Robust Bayesian model-based RL

September 19, 2018

1 Proposal

The idea is to combine parametric and nonparametric Bayesian models to model the transition model $P(S' \mid S, A)$, which we write as $P(S)$ hereafter for brevity. That is, we wish to combine a parametric model $\mathcal{M}_p$, ideally derived from domain-specific knowledge, and a nonparametric model $\mathcal{M}_{np}$, which we introduce in order to guard against misspecification.

The prior predictive density for S is therefore given by

$$P(S) = \rho_p P(S \mid \mathcal{M}_p) + (1 - \rho_p) P(S \mid \mathcal{M}_{np}), \tag{1}$$

where ρ_p is the total prior probability of $\mathcal{M}_p$.

1.1 Prior specification

There is not an obvious principled automatic procedure for choosing ρ_p. The first goal, then, is to find a method for eliciting an informative prior ρ_p. This could be done as follows. Suppose we have the following ingredients:

1. $P(S)$, an unconditional prior predictive density. This could, for example, be elicited from a domain expert.
2. $P(S \mid \mathcal{M}_p)$, a prior predictive density under the parametric model. This could again be elicited from a domain expert using established techniques for elicitation of priors for a parametric model, or could come from a model fit to past data.
3. $P(S \mid \mathcal{M}_{np})$, the predictive density under the nonparametric model. Nonparametric priors are more difficult to elicit, so these could be chosen automatically, or come from a nonparametric model fit to past data.

Given these quantities, we could use Equation 1 to find the best-fitting value of ρ_p. Some possible approaches to doing this include:

- Present an expert with data drawn from prior predictive distributions under different values of ρ_p, e.g. $\rho_p \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$, and simply ask which one matches their predictions. This actually only requires ingredients (2) and (3).

- Choose ρ_p to optimize some measure of the goodness of fit of the LHS and RHS of Equation 1. E.g. You could evaluate the LHS and RHS densities at some set of reference values (remember these are implicitly conditional densities), and take ρ_p to minimize the squared error loss between the two.

Since we won't have an expert from which to elicit in this project, we would obtain priors from (possibly fake) past data, and then use the second method described above for choosing ρ_p.

1.2 Models and computation

I'm not sure yet what modelling strategy is best to use, but conditional density regression using Gaussian mixtures and Dirichlet process prior seems like a good place to start, especially since there is a pymc3 example on this: `https://docs.pymc.io/notebooks/dependent_density_regression.html`. For the parametric model, something simple like a linear model would be good.

We can use pymc3 to do the computation.

1.3 Model-based RL

Given a parametric and nonparametric and data from a trajectory, we can update to get a posterior over possible transition dynamics. We can use "policy search" or model-based value iteration to get an estimated optimal policy.

As for exploration, one easy solution is Thompson sampling. However, if time permits, it would be good to explore methods which better approximate Bayes-optimality by tuning the rate of exploration. The Bayes optimal policy is given by

$$\pi = \arg\max_a \mathbb{E}_{\mathcal{T}|H^t} Q^T(\cdot, a)$$

where Q^T is the finite-horizon Q-function with horizon T, and $(\mathcal{T} \mid H^t)$ is our posterior over transition models $\mathcal{T}$ given observations H^t. The challenge is to approximate Bayes-optimal exploration with something computationally feasible, accounting for the time horizon T.

1.3.1 Tuning exploration bonus

Here is a sketch of a potential algorithm (at each time step t):

1. Draw dynamics model from posterior.
2. Roll out for times $t' = t+1, \ldots, T$, using rollout policy $\widetilde{\pi}^{t'} + \alpha(t', \theta)$, where $\widetilde{\pi}^{t'}$ may be an easy-to-compute estimated greedy policy (updated at each step) and $\alpha(t', \theta)$ is the exploration bonus to be tuned.
3. Draw a different dynamics model and repeat, accumulating a set of data.
4. Take a gradient step or similar on θ.

5. Repeat in order to obtain $\widehat{\theta}$.

6. Take action recommended by $\widehat{\pi}^t + \alpha(t, \widehat{\theta})$, where $\widehat{\pi}^t$ is estimated optimal policy.

Note that there is an existing literature on approximately Bayes-optimal RL that I haven't looked into much; see Section 4 of [1] for a brief review.

1.3.2 Model discrimination

Another possibility is approximate the Bayes-optimal policy by only updating the model probability $P(\mathcal{M}_1 \mid H^t)$. If we don't update the within-model posteriors, at each decision-point we can sample once from the posterior, and carry out Monte Carlo Tree Search (or other planning technique) in which we only update $P(\mathcal{M}_1 \mid H^t)$ at each node. Holding the within-model posteriors fixed, this only requires evaluating the likelihoods of the each new observation under the parameters we sampled at the outset.

This approach may be effective in cases where there is low within-model uncertainty (in this case, strong priors for the parametric model), but high between-model uncertainty, i.e. when $P(\mathcal{M}_1 \mid H^t)$ is close to 0.5. In such cases, most of the value of information comes from the value of discriminating between the two models.

It's doubtful that we'd have time, but possible elaborations of this idea include switching between methods which update $P(\mathcal{M}_1 \mid H^t)$ only, and those which fully update the posterior, based on some measure of relative within- and between-model value of information.

References

[1] Arthur Guez, David Silver, and Peter Dayan. Efficient bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems*, pages 1025–1033, 2012.