POLITEX: Policy Iteration using Expert Prediction

Nevena Lazic, Yasin-Abbasi Yadkori,
Kush Bhatia, Peter Bartlett,
Gellert Weisz, Csaba Szepesvari

http://proceedings.mlr.press/v97/lazic19a/lazic19a.pdf

Goal

RL algorithm

- Model-free
- Maximize (undiscounted!) total reward during learning

Want

Environment

- Finite action MDP
- Online access

Value function approximator

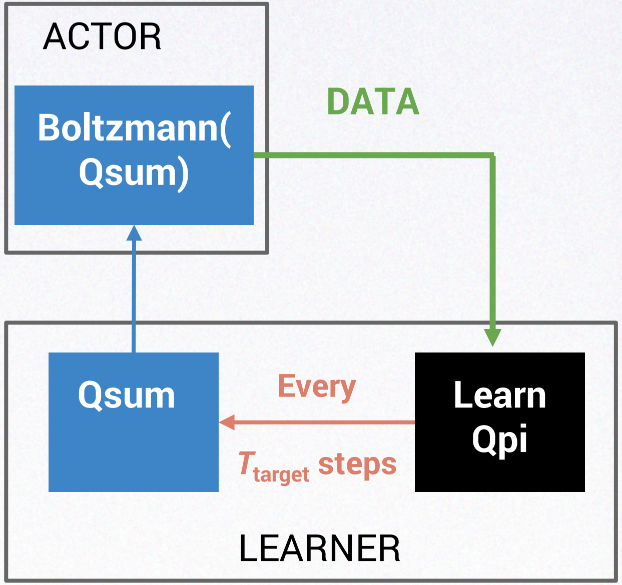
- Q-values approximated well from on-policy data

Have

The Politex algorithm

Loop

1. Set policy to Boltzmann on sum of past Q functions
2. Execute policy for some steps
3. Compute new Q function from collected data



“Meta” theoretical result

Theorem

Assume that for any policy π, after following π for n steps, the black-box produces an action-value function whose error is $\epsilon + 1/\sqrt{n}$ up to some universal constant.

Then the average regret[1] of Politex after T steps is $\epsilon + T^{-\frac{3}{4}}$.

[1]Regret=Loss relative to a reference policy (eg. optimal)

Can the assumption be met?

Learn Qpi ???

- How to build that black-box?
- LSPE (Nedic-Bertsekas, Yu-Bertsekas) for action-value functions, batch-version

- Linear value function approximation:

$$\hat{Q}_\pi = \Psi w_\pi$$

- Solve the "empirical" version of

$$\Psi w = \Pi_\pi(c - \lambda \mathbb{I} + H \Psi w)$$

- **Linear independence**: Columns of $[\Psi\ \mathbb{I}]$ are linearly independent.
- **Feature excitation**: For any π,

$$\lambda_{\min}(\Psi^\top \mathrm{diag}(\nu_\pi)\Psi) \geq \sigma > 0.$$

But why this algorithm???

π

Q^π

regret relative to π_0.

$$\sum_s \mu_0(s)(Q^\pi(s,\pi_0) - Q^\pi(s,\pi))$$

- Policy defines choice of action for each state
- => a separate online learning problem for each state

Stay close to previous policy

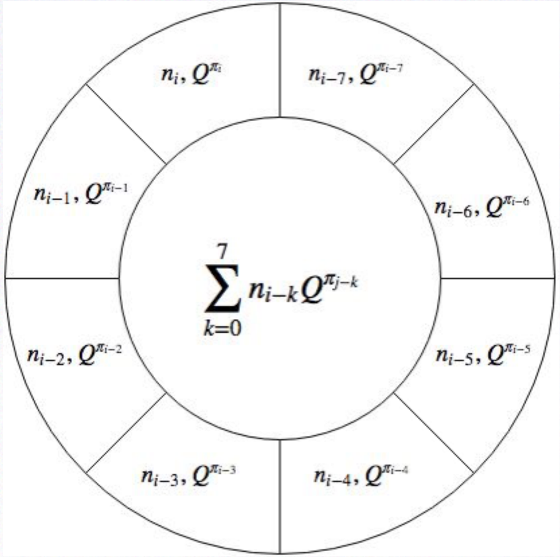
Maximise rewards in hindsight

Regret minimized: $\arg\max_{p \in \Delta_{k-1}} \eta p^\top x - \mathrm{KL}(p, p_{\text{prev}})$

Solution: Boltzmann policy on sum over past x vectors.

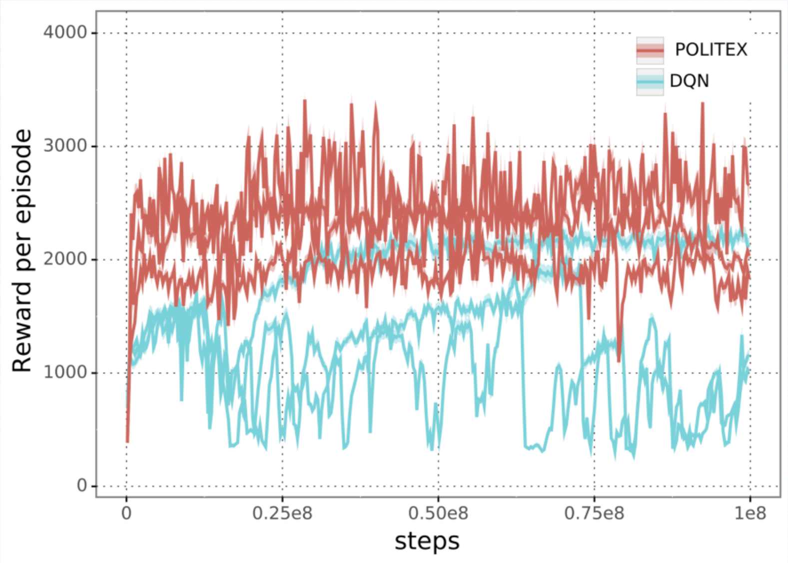
Implementation with neural networks

- Easy to keep average Q with linear function approximation **without** overhead
- Tricky with Neural Networks!
- Approximate solution:
  - Circular buffer of past networks
  - Saved periodically
  - **Constant** factor memory **overhead**
  - Prediction time: **constant** factor **overhead**
  - Training time: **no overhead**



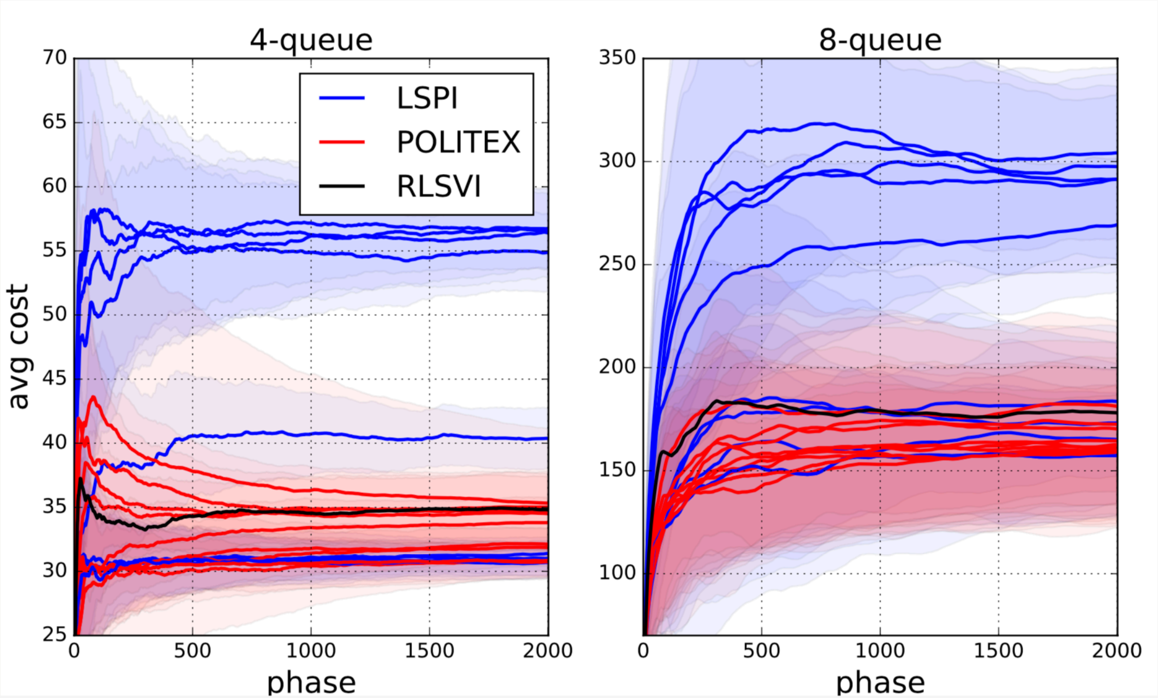
Results on Atari vs DQN

- ACME DQN with TD-weighted replay, few actor steps
- For POLITEX: short uniform replay buffer



Ms Pacman

Results on queuing problems



Relaxing the assumptions

Environment

- Finite-action MDP

Exploring policy

- Excites features/goes "everywhere"

Value function approximator

- Q-values approximated well from **off-policy** data

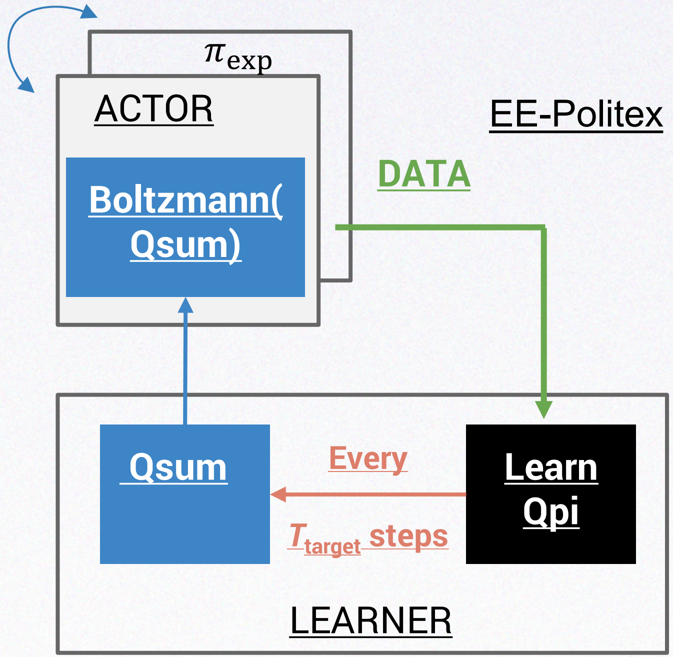
Have

- Easier to satisfy (broadens scope)

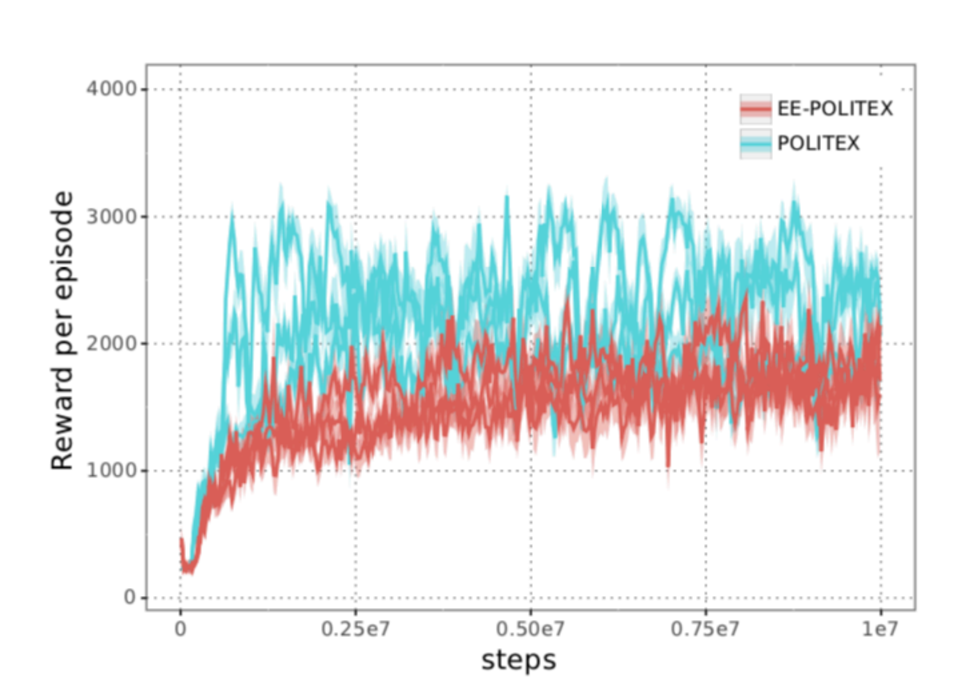
Exploration-enhanced Politex

Loop

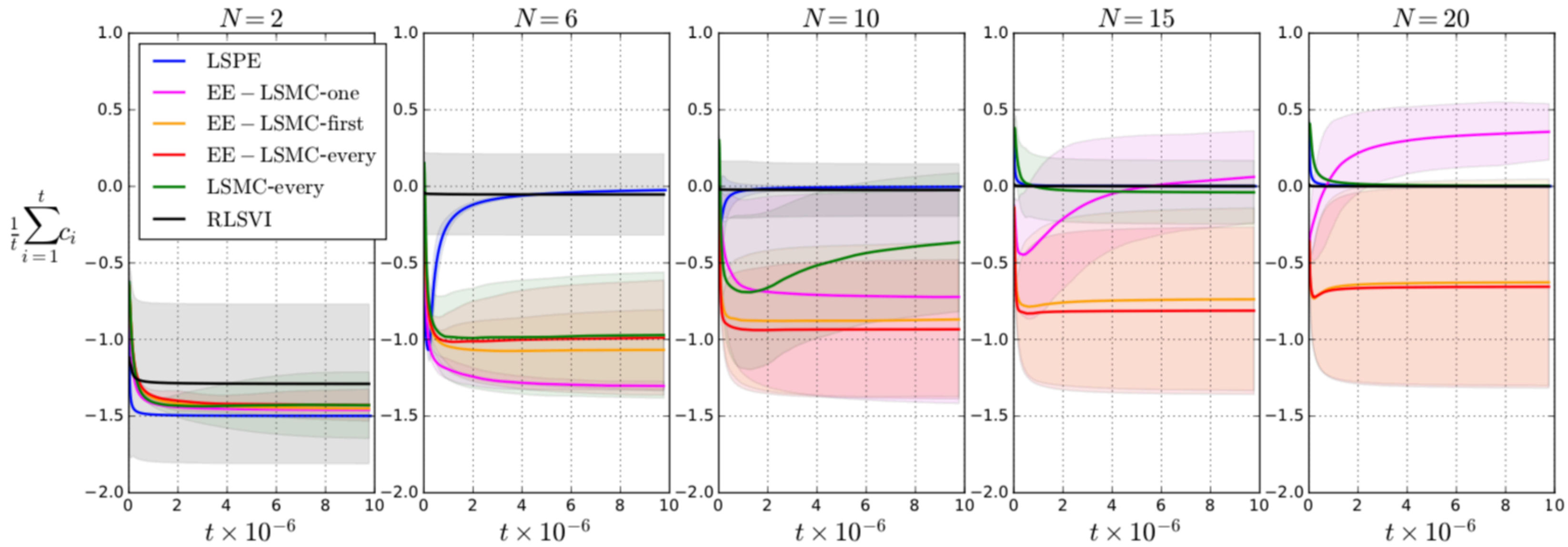
1. Set policy to Boltzmann on sum of past Q functions
2. Iterate:
    1. Execute exploring policy for some steps
    2. Execute current policy for some steps
3. Compute new Q function from collected data



Experimental results: Ms Pac-Man

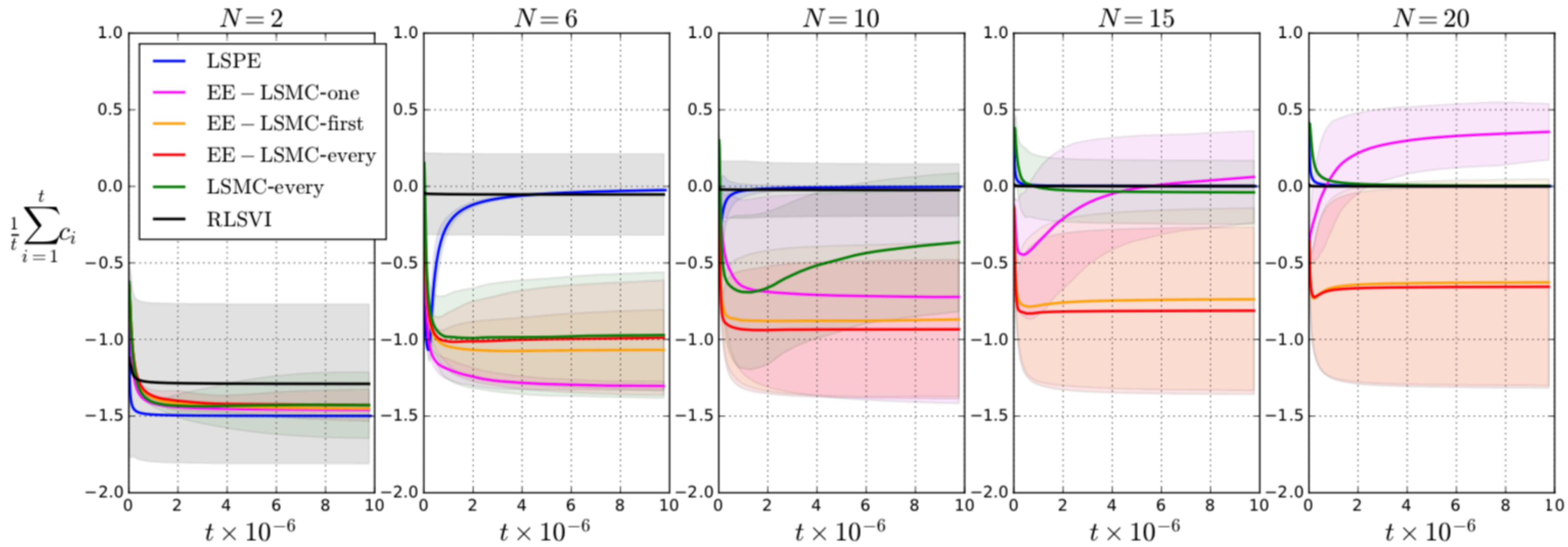


Experimental results: DeepSea

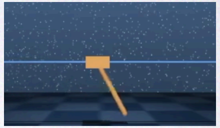


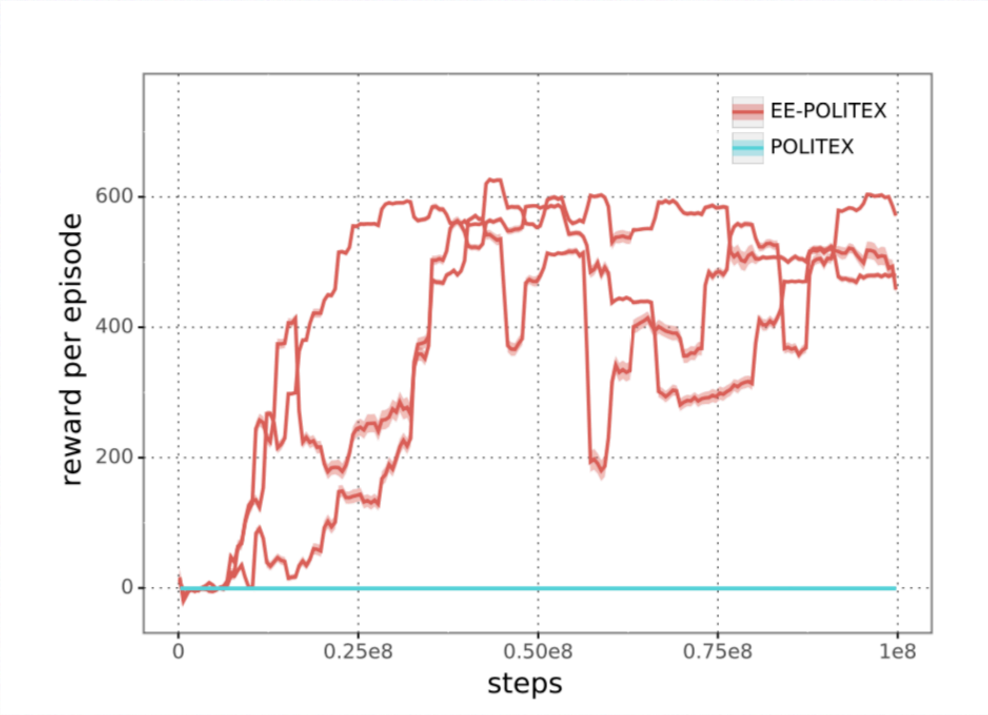
DeepMind

Experimental results: DeepSea



Swingup





EE-POLITEX
POLITEX
600
400
200
0
reward per episode
0
0.25e8
0.50e8
0.75e8
1e8
steps

Summary & future work

- First algorithm guaranteed to work in non-realizable VFA setting
  - Theoretical guarantees, also seems to works in practice!
- Adaptive learning rate/optimistic mirror descent to reduce regret
- Same family as MPO/PPO → why KL regularization?
  - But: Represent policy instead of Q values
  - And: Tunes learning rate differently - with KL.
- Future:
  - Find good pure exploration policies, continuous actions, more experiments.

Related work

- E. Even-Dar, S. M. Kakade, and Y. Mansour. "Online MDPs." Mathematics of Operations Research 34.3 (2009).
- H. Yu and D. P. Bertsekas. "Convergence results for some temporal difference methods based on least squares." IEEE Transactions on Automatic Control 54.7 (2009)
- Ian Osband, Zheng Wen, and Benjamin Van Roy. Generalization and exploration via randomized value functions. ICML, 2016.
- Degrave et al., Quinoa. NeurIPS DeepRL Workshop, 2018.
- Abdolmaleki et al., Maximum a-posteriori policy optimization. ICLR, 2018.
- Y. Abbasi-Yadkori, N. Lazić, and C. Szepesvári. "Regret bounds for model-free LQ control." AISTATS, 2019.