Weighted model estimation for offline model-based reinforcement learning

Toru Hishinuma, Kei Senda

Motivation

- Model's predictive performance is important for model exploitation in MBRL.
- Covariate shift in MBRL:
  * training data sampled by data-collecting policy.
  * future/test data sampled by improved policy.
- Weighted model estimation can improve predictive performance under covariate shift.

$$\text{weighted model loss} = -\sum w(s,a) \ln P(s'|s,a)$$

How to weight?

- Natural weight

$$w(s,a) = \frac{\text{distribution of real future data}}{\text{distribution of real training data}}$$

- Artificial weight (our idea)

$$w(s,a) = \frac{\text{distribution of simulated future data}}{\text{distribution of real training data}}$$

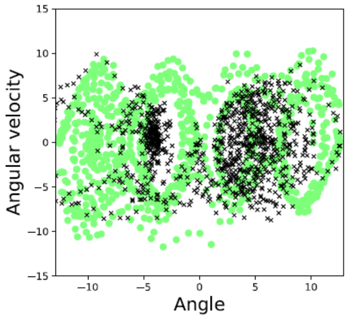
Real future data is inaccessible, but simulated one is accessible. So, our idea is easier-to-use.

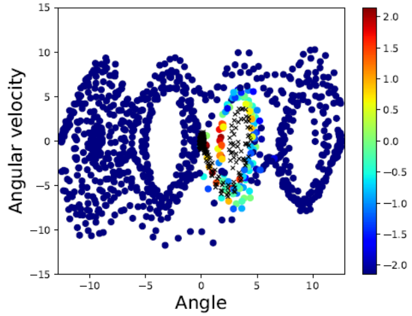
Summary of our algorithm

- EM-style optimization:
  * E-step: weighted model estimation.
  * M-step: policy optimization in simulation.
- Our idea of weighting is justified as evaluating upper bound of policy evaluation error.

Experiments

Pendulum swing-up prediction using small NNs





Color markers show training data and its weight.
Black markers show simulated future data.
Unweighted model estimation (left fig) cannot capture swing-up.
Weighted model estimation (right fig) can do.

D4RL MuJoCo benchmark

dataset	CQL [37]	original MOPO [8]	$\alpha = 0$	$\alpha = 0.2$
HalfCheetah-random	35.4	35.4 ± 2.5	48.7 ± 2.8	49.1 ± 3.2
HalfCheetah-medium	44.4	42.3 ± 1.6	75.7 ± 1.5	73.1 ± 5.2
HalfCheetah-medium-replay	46.2	53.1 ± 2.0	72.1 ± 1.4	65.5 ± 6.4
HalfCheetah-medium-expert	62.4	63.3 ± 38.0	73.9 ± 24.2	85.7 ± 21.6
Hopper-random	10.8	11.7 ± 0.4	30.2 ± 4.4	32.7 ± 0.5
Hopper-medium	86.6	28.0 ± 12.4	100.9 ± 2.7	104.1 ± 1.2
Hopper-medium-replay	48.6	67.5 ± 24.7	97.2 ± 10.9	104.0 ± 3.2
Hopper-medium-expert	111.0	23.7 ± 6.0	109.3 ± 1.1	104.9 ± 10.1
Walker2d-random	7.0	13.6 ± 2.6	16.5 ± 6.6	18.4 ± 7.6
Walker2d-medium	74.5	17.8 ± 19.3	81.7 ± 1.2	60.7 ± 29.0
Walker2d-medium-replay	32.6	39.0 ± 9.6	80.7 ± 3.1	82.7 ± 3.3
Walker2d-medium-expert	98.7	44.6 ± 12.9	59.5 ± 49.4	108.2 ± 0.5

Our algorithm weights with $w(s,a)^{\alpha}$.
If $\alpha = 0$, it is variant of MOPO [Yu+2020].
Ours with $\alpha = 0.2$ improves score of walker2d-medium-expert dataset.

Future issues

- Extension to Bayesian MBRL, by defining posterior distribution based on log-likelihood weighted with artificial weight.
- Combining with loss function in decision-aware model learning approaches.
- Model selection based on loss function weighted with artificial weight.
- Addressing far extrapolation.