A Boosting Approach to Reinforcement Learning

Nataly Brukhim

Princeton University

NeurIPS 2022

joint work with

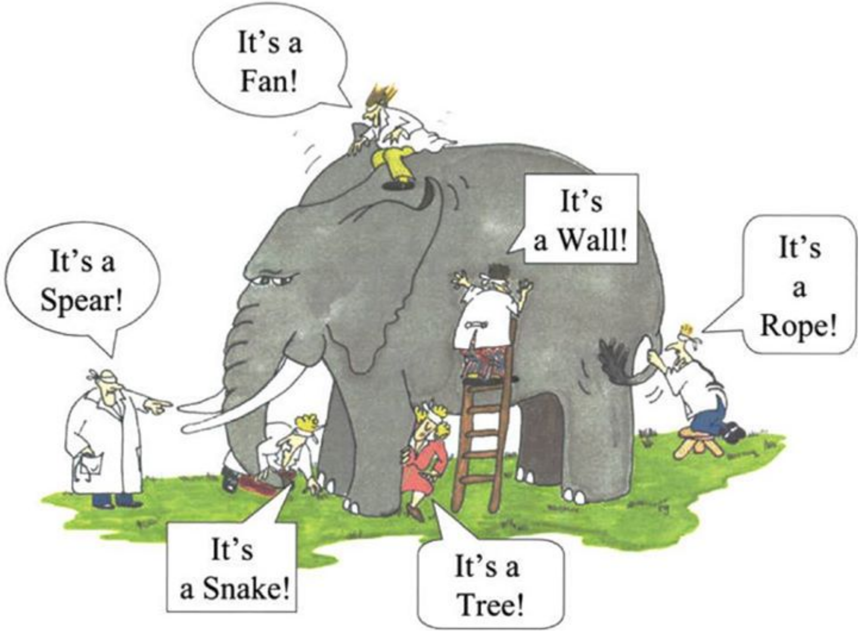
Elad Hazan
Google & Princeton University

Karan Singh
Carnegie Mellon University

Boosting

- Framework for combining ***weak*** learners to produce a ***strong*** learner

mildly accurate *arbitrarily good accuracy*



Boosting

- Framework for combining ***weak*** learners to produce a ***strong*** learner

  mildly accurate *arbitrarily good accuracy*

- ***Adaboost*** *(Adaptive Boosting)* [Freund-Schapire ’95]

  Gödel Prize 2003

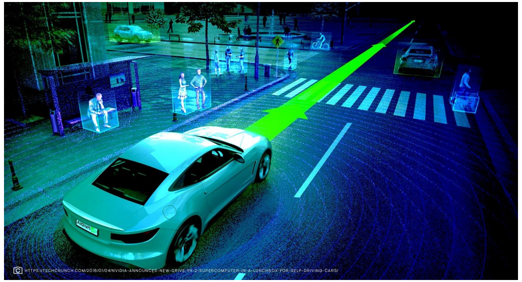
- Boosting is a fundamental methodology in ML with both:
  - Tremendous **practical** success
  - Solid **theoretical** foundations

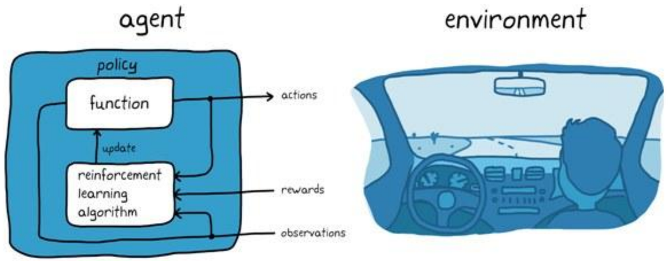
Boosting

- Well-understood, mature theory for Supervised Learning.

- Can we leverage this powerful tool to do

  Reinforcement Learning?





Markov Decision Process

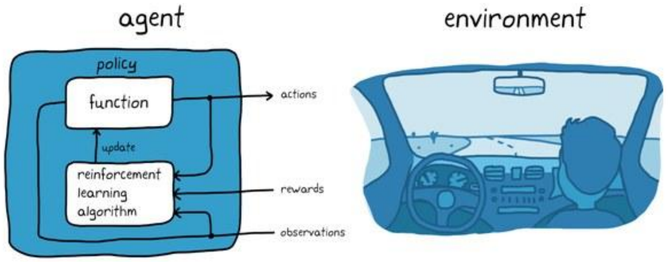
$s \in \mathcal{S}$ **states**

$a \in \mathcal{A}$ discrete **actions**

$r' \sim R(\cdot|s,a)$ **rewards**

$s' \sim P(\cdot|s,a)$ **transition** model

$$V^\pi = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| a_t \sim \pi(s_t)\right]$$



Goal Select **policy** $\pi \in \Pi \subset \mathcal{S} \to \Delta_{\mathcal{A}}$ to minimize $V^* - V^\pi$

Boosting for *Reinforcement Learning*

Weak
supervised learner
[small policy class]

$$succ(\pi) \geq \alpha \max_{\pi^* \in \Pi} succ(\pi^*)$$

Boosting for *Reinforcement Learning*

Weak
supervised learner
[small policy class]

→

Strong
supervised learner
[large policy class]

$$succ(\pi) \geq \alpha \max_{\pi^* \in \Pi} succ(\pi^*) \qquad succ(\pi) \geq 1 \cdot \max_{\pi^* \in \bar{\Pi}} succ(\pi^*) - \epsilon$$

Boosting for *Reinforcement Learning*

Weak supervised learner [small policy class] → **Strong** supervised learner [large policy class] → **Strong** **RL** learner [competitive w. OPT]

$$succ(\pi) \geq \alpha \max_{\pi^* \in \Pi} succ(\pi^*) \qquad succ(\pi) \geq 1 \cdot \max_{\pi^* \in \bar{\Pi}} succ(\pi^*) - \epsilon \qquad V(\pi) \geq \max_{\pi^*} V(\pi^*) - \mathcal{E}$$

Boosting for *Reinforcement Learning*

Main Result

An ***efficient*** boosting algorithm that when given a **weak** classifier with α edge:

$$succ(\pi) \geq \alpha \max_{\pi^* \in \Pi} succ(\pi^*)$$

outputs a policy to ε-minimize $V^* - V^\pi$ in $O(\text{poly}(\alpha, \varepsilon^{-1}, |\mathcal{A}|))$ episodes

- Sample complexity ***independent*** of $|\mathcal{S}|$ *(number of states)*
- Assuming Policy Completeness, State Coverage *(see the paper for details)*

Boosting for *Reinforcement Learning*

(Main Result) RL via Weak Learning

Given an exploratory policy class

Given access to an exploratory reset dist.

	Supervised weak learner	Online weak learner
Episodic model	$1/\alpha^4\varepsilon^5$	$1/\alpha^2\varepsilon^3$
Rollouts w. ν-resets	$1/\alpha^4\varepsilon^6$	$1/\alpha^2\varepsilon^4$

Type of weak learner

Boosting for *Reinforcement Learning*

(Main Result) RL via Weak Learning

Given an exploratory policy class

Given access to an exploratory reset dist.

Type of weak learner

	Supervised weak learner	Online weak learner
Episodic model	$1/\alpha^4\varepsilon^5$	$1/\alpha^2\varepsilon^3$
Rollouts w. ν-resets	$1/\alpha^4\varepsilon^6$	$1/\alpha^2\varepsilon^4$

(Bonus) RL via Supervised Learning

(improvement over known results in some settings)

	This work	CPI
Episodic model	$1/\varepsilon^3$	$1/\varepsilon^4$
Rollouts w. ν-resets	$1/\varepsilon^4$	$1/\varepsilon^4$

Boosting for *Reinforcement Learning*

Weak
supervised learner
[small policy class]

→

Strong
supervised learner
[large policy class]

→

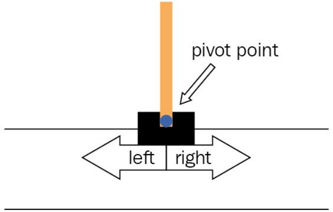
Strong
RL learner
[competitive w. OPT]

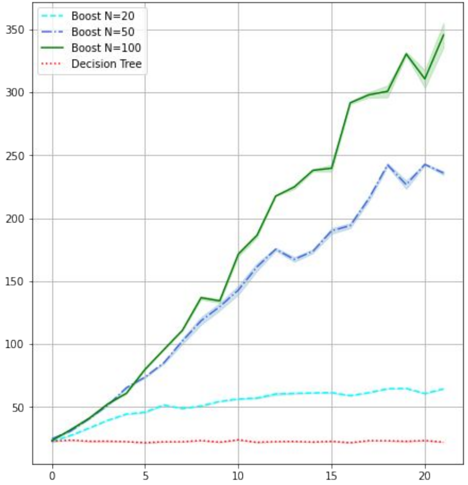
$succ(\pi) \geq \alpha \max_{\pi^* \in \Pi} succ(\pi^*)$ $\quad$ $succ(\pi) \geq 1 \cdot \max_{\pi^* \in \bar{\Pi}} succ(\pi^*) - \epsilon$ $\quad$ $V(\pi) \geq \max_{\pi^*} V(\pi^*) - \mathcal{E}$

End Result:

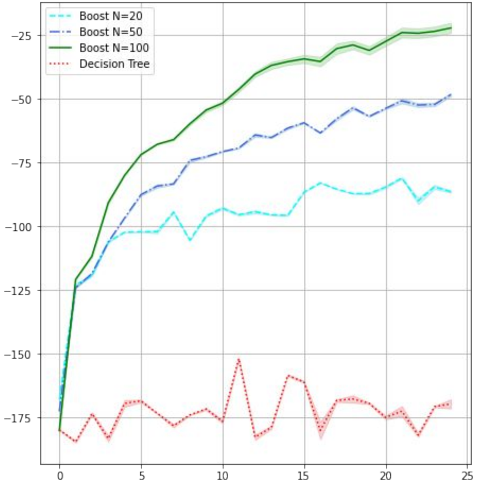
- *Depth-2* neural network on top of weak learners to boost accuracy
- Uses recent *agnostic* boosting results [Hazan-Singh’21,Brukhim-Hazan’21].
- Improvements on the RL to SL reduction
    - Novel analysis of the *Frank-Wolfe* method for non-convex functions

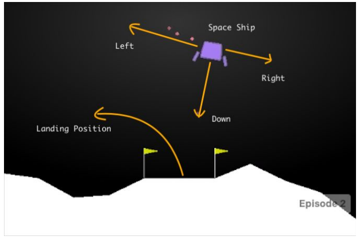
Prelim Experiments





CartPole





LunarLander