi, and Alexandre Pouget. Dynamic allocation of limited memory resources in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:16948–16960, 2020.

[3]For a variable horizon T, Eq. 10 differs from the original DRA equation, but I think that Eq. 10 might even be better in that it could make sense to pay a cost for memory per unit time.