Safe Policy Improvement with Baseline Bootstrapping

Romain Laroche, Paul Trichelair, Rémi Tachet des Combes

Microsoft Research

Problem setting

Batch setting

- Fixed dataset, no direct interaction with the environment.
- Access to the behavioural policy, called baseline.
- Objective: improve the baseline with high probability.
- Commonly encountered in real world applications.

Problem setting

Batch setting

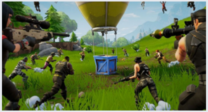
Fixed dataset, no direct interaction with the environment.

Access to the behavioural policy, called baseline.

Objective: improve the baseline with high probability.

Commonly encountered in real world applications.

Distributed systems



Problem setting

Batch setting

Fixed dataset, no direct interaction with the environment.

Access to the behavioural policy, called baseline.

Objective: improve the baseline with high probability.

Commonly encountered in real world applications.

Distributed systems

Long trajectories

Contributions

Novel batch RL algorithm: SPIBB

- SPIBB comes with reliability guarantees in finite MDPs.
- SPIBB is as computationally efficient as classic RL.

Contributions

Novel batch RL algorithm: SPIBB

SPIBB comes with reliability guarantees in nite MDPs.

SPIBB is as computationally ef cient as classic RL.

Finite MDPs benchmark

Extensive benchmark of existing algorithms.

Empirical analysis on random MDPs and baselines.

Contributions

Novel batch RL algorithm: SPIBB

- SPIBB comes with reliability guarantees in nite MDPs.
- SPIBB is as computationally ef cient as classic RL.

Finite MDPs benchmark

- Extensive benchmark of existing algorithms.
- Empirical analysis on random MDPs and baselines.

In nite MDPs benchmark

- Model-free SPIBB for use with function approximation.
- First deep RL algorithm reliable in the batch setting.

Robust Markov Decision Processes

[Iyengar, 2005, Nilim and El Ghaoui, 2005]

- True environment $M = \langle X, A, P, R, \rangle$ is unknown.
- Maximum Likelihood Estimation (MLE) MDP built from counts: $\widehat{M} = \langle X, A, \widehat{P}, \widehat{R}, \rangle$.
- Robust MDP set $(\widehat{M}, e)$: $M \in (\widehat{M}, e)$ with probability at least 1 .
- Error function $e(x, a)$ derived from concentration bounds.

Existing algorithms

[Petrik et al., 2016]: SPI by robust baseline regret minimization

Robust MDPs considers the maxmin of the value over ,

! favors over-conservative policies.

They also consider the maxmin of the value improvement,

! NP-hard problem.

RaMDP hacks the reward to account for uncertainty:

$$\tilde{R}(x;a) \quad \hat{R}(x;a) \quad \sqrt{\frac{\text{adj}}{N_D(x;a)}};$$

! not theoretically grounded.

Existing algorithms

[Petrik et al., 2016]: SPI by robust baseline regret minimization

Robust MDPs considers the maxmin of the value over ,

! favors over-conservative policies.

They also consider the maxmin of the value improvement,

! NP-hard problem.

RaMDP hacks the reward to account for uncertainty:

$$\tilde{R}(x;a) \qquad \hat{R}(x;a) \qquad \frac{\text{adj}}{\sqrt{N_D(x;a)}};$$

! not theoretically grounded.

[Thomas, 2015]: High-Con dence Policy Improvement

HCPI searches for the best regularization hyperparameter to allow safe policy improvement.

Safe Policy Improvement with Baseline Bootstrapping

Safe Policy Improvement with Baseline Bootstrapping (SPIBB)

- Tractable approximate solution to the robust policy improvement formulation.
- SPIBB allows policy update only with suf cient evidence.
- Suf cient evidence = state-action count that exceeds some threshold hyperparameter $N_{\wedge}$.

Safe Policy Improvement with Baseline Bootstrapping

Safe Policy Improvement with Baseline Bootstrapping (SPIBB)

Tractable approximate solution to the robust policy improvement formulation.

SPIBB allows policy update only with suf cient evidence.

Suf cient evidence = state-action count that exceeds some threshold hyperparameter N^ .

SPIBB algorithm

Construction of the bootstrapped set:

$$B = f (x;a) 2 X \quad A \quad ; N_D(x;a) < N \quad g.$$

Optimization over a constrained policy set:

$$_{spibb} = \text{argmax}_{\ 2 \quad b} \ (\ ; \widehat{M});$$

$$_b = f \quad ; \quad \text{s.t.} \quad (ajx) = \quad _b(ajx) \text{ if } (x;a) 2 \ B \ g:$$

SPIBB policy iteration

SPIBB policy iteration

Policy improvement step example

Q-value	Baseline policy	Bootstrapping	SPIBB policy update
$Q^{(i)}_{\widehat{M}}(x, a_1) = 1$	$\pi_b(a_1 \vert x) = 0.1$	$(x, a_1) \in \mathfrak{B}$	
$Q^{(i)}_{\widehat{M}}(x, a_2) = 2$	$\pi_b(a_2 \vert x) = 0.4$	$(x, a_2) \notin \mathfrak{B}$	
$Q^{(i)}_{\widehat{M}}(x, a_3) = 3$	$\pi_b(a_3 \vert x) = 0.3$	$(x, a_3) \notin \mathfrak{B}$	
$Q^{(i)}_{\widehat{M}}(x, a_4) = 4$	$\pi_b(a_4 \vert x) = 0.2$	$(x, a_4) \in \mathfrak{B}$	

SPIBB policy iteration

Policy improvement step example

Q-value	Baseline policy	Bootstrapping	SPIBB policy update
$Q^{(i)}_{bM}(x;a_1) = 1$	$_b(a_1 \mid x) = 0.1$	$(x;a_1) \in B$	$^{(i+1)}(a_1 \mid x) = 0.1$
$Q^{(i)}_{bM}(x;a_2) = 2$	$_b(a_2 \mid x) = 0.4$	$(x;a_2) \notin B$	
$Q^{(i)}_{bM}(x;a_3) = 3$	$_b(a_3 \mid x) = 0.3$	$(x;a_3) \notin B$	
$Q^{(i)}_{bM}(x;a_4) = 4$	$_b(a_4 \mid x) = 0.2$	$(x;a_4) \in B$	$^{(i+1)}(a_4 \mid x) = 0.2$

SPIBB policy iteration

Policy improvement step example

Q-value	Baseline policy	Bootstrapping	SPIBB policy update
$Q^{(i)}_{\pi_b \widehat{M}}(x, a_1) = 1$	$\pi_b(a_1 \mid x) = 0.1$	$(x, a_1) \in \mathcal{B}$	$\pi^{(i+1)}(a_1 \mid x) = 0.1$
$Q^{(i)}_{\pi_b \widehat{M}}(x, a_2) = 2$	$\pi_b(a_2 \mid x) = 0.4$	$(x, a_2) \notin \mathcal{B}$	$\pi^{(i+1)}(a_2 \mid x) = 0.0$
$Q^{(i)}_{\pi_b \widehat{M}}(x, a_3) = 3$	$\pi_b(a_3 \mid x) = 0.3$	$(x, a_3) \notin \mathcal{B}$	$\pi^{(i+1)}(a_3 \mid x) = 0.7$
$Q^{(i)}_{\pi_b \widehat{M}}(x, a_4) = 4$	$\pi_b(a_4 \mid x) = 0.2$	$(x, a_4) \in \mathcal{B}$	$\pi^{(i+1)}(a_4 \mid x) = 0.2$

Theoretical analysis

Theorem (Convergence)

Policy iteration converges to a policy π_{spibb} that is Π_b-optimal in the MLE MDP $\widehat{M}$.

Theoretical analysis

Theorem (Convergence)

Policy iteration converges to a policy $_{spibb}$ that is $_{b}$-optimal in the MLE MDP $\widehat{M}$.

Theorem (Safe policy improvement)

With high probability 1 :

$$(\ _{spibb}; M\ ) \quad (\ _{b}; M\ ) \quad (\ _{spibb}; \widehat{M}) \quad (\ _{b}; \widehat{M}) \quad \frac{4V_{max}}{1} \sqrt{\frac{2}{N_{\wedge}} \log \frac{2|X||A|\ 2^{|X|}}{}}$$

Model-free formulation

SPIBB algorithm

It may be formulated in a model-free manner by setting the targets:

$$y_j^{(i)} = r_j + \gamma \sum_{a' | (x'_j, a') \in \mathfrak{B}} \pi_b(a'|x'_j) Q^{(i)}(x'_j, a') + \gamma \left(\sum_{a' | (x'_j, a') \notin \mathfrak{B}} \pi_b(a'|x'_j) \right) \max_{a' | (x'_j, a') \notin \mathfrak{B}} Q^{(i)}(x'_j, a').$$

Model-free formulation

SPIBB algorithm

It may be formulated in a model-free manner by setting the targets:

$$y_j^{(i)} = r_j + \sum_{a' \mid (x'_j, a') \in \mathfrak{B}} \pi_b(a' \mid x'_j) Q^{(i)}(x'_j, a') + \left(\sum_{a' \mid (x'_j, a') \notin \mathfrak{B}} \pi_b(a' \mid x'_j) \right) \max_{a' \mid (x'_j, a') \notin \mathfrak{B}} Q^{(i)}(x'_j, a').$$

Theorem (Model-free formulation equivalence)

In finite MDPs, the model-free formulation admits a unique fixed point that coincides with the Q-value of π_{spibb}.

25-state stochastic gridworld – mean

25-state stochastic gridworld – 1%-CVaR

Random MDPs, random baseline – 1%-CVaR

Gridworld – RaMDP hyperparameter sensitivity

Gridworld – SPIBB hyperparameter sensitivity

Helicopter domain (continuous task)

Helicopter domain - benchmark (improved results)

Vanilla DQN is off the chart

mean = 0.22,

10%-CVaR = -1 (minimal score).

Conclusion

SPIBB

- Assumes xed dataset, and known behavioural policy.
- Tractable, provably reliable, sample-ef cient algorithm.
- Successfully transferred to DQN architectures.

Conclusion

SPIBB

- Assumes xed dataset, and known behavioural policy.
- Tractable, provably reliable, sample-ef cient algorithm.
- Successfully transferred to DQN architectures.

Follow-up work

- Factored SPIBB [Simão and Spaan, 2019a].
- Structure learning coupled [Simão and Spaan, 2019b].
- Soft SPIBB [Nadjahi et al., 2019].

Conclusion

SPIBB

- Assumes xed dataset, and known behavioural policy.
- Tractable, provably reliable, sample-ef cient algorithm.
- Successfully transferred to DQN architectures.

Follow-up work

- Factored SPIBB [Simão and Spaan, 2019a].
- Structure learning coupled [Simão and Spaan, 2019b].
- Soft SPIBB [Nadjahi et al., 2019].

Still to do

- Improve the pseudo-count/error estimates.
- Investigate an online SPIBB inspired algorithm.

Thanks for your attention (POSTER #101)

Iyengar, G. N. (2005).
Robust dynamic programming.
Mathematics of Operations Research.

Nadjahi, K., Laroche, R., and Tachet des Combes, R. (2019).
Safe policy improvement with soft baseline bootstrapping.
In *Proceedings of the 17th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*.

Nilim, A. and El Ghaoui, L. (2005).
Robust control of markov decision processes with uncertain transition matrices.
Operations Research.

Petrik, M., Ghavamzadeh, M., and Chow, Y. (2016).
Safe policy improvement by minimizing robust baseline regret.
In *Proceedings of the 29th Advances in Neural Information Processing Systems (NIPS)*.

Simão, T. D. and Spaan, M. T. J. (2019a).
Safe policy improvement with baseline bootstrapping in factored environments.
In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Simão, T. D. and Spaan, M. T. J. (2019b).
Structure learning for safe policy improvement.
In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.

Thomas, P. S. (2015).
Safe reinforcement learning.
PhD thesis, Stanford university.