Variational inference for reinforcement learning in partially observable markov decision processes

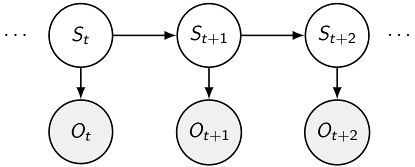
Arseny Kuznetsov

Samsung AI center Moscow

April 26, 2019

Overview

State space model



HMM is a tuple (S, T, Ω, O):

- S – state space
- T – transition function
- Ω – space of observations
- O – observation function

We want to perform filtering, i. e. to estimate

$$p(s_t|o_{1:t}),\ \ p(o_{1:t})$$

SMC procedure: Bootstrap filter (Gordon, 1993)

At $t = 1$

1. Sample N particles $s_1^i \sim q(s_1)$
2. Compute weights $w_1(s_1^i) = \frac{p(s_1^i)p(o_1^i|s_1^i)}{q(s_1^i)}$ $\quad W_1^i = \frac{w_1(s_1^i)}{\sum_i w_1(s_1^i)}$

SMC procedure: Bootstrap filter (Gordon, 1993)

At $t = 1$

1. Sample N particles $s_1^i \sim q(s_1)$
2. Compute weights $w_1(s_1^i) = \frac{p(s_1^i)p(o_1^i|s_1^i)}{q(s_1^i)}$ $\quad W_1^i = \frac{w_1(s_1^i)}{\sum_i w_1(s_1^i)}$

At $t \geq 2$

1. Sample ancestor indices $u_{t-1}^i \sim \mathrm{Cat}(W_{t-1}^1, ..., W_{t-1}^N)$
2. Sample N particles $s_t^i \sim q(s_t|s_{1:t-1}^{u_{t-1}^i})$
3. Compute weights

$$w_t(s_{1:t}^i) = \frac{p(o_t^i|s_t^i)p(s_t^i|s_{t-1}^{u_i})}{q(s_t^i|s_{1:t-1}^{u_{t-1}^i})} \qquad W_t^i = \frac{w_t(s_{1:t}^i)}{\sum_i w_t(s_{1:t}^i)}$$

SMC procedure: Bootstrap filter (Gordon, 1993)

At $t = 1$

1. Sample N particles $s_1^i \sim q(s_1)$
2. Compute weights $w_1(s_1^i) = \frac{p(s_1^i)p(o_1^i|s_1^i)}{q(s_1^i)}$ $\qquad W_1^i = \frac{w_1(s_1^i)}{\sum_i w_1(s_1^i)}$

At $t \geq 2$

1. Sample ancestor indices $u_{t-1}^i \sim \mathrm{Cat}(W_{t-1}^1, ..., W_{t-1}^N)$
2. Sample N particles $s_t^i \sim q(s_t|s_{1:t-1}^{u_{t-1}^i})$
3. Compute weights

$$w_t(s_{1:t}^i) = \frac{p(o_t^i|s_t^i)p(s_t^i|s_{t-1}^{u_i})}{q(s_t^i|s_{1:t-1}^{u_{t-1}^i})} \qquad W_t^i = \frac{w_t(s_{1:t}^i)}{\sum_i w_t(s_{1:t}^i)}$$

Estimate normalization constant and target distribution

$$\widehat{p}(o_{1:t}) = \prod_{\tau=1}^{t} \frac{1}{N} \sum_{i=1}^{N} w_\tau(s_{1:\tau}^i) \qquad \widehat{p}_t(s_{1:t}|o_{1:t}) = \sum_{i=1}^{N} W_t^i \delta(s_{1:t} - s_{1:t}^i)$$

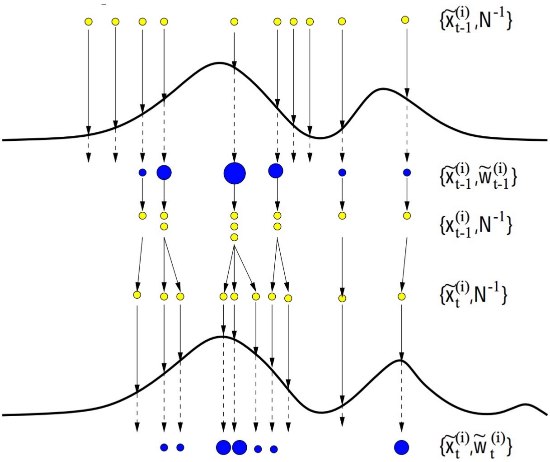


Figure 1. Process of particle filtering

Variational Sequential Monte Carlo

Lets consider learnable model $p_\theta(s_{t+1}|s_t)$, $p_\theta(o_t|s_t)$ and proposal $q_\lambda(s_{t+1}|s_t)$.

$$\tilde{\phi}(s_{1:T}^{1:N}, u_{1:T-1}^{1:N}) = \left[\prod_{i=1}^{N} q(s_1^i)\right] \prod_{t=2}^{T} \prod_{i=1}^{N} \left[W_{t-1}^{u_{t-1}^i} q_\lambda(s_t^i | s_{t-1}^{u_{t-1}^i})\right] \tag{1}$$

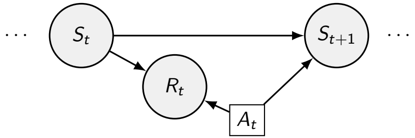
VSMC (Naesseth et al., 2017) shows that

$$\tilde{\mathcal{L}} \triangleq \mathbb{E}_{\tilde{\phi}} \sum_{t=1}^{T} \log \left(\frac{1}{N} \sum_{i=1}^{N} w_t^i \right)$$

is a lower bound on $\log p_\theta(o_{1:T})$.

DVRL: Deep Variational Reinforcement Learning for POMDPs

RL in MDP



- A – action space
- T – $p(s_{t+1}|s_t, a_t)$
- Reward distribution $p(r_t|s_t, a_t)$
- $\gamma \in [0, 1]$ – discount factor

Usual denotations:

$$G_t = \sum_{i=0}^{T-t} \gamma^i R_{t+i}$$

$$V^\pi(s) \triangleq \mathbb{E}_\pi [G_t|S_t = s]$$

$$Q^\pi(s, a) \triangleq \mathbb{E}_\pi [G_t|S_t = s, A_t = a]$$

Advantage Actor Critic (A2C)

$$Q^{t,i}(s_{t+i}, a_{t+i}) \triangleq \left(\sum_{j=0}^{n_s-i-1} \gamma^j r_{t+i+j} \right) + \gamma^{n_s-i} V_\eta^-(s_{t+n_s})$$

Advantage Actor Critic (A2C)

$$Q^{t,i}(s_{t+i}, a_{t+i}) \triangleq \left(\sum_{j=0}^{n_s-i-1} \gamma^j r_{t+i+j} \right) + \gamma^{n_s-i} V_\eta^-(s_{t+n_s})$$

$$A^{t,i}(s_{t+i}, a_{t+i}) \triangleq Q^{t,i}(s_{t+i}, a_{t+i}) - V_\eta(s_{t+i})$$

$$\mathcal{L}_t^A = -\frac{1}{n_e n_s} \sum_{envs}^{n_e} \sum_{i=0}^{n_s-1} \log \pi_\rho(a_{t+i}|s_{t+i}) A^{t,i,-}(s_{t+i}, a_{t+i})$$

Advantage Actor Critic (A2C)

$$Q^{t,i}(s_{t+i}, a_{t+i}) \triangleq \left(\sum_{j=0}^{n_s - i - 1} \gamma^j r_{t+i+j} \right) + \gamma^{n_s - i} V_\eta^-(s_{t+n_s})$$

$$A^{t,i}(s_{t+i}, a_{t+i}) \triangleq Q^{t,i}(s_{t+i}, a_{t+i}) - V_\eta(s_{t+i})$$

$$\mathcal{L}_t^A = -\frac{1}{n_e n_s} \sum_{envs}^{n_e} \sum_{i=0}^{n_s - 1} \log \pi_\rho(a_{t+i} | s_{t+i}) A^{t,i,-}(s_{t+i}, a_{t+i})$$

$$\mathcal{L}_t^V = \frac{1}{n_e n_s} \sum_{envs}^{n_e} \sum_{i=0}^{n_s - 1} A^{t,i}(s_{t+i}, a_{t+i})^2$$

Advantage Actor Critic (A2C)

$$Q^{t,i}(s_{t+i}, a_{t+i}) \triangleq \left(\sum_{j=0}^{n_s-i-1} \gamma^j r_{t+i+j} \right) + \gamma^{n_s-i} V_\eta^-(s_{t+n_s})$$

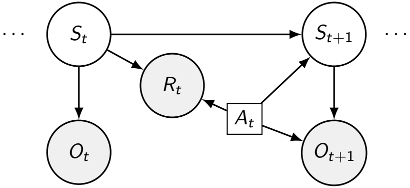
$$A^{t,i}(s_{t+i}, a_{t+i}) \triangleq Q^{t,i}(s_{t+i}, a_{t+i}) - V_\eta(s_{t+i})$$

$$\mathcal{L}_t^A = -\frac{1}{n_e n_s} \sum_{envs}^{n_e} \sum_{i=0}^{n_s-1} \log \pi_\rho(a_{t+i}|s_{t+i}) A^{t,i,-}(s_{t+i}, a_{t+i})$$

$$\mathcal{L}_t^V = \frac{1}{n_e n_s} \sum_{envs}^{n_e} \sum_{i=0}^{n_s-1} A^{t,i}(s_{t+i}, a_{t+i})^2$$

$$\mathcal{L}_t^H = -\frac{1}{n_e n_s} \sum_{envs}^{n_e} \sum_{i=0}^{n_s-1} H(\pi_\rho(\cdot|s_{t+i}))$$

POMDP



- Combination of SSM and MDP.
- We don't have access to underlying states anymore.

Recurrent latent state update

DVRL uses two-part state representation: $s_t^i = (h_t^i, z_t^i)$.
State update basically repeats ont VSMC time-step.

1. $$u_{t-1}^k \sim Cat\left(\frac{w_{t-1}^k}{\sum_{j=1}^K w_{t-1}^j}\right)$$
2. $$z_t^k \sim q_\phi(z_t^k | h_{t-1}^{u_{t-1}^k}, a_{t-1}, o_t)$$
3. $$h_t^k = \psi_\theta^{RNN}(h_{t-1}^{u_{t-1}^k}, z_t^k, a_{t-1}, o_t)$$
4. $$w_t^k = \frac{p_\theta(z_t^k | h_{t-1}^{u_{t-1}^k}, a_{t-1}) p_\theta(o_t | h_{t-1}^{u_{t-1}^k}, z_t^k, a_{t-1})}{q_\phi(z_t^k | h_{t-1}^{u_{t-1}^k}, a_{t-1}, o_t)}$$
5. $$\hat{s}_t = \nu_\theta^{RNN}(\{(h_t^k, z_t^k, w_t^k)\}_{k=1}^K)$$

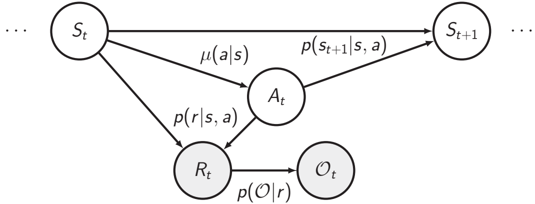
DVRL equals A2C plus PF

$$\mathcal{L}_t^{ELBO} = -\frac{1}{n_e n_s} \sum_{envs}^{n_e} \sum_{i=0}^{n_s-1} \log \left(\frac{1}{K} \sum_{k=1}^{K} w_{t+i}^k \right)$$

$$\mathcal{L}_t^{DVRL} = \mathcal{L}_t^A + \lambda^H \mathcal{L}_t^H + \lambda^V \mathcal{L}_t^V + \lambda^E \mathcal{L}_t^{ELBO}$$

ROBiT: Joint Reward Optimization and Belief Tracking through Approximate Inference

Variational inference in MDP



$\mathcal{O}_t$ – surrogate variable, it indicates "optimality" at moment t.

$$p(\mathcal{O}_t = 1|r_t) \propto \exp(r_t)$$

- Trajectory, i.e. all states and actions

$$\mathbf{z} \triangleq \{\mathbf{s}_{1:T+1}, \mathbf{a}_{1:T}\}$$

- Prior over trajectory with respect to prior policy μ

$$p(\mathbf{z}) = p(s_1) \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\mu(a_t|s_t),$$

- Variational posterior over trajectory with respect to policy π

$$q(\mathbf{z}) = p(s_1) \prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t),$$

- Likelihood of optimality given trajectory

$$p(\mathcal{O}_{1:T} = 1|\mathbf{z}) \;=\; \prod_{t=1}^{T} p(\mathcal{O}_t = 1|s_t, a_t) \;\propto\; \prod_{t=1}^{T} \exp r(s_t, a_t)$$

- Marginal log likelihood of optimality variable

$$\begin{aligned} \log p(\mathcal{O}_{1:T} = 1) \quad &\geq \quad \mathbb{E}_{q(\mathbf{z})} \log \frac{p(\mathcal{O}_{1:T} = 1|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z})} \\ &= \mathbb{E}_{q(\mathbf{z})} \sum_{t=1}^{T} \left[r(s_t, a_t) - \mathrm{KL}\left(\pi(\cdot|s_t) || \mu(\cdot|s_t) \right) \right] \end{aligned}$$

Soft actor critic

In case of uniform prior lower bound becomes Maximum Entropy objective (Soft Actor Critic, Soft Q-learning):

$$\log p(\mathcal{O}_{1:T} = 1) \quad \geq \quad \mathbb{E}_{q(\mathbf{z})} \sum_{t=1}^{T} [r(s_t, a_t) + \mathrm{H}\left(\pi(\cdot|s_t)\right)]$$

Soft actor critic:

- Sample efficient, best scores
- Insensitive to hyperparameter change

Why does it work well?

- Tries to find several way to solve a task, multimodal policies
- Finds conservative, safe solutions

ROBiT lower bounds

Lets denote:

$$x_t = [r_t, \mathcal{O}_t, o_{t+1}]$$

$$g_t = [r_{1:t-1}, a_{1:t-1}, o_{1:t}, \mathcal{O}_{1:t-1}]$$

ROBiT lower bounds

Lets denote:

$$x_t = [r_t, \mathcal{O}_t, o_{t+1}]$$

$$g_t = [r_{1:t-1}, a_{1:t-1}, o_{1:t}, O_{1:t-1}]$$

Construct lower bound on marginal loglikelihood:

$$\begin{aligned} L_t &= \log p(x_t, \mathcal{O}_{t+1:T} | g_t) \\ &= \log \mathbb{E}_{s_t} \left[p(x_t, \mathcal{O}_{t+1:T} | s_t) \right] \\ &\geq \mathbb{E}_{s_t} \left[\log p(x_t, \mathcal{O}_{t+1:T} | s_t) \right] \\ &\geq \mathbb{E}_{s_t} \mathbb{E}_{a_t, s_{t+1} \sim q} \left[\log \frac{p(a_t, x_t, s_{t+1}, \mathcal{O}_{t+1:T} | s_t)}{\pi(a_t | s_t) q(s_{t+1} | s_t, a_t, x_t)} \right] \end{aligned}$$

$$\mathbb{E}_q \left[\underbrace{\log p(\mathcal{O}_{t+1:T} | s_{t+1})}_{\text{value estimate}} + \underbrace{\log p(\mathcal{O}_t | r_t)}_{\text{current reward}} - \underbrace{\mathrm{KL}(\pi | \mu)}_{\text{regularization}} + \underbrace{\log \overline{w}_t}_{\text{VAE on state}} \right]$$

$$\text{where } \overline{w}^t = \frac{p(r_t | s_t, a_t) p(s_{t+1} | s_t, a_t) p(o_{t+1} | s_{t+1}, a_t)}{q(s_{t+1} | s_t, a_t, r_t, o_{t+1})}$$

$$\mathbb{E}_q\left[\underbrace{\log p(\mathcal{O}_{t+1:T}|s_{t+1})}_{\text{value estimate}} + \underbrace{\log p(\mathcal{O}_t|r_t)}_{\text{current reward}} - \underbrace{\mathrm{KL}(\pi|\mu)}_{\text{regularization}} + \underbrace{\log \overline{w}_t}_{\text{VAE on state}}\right]$$

$$\text{where } \overline{w}^t = \frac{p(r_t|s_t, a_t)p(s_{t+1}|s_t, a_t)p(o_{t+1}|s_{t+1}, a_t)}{q(s_{t+1}|s_t, a_t, r_t, o_{t+1})}$$

We approximate future optimality with value function.

$$V_\varphi(s_{t+1}) \approx \log p(\mathcal{O}_{t+1:T}|s_{t+1})$$

$$\pi(a|g_t) = \sum_{k=1}^{K} W_t^{k,N} \pi_\phi(a|s_t^k).$$

$$V(g_t) = \frac{1}{K} \sum_{k=1}^{K} V_\varphi(s_t^k)$$

- Naive particle filter implementation would need to have multiple a_t^k particles.
- But we can have only one action in the environment.

SIVI bound

We utilize a variant of SIVI lower bound. Additional state particles are sampled to estimate action probability.

$$\tilde{s}_t^{0:N} \sim p_\theta(s_t|g_t), \quad a_t \sim \pi(\tilde{s}_t^0) \tag{2}$$

$$\pi(a_t|\tilde{s}_t^{0:N}) = \frac{1}{N+1}\sum_{n=0}^{N}\pi(a_t|\tilde{s}_t^n). \tag{3}$$

$$w_{t+1}^{k,N}(s_t^k, a_t, s_{t+1}^k) = \frac{p_\theta(a_t, x_t, s_{t+1}^k|s_t^k)}{\pi(a_t|\tilde{s}_t^{0:N})q(s_{t+1}^k|s_t^k, a_t, x_t)} \tag{4}$$

$$\mathcal{L}_t^{\mathrm{SI}} = \mathbb{E}_{\tilde{s}_t^{0:N}, s_t^{1:K}, a_t, s_{t+1}^{1:K}} \left[\log \frac{1}{K} \sum_{k=1}^{K} w_{t+1}^{k,N}(s_t^k, a_t, s_{t+1}^k) \right]$$

where the expectation is taken with respect to

$$\prod_{n=0}^{N} p_\theta(\tilde{s}_t^n | g_t) \pi(a_t | \tilde{s}_t^0) \prod_{k=1}^{K} p_\theta(s_t^k | g_t) q_\phi(s_{t+1}^k | s_t^k, a_t, x_t), \qquad (5)$$

$\mathcal{L}_t^{SI}$ is still a lower bound on L_t

$$\mathcal{L}^{\mathrm{ROBiT}} = \mathcal{L}_t^{\mathrm{SI}} + \lambda^V \mathcal{L}_t^{\mathrm{V}} \qquad (6)$$

Recap of the method

1. Theoretically grounded belief agregation
2. Showed that three separate DVRL losses can be viewed as one joint loss
3. From practical viewpoint we added reward likelihood into particle filtering

Experiments

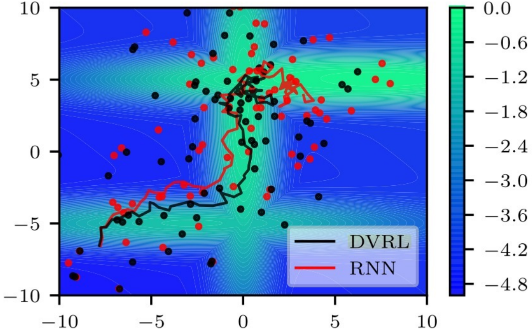


Figure 2. Mountain Hike environment

ROBiT-R – described method.
ROBiT-B – μ is not uniform, but exponential average of π.

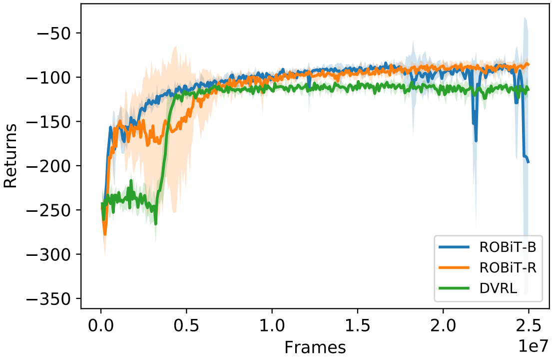


Figure 3. Comparison of returns on Mountain Hike.

The results for two variations of ROBiT method and DVRL for flickering Atari environments.

Env	ROBiT-B	ROBiT-G	DVRL
Asteroids	1627 ± 47	1598 ± 58	1539 ± 73
BeamRider	$\mathbf{3294 \pm 123}$	2039 ± 288	1663 ± 183
Bowling	27.4 ± 1.0	$\mathbf{30.4 \pm 0.1}$	$\mathbf{29.5 \pm 0.2}$
Centipede	4550 ± 190	3848 ± 149	4240 ± 116
ChopperCommand	$\mathbf{9895 \pm 536}$	8313 ± 1086	6602 ± 449
DoubleDunk	-11.2 ± 0.9	-6.3 ± 6.2	$-\mathbf{6.0 \pm 1.1}$
Frostbite	292 ± 12	279 ± 12	297 ± 7
IceHockey	-5.1 ± 0.1	$-\mathbf{4.3 \pm 0.1}$	-4.9 ± 0.2
MsPackman	2512 ± 458	2238 ± 103	2221 ± 199
Pong	14.7 ± 3.8	11.6 ± 2.2	18.2 ± 2.7

Thank you!

References

- Naesseth, Christian A., et al. "Variational Sequential Monte Carlo." arXiv preprint arXiv:1705.11140 (2017).
- Le, Tuan Anh, et al. "Auto-Encoding Sequential Monte Carlo." arXiv preprint arXiv:1705.10306 (2017).
- Igl, Maximilian, et al. "Deep Variational Reinforcement Learning for POMDPs." arXiv preprint arXiv:1806.02426 (2018)
- Levine, Sergey. "Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. " arXiv preprint arXiv:1805.00909 (2018)