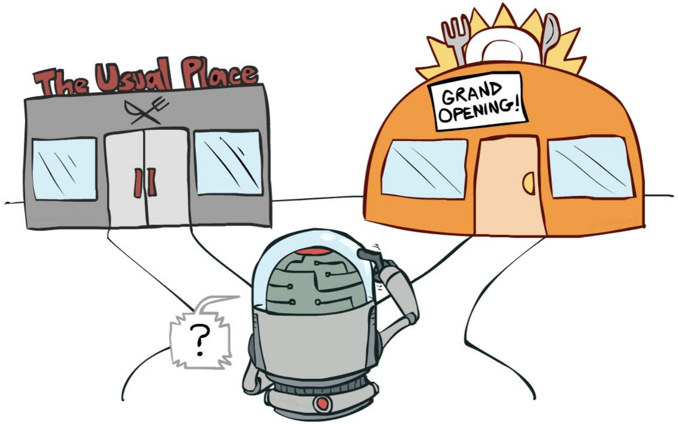
Deep Exploration via Randomized Value Function

RLRG 2022W2
Helen Zhang



Slides partially stolen from Bingshan's OTS presentation, and Chris and Jason' MLRG in 2019, and various paper/poster/slides from authors

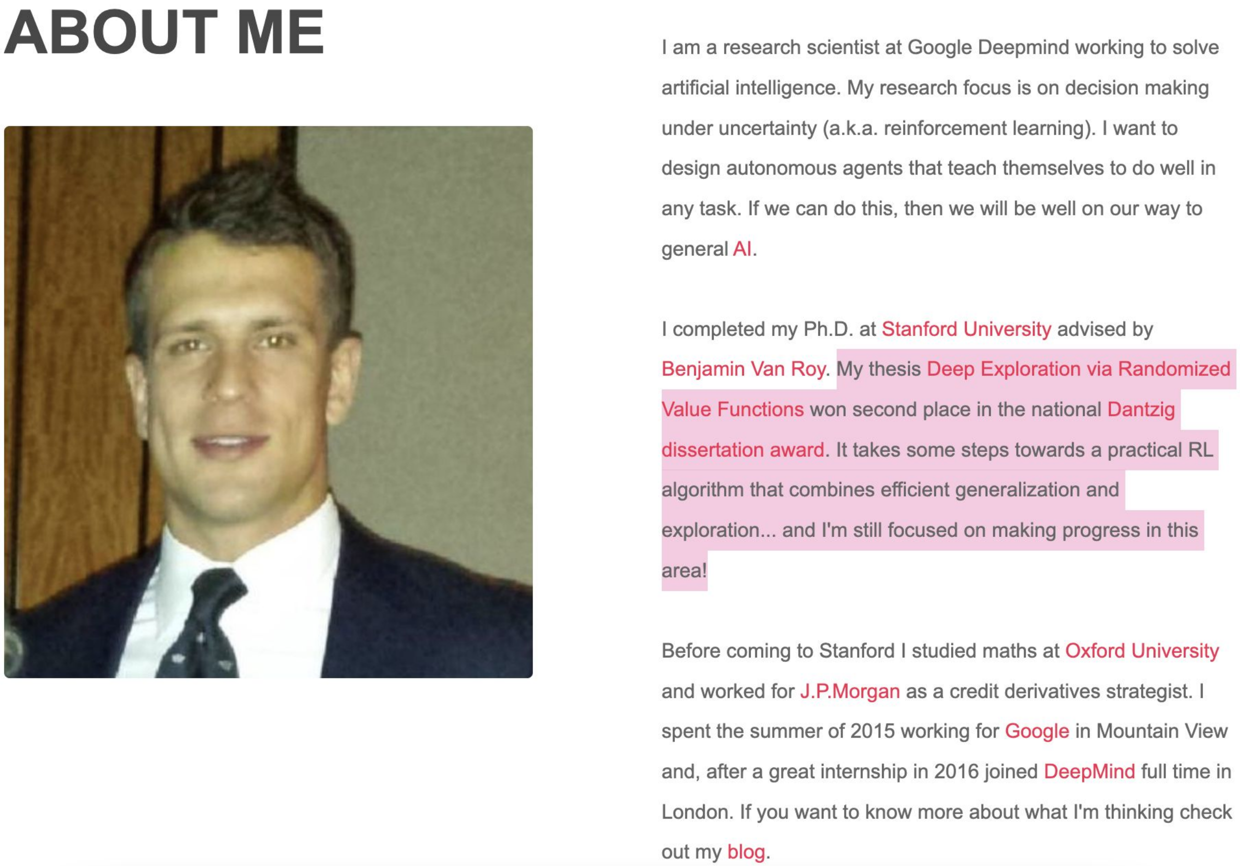
Authors

Ian Osband
DeepMind

Benjamin Van Roy
Stanford University

Daniel J. Russo
Columbia University

Zheng Wen
Adobe Research



ABOUT ME

I am a research scientist at Google Deepmind working to solve artificial intelligence. My research focus is on decision making under uncertainty (a.k.a. reinforcement learning). I want to design autonomous agents that teach themselves to do well in any task. If we can do this, then we will be well on our way to general AI.

I completed my Ph.D. at Stanford University advised by Benjamin Van Roy. My thesis Deep Exploration via Randomized Value Functions won second place in the national Dantzig dissertation award. It takes some steps towards a practical RL algorithm that combines efficient generalization and exploration... and I'm still focused on making progress in this area!

Before coming to Stanford I studied maths at Oxford University and worked for J.P.Morgan as a credit derivatives strategist. I spent the summer of 2015 working for Google in Mountain View and, after a great internship in 2016 joined DeepMind full time in London. If you want to know more about what I'm thinking check out my blog.

Outline

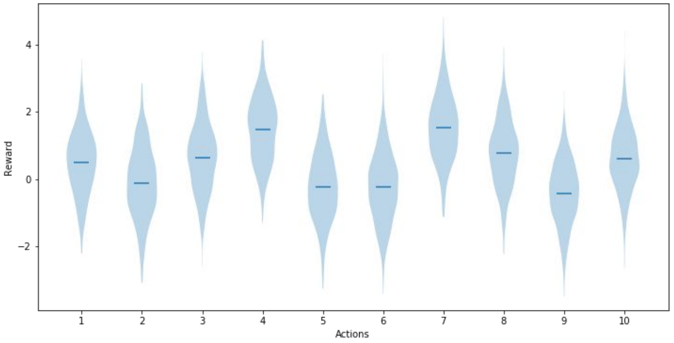
- Bandits
    - UCB
    - Thompson Sampling
- MDP
    - Least square value iteration
    - RLSVI: exploration via randomized value function
- Regret Analysis
- Practical Variants/Experiments

Exploration in Bandits

Stochastic Multi-arm Bandits

A stochastic MAB instance $\Theta := ([K]; \mu_1, \mu_2, \ldots, \mu_K)$
In every round $t = 1, 2, \ldots, T$

1. Environment generates a reward vector $\left(X_1(t), \ldots, \underbrace{X_j(t)}_{\sim \mathrm{Ber}(\mu_j)}, \ldots, X_K(t) \right)$
2. Simultaneously, Learner pulls an arm $J_t \in [K]$
3. Environment reveals $X_{J_t}(t)$; Learner observes and obtains $X_{J_t}(t)$



Goal: minimize regret (equivalent to maximize reward)

Reward of best arm

$$
\begin{aligned}
&\mathcal{R}(T;\Theta) \\
= \;& \mathbb{E}\left[\sum_{t=1}^{T} \left(\max_{j\in[K]} \mu_j - \mu_{J_t} \right)\right] \\
= \;& \sum_{t=1}^{T} \mathbb{E}\left[\mu_* - \mu_{J_t}\right] \\
= \;& \sum_{t=1}^{T} \mathbb{E}\left[\Delta_{J_t}\right] \quad \text{, where } J_t \text{ is random, } \mu_* = \max\nolimits_{j\in[K]} \mu_j, \text{ and } \Delta_j = \mu_* - \mu_j
\end{aligned}
$$

Reward of Algorithm

Upper Confidence Bound

Optimism in the face of uncertainty

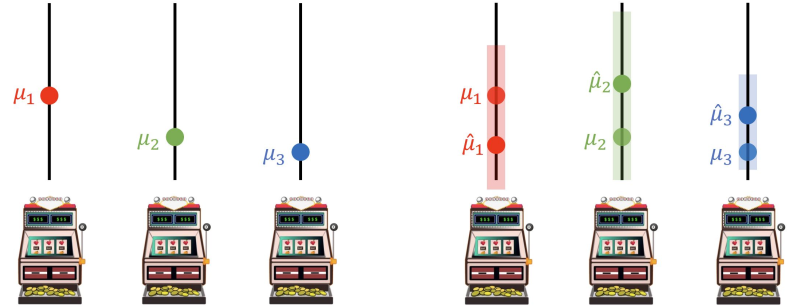
- Compute the empirical mean of each arm and a confidence interval;
- Use the upper confidence bound as a proxy for goodness of arm.

Algorithm 2 UCB

```
for $t = 1, 2, \cdots, T$ do
    $\forall i \in [n]$, compute the upper confidence bound $\bar{\mu}_i(t) = \hat{\mu}_{i,O_i(t-1)} + \sqrt{\frac{2\ln(t)}{O_i(t-1)}}$
    Pull arm $i_t \in \arg\max_{i \in [n]} \bar{\mu}_i(t)$
    Observe reward $x_{i_t}^t$
end for
```

Bonus term

$O_i(t-1)$: Number of times arm i has been pulled



Hoeffding's inequality, w.h.p. :

$$\bar{\mu}_{j,O_j(t-1)} = \hat{\mu}_{j,O_j(t-1)} + \sqrt{\frac{3\ln(t)}{O_j(t-1)}} \geq \mu_j, \forall j \in [K]$$

Importance of optimism:

Arm pulled in round J

UCB of arm pulled in round J

$$\Delta_{J_t} := \mu_* - \mu_{J_t} \leq \bar{\mu}_{*,O_*(t-1)} - \mu_{J_t} \leq \bar{\mu}_{J_t,O_{J_t}(t-1)} - \mu_{J_t} = \sqrt{\frac{3\ln(t)}{O_{J_t}(t-1)}}$$

Best arm

UCB of best arm

Regret: $\sum_{t=1}^{T} \mathbb{E}\left[\Delta_{J_t}\right] \leq \sum_{t=1}^{T} \mathbb{E}\left[\sqrt{\frac{3\ln(t)}{O_{J_t}(t-1)}}\right] = O\left(\sqrt{KT\log(T)}\right)$

Thompson Sampling

> *“Randomly take action according to the probability you believe it is the optimal action” - Thompson 1933*

An empirical MAB instance $\widehat{\Theta} := \left([K]; \hat{\mu}_{1,O_1(t-1)}, \hat{\mu}_{2,O_2(t-1)}, \dots, \hat{\mu}_{K,O_K(t-1)}\right)$

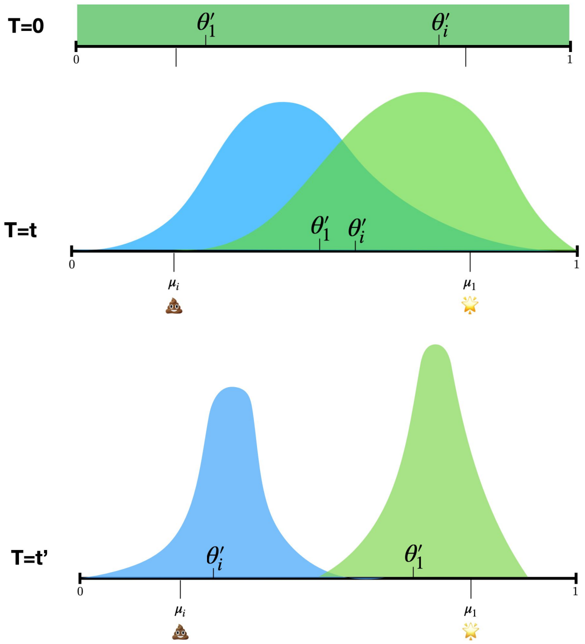
Data-dependent distributions $\widetilde{\theta} := \left([K]; \tilde{\theta}_{1,O_1(t-1)}, \tilde{\theta}_{2,O_2(t-1)}, \dots, \tilde{\theta}_{K,O_K(t-1)}\right),$

where each $\tilde{\theta}_{j,O_j(t-1)} = \mathcal{N}\left(\hat{\mu}_{j,O_j(t-1)}, \frac{3\ln(t)}{O_j(t-1)}\right)$

A sampled MAB instance $\widetilde{\Theta} := \left([K]; \tilde{\mu}_{1,t}, \tilde{\mu}_{2,t}, \dots, \tilde{\mu}_{K,t}\right)$

where each $\tilde{\mu}_{j,t} \sim \tilde{\theta}_{j,O_j(t-1)} \Rightarrow \tilde{\mu}_{j,t} \sim \mathcal{N}\left(\hat{\mu}_{j,O_j(t-1)}, \boxed{\frac{3\ln(t)}{O_j(t-1)}}\right)$

Standard TS: behave greedy in $\widetilde{\Theta}$, pull $J_t \leftarrow \max_{j\in[K]} \tilde{\mu}_{j,t}$



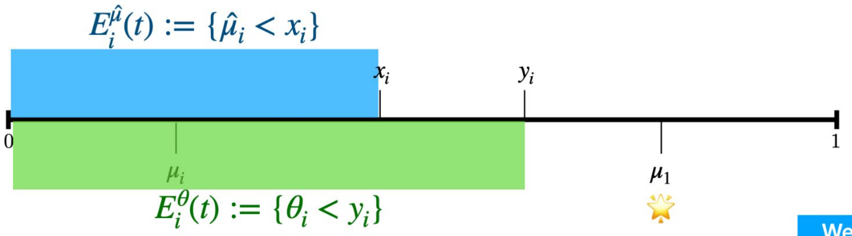
Proof Sketch

Use two events to split up the expectation:

- $E_i^{\theta}(t)$ - the event that the sampled parameter is far from μ_i
- $E_i^{\hat{\mu}}(t)$ - the event that the estimated mean $\hat{\mu}_i$ is from from μ_i

Posterior deviation

Empirical deviation



We'll show that…

$$\mathbb{E}\left[k_i(T)\right] = \sum_{t=1}^{T} \Pr(i(t) = i) = \sum_{t=1}^{T} \Pr\left(i(t) = i, E_i^{\mu}(t), E_i^{\theta}(t)\right)$$

$$+ \sum_{t=1}^{T} \Pr\left(i(t) = i, E_i^{\mu}(t), \overline{E_i^{\theta}(t)}\right)$$

$$+ \sum_{t=1}^{T} \Pr\left(i(t) = i, \overline{E_i^{\mu}(t)}\right)$$

Number of times arm i is pulled

Bounded by linear function prob of playing ⭐

Rare once mean is concentrated

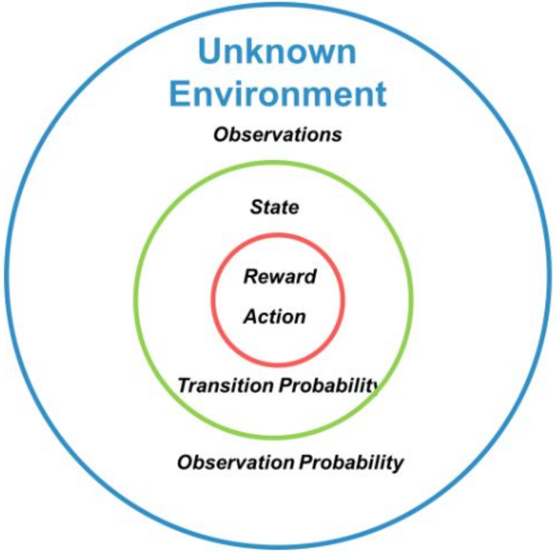
Rare (using Chernoff)

Exploration in MDPs

MDP

Markov Decision Processes (MDPs) provide a framework for modelling **sequential decision making**, where the environment has different states which change over time as a result of the agent’s actions.

- A learning agent draws a trajectory (a sequence of state-action pairs) and try to maximize cumulative reward
- Bandit can be viewed as an MDP with one state and K actions.



Unknown Environment

Observations

State

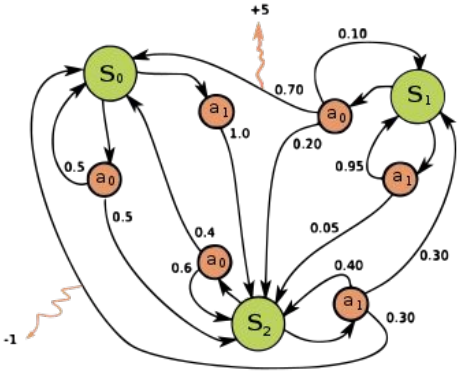
Reward

Action

Transition Probability

Observation Probability

- Bandit Problem
- MDP
- POMDP



Least Square Value Iteration

Adapting value-iteration with imperfect statistical knowledge and limited compute.

```
Algorithm 2 vi
Input:   $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho)$   MDP
         $H \in \mathbb{N}$                                                           planning horizon
Output:  $Q^*_H$                                                                     optimal value function for $H$-period problem
$Q^*_0 \leftarrow 0$
for $h$ in $(0, \dots, H-1)$ do
    $Q^*_{h+1}(s,a) \leftarrow \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,a}(s') \big( \int r \mathcal{R}_{s,a,s'}(dr) + \max_{a' \in \mathcal{A}} Q^*_h(s',a') \big) \quad \forall s,a \in \mathcal{S} \times \mathcal{A}$
return $Q^*_H$
```

Empirical temporal difference loss:

$$\mathcal{L}(\theta;\theta^-,\mathcal{D}) := \sum_{t\in\mathcal{D}} \Big(r_t + \max_{a'\in\mathcal{A}} Q_{\theta^-}(s'_t,a') - Q_\theta(s_t,a_t)\Big)^2$$

Regularized towards prior:

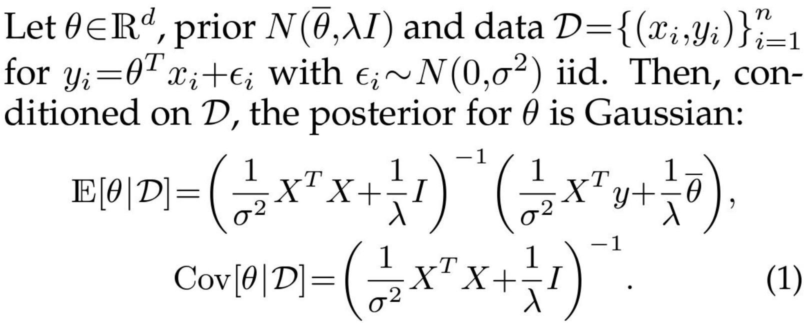
$$\mathcal{R}(\theta;\theta^p) := \frac{v}{\lambda}\|\theta^p - \theta\|_2^2.$$

```
Algorithm 3 learn_lsvi
Agent:    $\mathcal{L}(\theta{=}\cdot;\theta^-{=}\cdot,\mathcal{D}{=}\cdot)$   TD error loss function
          $\mathcal{R}(\theta{=}\cdot;\theta^p{=}\cdot)$                        regularization function
          buffer                                                                memory buffer of observations
          prior                                                                 prior distribution of $\theta$
          $H \in \mathbb{N}$                                                    planning horizon
Updates:  $\tilde{\theta}$                                                      agent value function estimate
$\tilde{\theta}_0 \leftarrow$ null
Data $\tilde{\mathcal{D}} \leftarrow$ buffer.data()
Prior parameter $\tilde{\theta}^p \leftarrow$ prior.mean()
for $h$ in $(0, \dots, H-1)$ do
    $\tilde{\theta}_{h+1} \leftarrow \operatorname{argmin}_{\theta \in \mathbb{R}^D} \big( \mathcal{L}(\theta;\tilde{\theta}_h,\tilde{\mathcal{D}}) + \mathcal{R}(\theta;\tilde{\theta}^p) \big)$
update value function estimate $\tilde{\theta} \leftarrow \tilde{\theta}_H$
```

Randomized LSVI

Key idea: replace least square computation with an alternative value iteration that trains on randomly perturbed version of the data

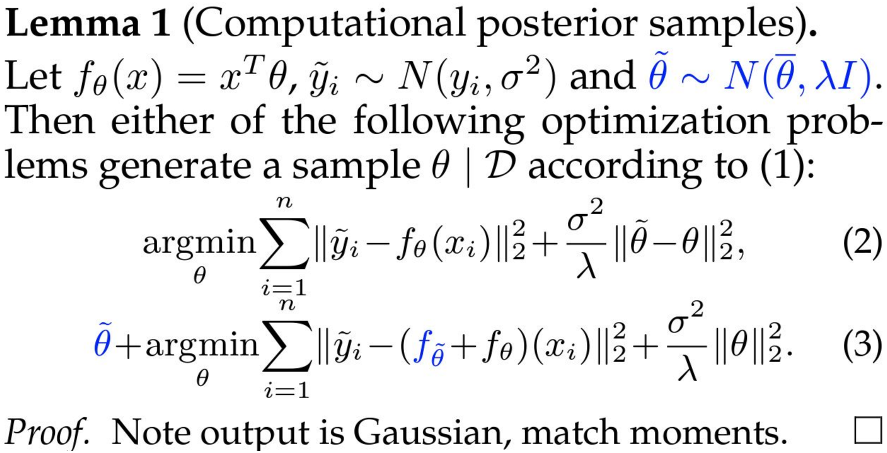
- Consider conventional linear regression:



Let $\theta \in \mathbb{R}^d$, prior $N(\overline{\theta}, \lambda I)$ and data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ for $y_i = \theta^T x_i + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma^2)$ iid. Then, conditioned on $\mathcal{D}$, the posterior for θ is Gaussian:

$$\mathbb{E}[\theta|\mathcal{D}] = \left(\frac{1}{\sigma^2} X^T X + \frac{1}{\lambda} I\right)^{-1} \left(\frac{1}{\sigma^2} X^T y + \frac{1}{\lambda} \overline{\theta}\right),$$

$$\mathrm{Cov}[\theta|\mathcal{D}] = \left(\frac{1}{\sigma^2} X^T X + \frac{1}{\lambda} I\right)^{-1}. \qquad (1)$$



Lemma 1 (Computational posterior samples). Let $f_\theta(x) = x^T \theta$, $\tilde{y}_i \sim N(y_i, \sigma^2)$ and $\tilde{\theta} \sim N(\overline{\theta}, \lambda I)$. Then either of the following optimization problems generate a sample $\theta \mid \mathcal{D}$ according to (1):

$$\underset{\theta}{\mathrm{argmin}} \sum_{i=1}^n \|\tilde{y}_i - f_\theta(x_i)\|_2^2 + \frac{\sigma^2}{\lambda} \|\tilde{\theta} - \theta\|_2^2, \qquad (2)$$

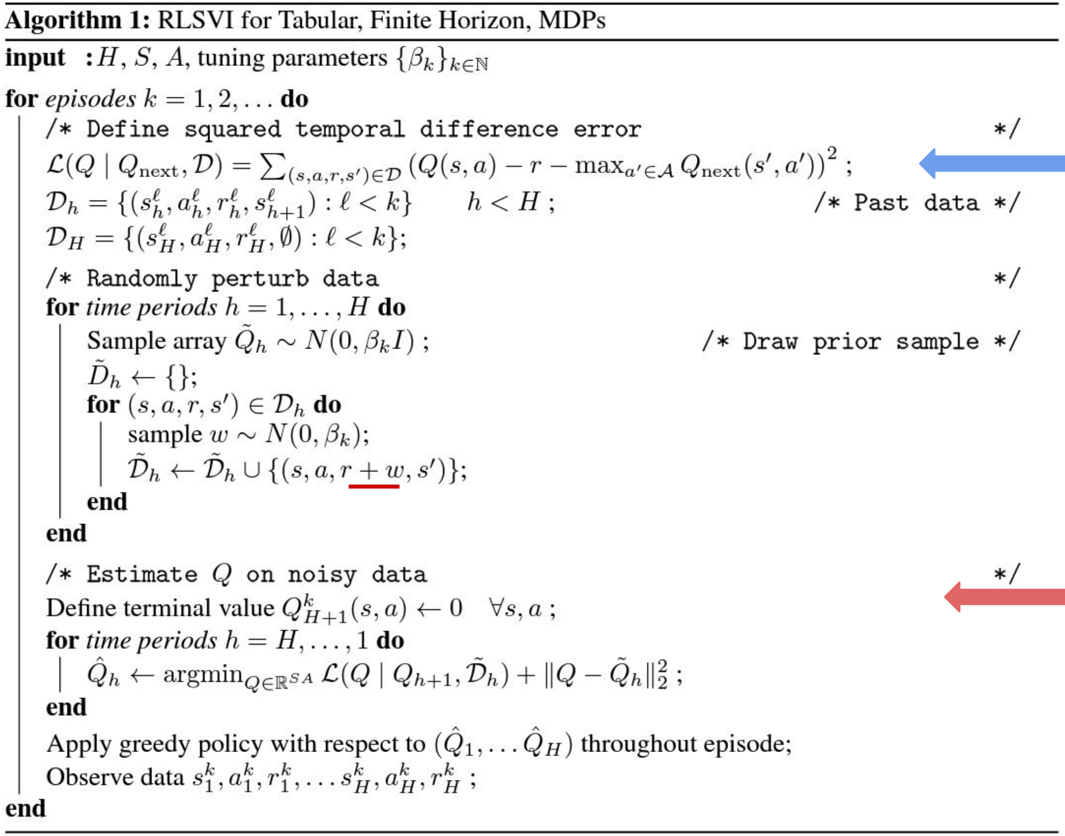
$$\tilde{\theta} + \underset{\theta}{\mathrm{argmin}} \sum_{i=1}^n \|\tilde{y}_i - (f_{\tilde{\theta}} + f_\theta)(x_i)\|_2^2 + \frac{\sigma^2}{\lambda} \|\theta\|_2^2. \qquad (3)$$

Proof. Note output is Gaussian, match moments. ☐

Relies on Gaussian conjugacy and linear models, which cannot easily be extended to deep NN

Computationally tractable approximate posterior, drive deep exploration via randomized value functions.

Algorithm



Algorithm 1: RLSVI for Tabular, Finite Horizon, MDPs

```
input  : $H$, $S$, $A$, tuning parameters $\{\beta_k\}_{k \in \mathbb{N}}$
for episodes $k = 1, 2, \dots$ do
    /* Define squared temporal difference error */
    $\mathcal{L}(Q \mid Q_{\text{next}}, \mathcal{D}) = \sum_{(s,a,r,s') \in \mathcal{D}} \left(Q(s,a) - r - \max_{a' \in \mathcal{A}} Q_{\text{next}}(s', a')\right)^2$ ;
    $\mathcal{D}_h = \{(s_h^\ell, a_h^\ell, r_h^\ell, s_{h+1}^\ell) : \ell < k\} \quad h < H$ ;    /* Past data */
    $\mathcal{D}_H = \{(s_H^\ell, a_H^\ell, r_H^\ell, \emptyset) : \ell < k\}$;
    /* Randomly perturb data */
    for time periods $h = 1, \dots, H$ do
        Sample array $\tilde{Q}_h \sim N(0, \beta_k I)$ ;    /* Draw prior sample */
        $\tilde{\mathcal{D}}_h \leftarrow \{\}$;
        for $(s, a, r, s') \in \mathcal{D}_h$ do
            sample $w \sim N(0, \beta_k)$;
            $\tilde{\mathcal{D}}_h \leftarrow \tilde{\mathcal{D}}_h \cup \{(s, a, \underline{r + w}, s')\}$;
        end
    end
    /* Estimate Q on noisy data */
    Define terminal value $Q_{H+1}^k(s, a) \leftarrow 0 \quad \forall s, a$ ;
    for time periods $h = H, \dots, 1$ do
        $\hat{Q}_h \leftarrow \operatorname{argmin}_{Q \in \mathbb{R}^{SA}} \mathcal{L}(Q \mid Q_{h+1}, \tilde{\mathcal{D}}_h) + \|Q - \tilde{Q}_h\|_2^2$ ;
    end
    Apply greedy policy with respect to $(\hat{Q}_1, \dots \hat{Q}_H)$ throughout episode;
    Observe data $s_1^k, a_1^k, r_1^k, \dots s_H^k, a_H^k, r_H^k$ ;
end
```

Least square regression

Draw noise from gaussian

Perturb dataset with noisy reward

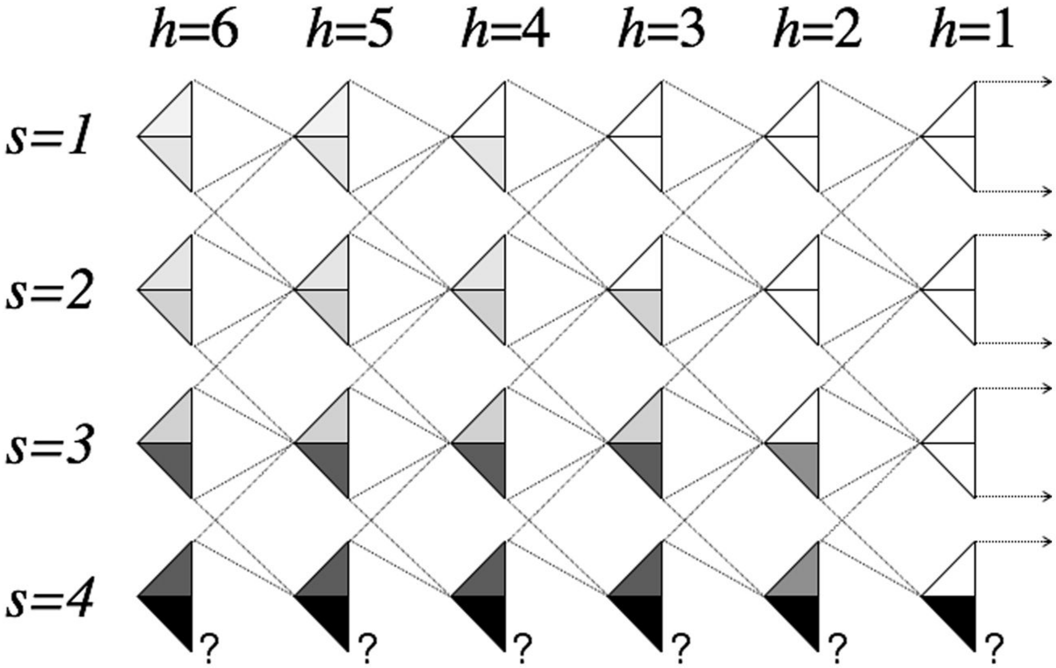
Compute Q function on noisy data

Run greedily

Deep Exploration Intuition

Consider a simple MDP with 4 stats, 2 actions

Suppose we are highly uncertain about state-action pair (4, down), but are pretty sure about others.



Regret Analysis

Finite-horizon Time-inhomogeneous MDP

Assumption 2 (Finite-horizon time-inhomogeneous MDP). *The state space factorizes as* $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{H-1}$ *where* $|\mathcal{S}_0| = \cdots = |\mathcal{S}_{H-1}| < \infty$. *For any MDP* $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \rho)$,

$$\sum_{s' \in \mathcal{S}_{t+1}} \mathcal{P}_{s,a}(s') = 1 \qquad \forall t \in \{0, \ldots, H-2\},\ s \in \mathcal{S}_t,\ a \in \mathcal{A},$$

and

$$\sum_{s' \in \mathcal{S}} \mathcal{P}_{s,a}(s') = 0 \qquad \forall s \in \mathcal{S}_{H-1},\ a \in \mathcal{A}.$$

Each state $s \in \mathcal{S}_t$ can be written as a pair $s = (t, x)$ where $t \in \{0, \ldots, H-1\}$ and $x \in \mathcal{X} = \{1, \ldots, |\mathcal{S}_0|\}$. Similarly, a policy $\pi : \mathcal{S} \to \mathcal{A}$ can be viewed as a sequence $\pi = (\pi_0, \ldots, \pi_{H-1})$ where $\pi_t : x \mapsto \pi((t, x))$. Our notation can be specialized to this time-inhomogenous problem, writing transition probabilities as $\mathcal{P}_{t,x,a}(x') \equiv \mathcal{P}_{(t,x),a}((t+1, x'))$ and reward probabilities as $\mathcal{R}_{t,x,a,x'}(r) \equiv \mathcal{R}_{(t,x),a,(t+1,x')}(r)$. For consistency, we also use different notation for the optimal value function, writing

$$V^{\pi}_{\mathcal{M},t}(x) \equiv V^{\pi}_{\mathcal{M}}((t, x))$$

and define $V^{*}_{\mathcal{M},t}(x) := \max_{\pi} V^{\pi}_{\mathcal{M},t}(x)$. Similarly, we can define the state-action value function under the MDP at timestep $t \in \{0, \ldots, H-1\}$ by

$$Q^{*}_{\mathcal{M},t}(x, a) = \mathbb{E}[r_{t+1} + V^{*}_{\mathcal{M},t+1}(x_{t+1}) \mid \mathcal{M}, x_t = x, a_t = a] \qquad \forall x \in \mathcal{X}, a \in \mathcal{A}.$$

Bayesian Regret Bound

Average over distribution

Regret / L should converge to 0

Value of optimal policy

$$\mathrm{Regret}(\mathcal{M}, \mathrm{alg}, L) = \sum_{\ell=1}^{L} \mathbb{E}_{\mathcal{M},\mathrm{alg}} \left[V^{*}(s_0^{\ell}) - V^{\pi^{\ell}}(s_0^{\ell}) \right]$$

$$\mathrm{BayesRegret}(\mathrm{alg}, L) = \mathbb{E}\left[\mathrm{Regret}(\mathcal{M}, \mathrm{alg}, L)\right].$$

For $|\mathcal{S}_0| = \ldots = |\mathcal{S}_{H-1}| = |\mathcal{X}|$,

$$\mathrm{BayesRegret}(\mathrm{RLSVI}_{\bar{\theta},v,\lambda}, L) \leq 6H^2\sqrt{\beta|\mathcal{X}||\mathcal{A}|L\log_{+}(1+|\mathcal{X}||\mathcal{A}|HL)}\,\log_{+}\left(1+\frac{L}{|\mathcal{X}||\mathcal{A}|}\right),$$

RLSVI requires a number of episodes that is just linear in the number of states to reach near optimal performance.

Regret Decomposition

(Hiding a lot of details…)

Optimal

Algorithm’s estimate value

True expected value of π

$$\underline{V^*_{\mathcal{M},0}(x)} - \underline{V^\pi_{\mathcal{M},0}(x)} = \left(\underline{\max_{a\in\mathcal{A}} Q^*_{\mathcal{M},0}(x,a)} - \underline{\max_{a\in\mathcal{A}} Q_0(x,a)}\right) + \left(\underline{\max_{a\in\mathcal{A}} Q_0(x,a)} - \underline{V^\pi_{\mathcal{M},0}(x)}\right)$$

$$\begin{aligned} = \quad & \max\nolimits_{a\in\mathcal{A}} Q^*_{\mathcal{M},0}(x,a) - \max\nolimits_{a\in\mathcal{A}} Q_0(x,a) && \text{(pessimism of } Q_0\text{)} \\ + \quad & \mathbb{E}_{\mathcal{M},\pi}\left[\textstyle\sum_{t=0}^{H-1}(Q_t - F_{\mathcal{M},t}Q_{t+1})(x_t,a_t) \mid x_0 = x\right] && \text{(on policy Bellman error)} \end{aligned}$$

Bellman Operator

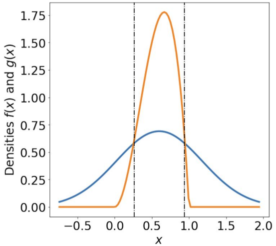
If the function Q_0 is optimistic at an initial state x, in the sense that $\max_a Q_0(x,a) \geq \max_a Q^*_{\mathcal{M},0}(x,a)$, then regret in the episode is bounded by on policy Bellman error under $(Q_0, \ldots, Q_H)$.

Stochastic Optimism

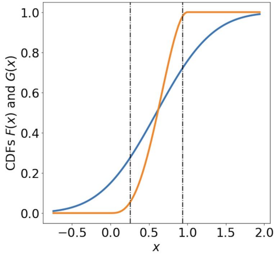
Assumption 3 (Independent Dirichlet prior for outcomes).
Rewards take values in $\{0,1\}$ *and so the cardinality of the outcome space is* $|\mathcal{X}\times\{0,1\}| = 2|\mathcal{X}|$. *For each,* $(t,x,a) \in \{0,\ldots,H-2\}\times\mathcal{X}\times\mathcal{A}$, *the outcome distribution is drawn from a Dirichlet prior*

$$\mathcal{P}^{O}_{t,x,a}(\cdot) \sim \mathrm{Dirichlet}(\alpha_{0,t,x,a})$$

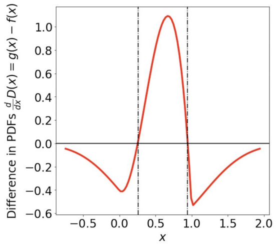
for $\alpha_{0,t,x,a} \in \mathbb{R}^{2|\mathcal{X}|}_{+}$ *and each* $\mathcal{P}^{O}_{t,x,a}$ *is drawn independently across* (t,x,a). *Assume there is* $\beta \geq 3$ *such that* $\mathbb{1}^{T}\alpha_{0,t,a,x} = \beta$ *for all* (t,x,a).



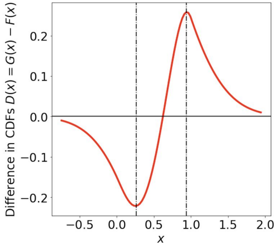
(a) Comparison of PDFs



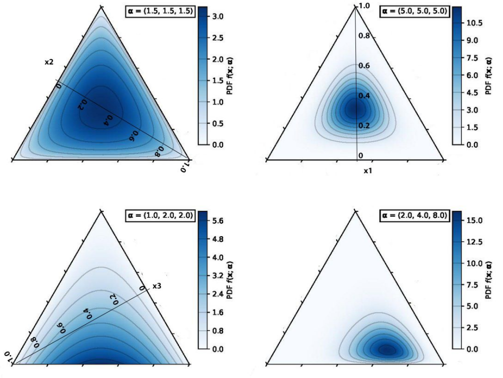
(b) Comparison of CDFs



(c) Difference in PDFs $D'(s) = g(s) - f(s)$



(d) Difference in CDFs $D(s) = G(s) - F(s)$



Definition 2 (Stochastic optimism).
A random variable X *is stochastically optimistic with respect to another random variable* Y, *written* $X \geq_{SO} Y$, *if for all convex increasing functions* $u:\mathbb{R}\to\mathbb{R}$

$$\mathbb{E}[u(X)] \geq \mathbb{E}[u(Y)]. \tag{6.7}$$

Lemma 4 (Gaussian vs Dirichlet optimism).
Let $Y = P^{T}V$ *for* $V \in \mathbb{R}^{n}$ *fixed and* $P \sim \mathrm{Dirichlet}(\alpha)$ *with* $\alpha \in \mathbb{R}^{n}_{+}$ *and* $\sum_{i=1}^{n}\alpha_i \geq 3$. *Let* $X \sim N(\mu,\sigma^2)$ *with* $\mu \geq \frac{\sum_{i=1}^{n}\alpha_i V_i}{\sum_{i=1}^{n}\alpha_i}$, $\sigma^2 \geq 3\left(\sum_{i=1}^{n}\alpha_i\right)^{-1}\mathrm{Span}(V)^2$, *then* $X \geq_{SO} Y$.

Bellman operator underlying RLSVI is stochastically optimistic relative to the true Bellman operator

Bellman Error

Expirical Bellman Update

$$(6.4) \qquad F_{\ell,t}Q(x,a) = \frac{(v/\lambda)\bar{\theta} + n_\ell(y)V_Q^T\hat{\mathcal{P}}^O_{\ell,y}}{(v/\lambda) + n_\ell(y)} + w_\ell(y) \qquad \forall y = (t,x,a).$$

By equation (6.4), we find

$$F_{\ell,t}Q(x,a) - \mathbb{E}[F_{\mathcal{M},t}Q(x,a)|\mathcal{H}_{\ell-1}, x_1^\ell, a_1^\ell, .., x_t^\ell, a_t^\ell] \le \frac{\beta(\|\bar{\theta}\|_\infty + \boxed{\|V_Q\|_\infty})}{\beta + n_\ell(y)} + \boxed{w_\ell(y)}.$$

By Gaussian maximal inequality:

Corollary 3. *For each $t \le H$ and $\ell \le L$*

$$\mathbb{E}[w_\ell(t,x_t,a_t)] \le \sqrt{2\log(|\mathcal{A}||\mathcal{X}|)\mathbb{E}[\sigma_\ell(t,x_t,a_t)^2]}.$$

Bounding noise term

Corollary 4. *If RLSVI is applied with parameters $(\lambda, v, \bar{\theta})$ with $v/\lambda = \beta \ge 3$, $v = 3H^2$ and $\bar{\theta} = H\mathbb{1}$,*

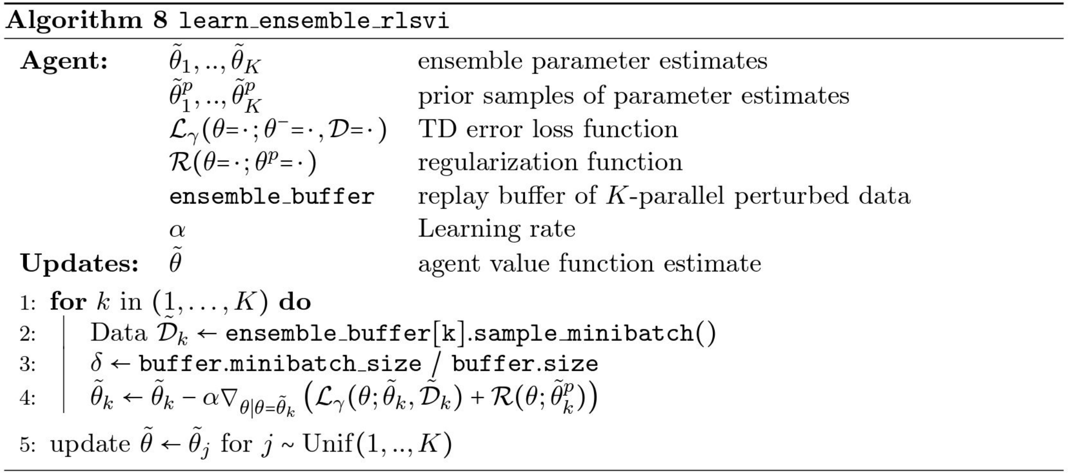
$$\mathbb{E}[\max_{\ell \le L, t<H} \|V_{Q_{\ell,t+1}}\|_\infty] \le 2H + H^2\sqrt{2\log(1 + |\mathcal{X}||\mathcal{A}|HL)}.$$

Bounding norm of value function sampled by RLSVI

Practical Variants/Experiments

Practical Variants

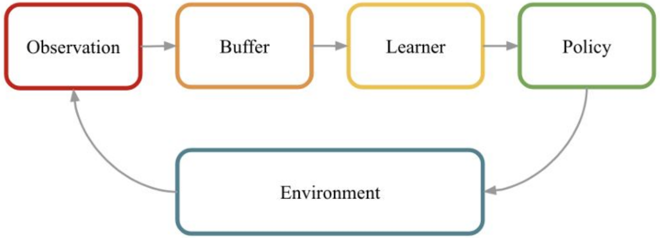
- Finite buffer experience replay
- Discount factor approximating effective planning horizon
- Incremental parameter update with (batch) gradient descent
- Ensemble sampling



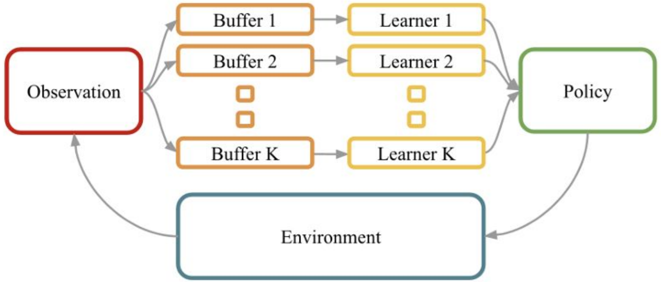
Algorithm 8 `learn_ensemble_rlsvi`

```
Agent:    $\tilde{\theta}_1, .., \tilde{\theta}_K$                                  ensemble parameter estimates
          $\tilde{\theta}^p_1, .., \tilde{\theta}^p_K$                              prior samples of parameter estimates
          $\mathcal{L}_\gamma(\theta{=}\cdot; \theta^-{=}\cdot, \mathcal{D}{=}\cdot)$   TD error loss function
          $\mathcal{R}(\theta{=}\cdot; \theta^p{=}\cdot)$                           regularization function
          ensemble_buffer                                                           replay buffer of $K$-parallel perturbed data
          $\alpha$                                                                  Learning rate
Updates:  $\tilde{\theta}$                                                          agent value function estimate

for $k$ in $(1, \ldots, K)$ do
    Data $\tilde{\mathcal{D}}_k \leftarrow$ ensemble_buffer[k].sample_minibatch()
    $\delta \leftarrow$ buffer.minibatch_size / buffer.size
    $\tilde{\theta}_k \leftarrow \tilde{\theta}_k - \alpha \nabla_{\theta|\theta=\tilde{\theta}_k} \left( \mathcal{L}_\gamma(\theta; \tilde{\theta}_k, \tilde{\mathcal{D}}_k) + \mathcal{R}(\theta; \tilde{\theta}^p_k) \right)$
update $\tilde{\theta} \leftarrow \tilde{\theta}_j$ for $j \sim \mathrm{Unif}(1, .., K)$
```

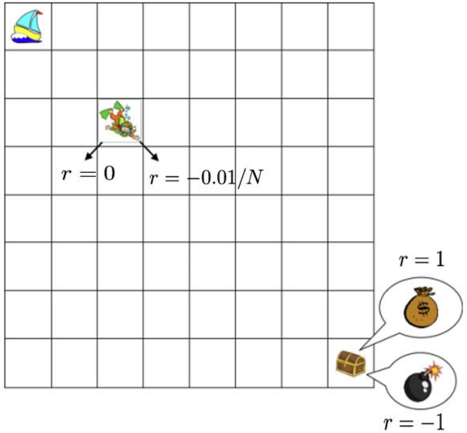


(a) learning a single value function



(b) learning multiple value functions in parallel

Tabular: DeepSea



- **Environment description:**
    - ○ State space = N x N grid.
    - ○ Begin top left, fall one row each step.
    - ○ Actions “left” or “right” vary per state.
    - ○ Big reward +1 in chest.
    - ○ Small cost -0.1/N for moving “right”.

- **1 policy > 0, 1 policy = 0, all others < 0.**
- ***… “a piece of hay in a needle-stack”***

- **No deep exploration** → 2^N episodes to learn.

‘Time to learn’ := #episodes until AveRegret < 0.9.

- ϵ-greedy = DQN with annealing dithering.
- BS = BootDQN without explicit prior.
- BSR = BootDQN with regularize $\|\theta_k - \theta_k^{\text{init}}\|$.
- BSP = BootDQN with prior, $Q_k = f_{\theta_k} + p_k$.

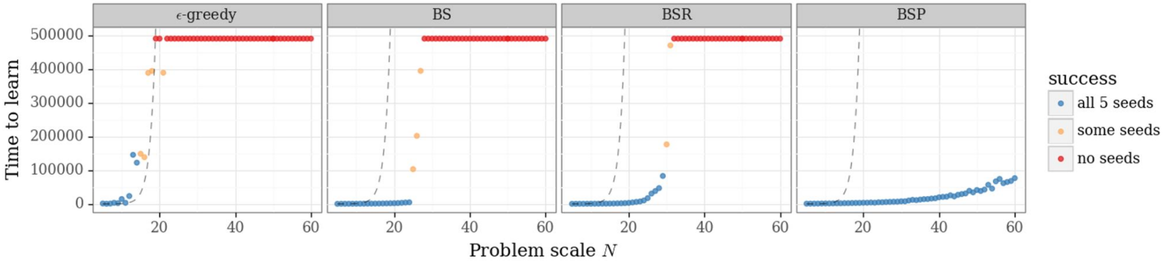


Figure 3: Only BSP scales to large problems. Plotting log-log suggests an empirical scaling $T_{\text{learn}} = \tilde{O}(N^3)$.

Deep Learning: Cart-Pole Swing Up

Agent begins each episode with the pole hanging down and has to learn to swing it up.

Reward structure requires deep exploration:

- Agent pays a cost for any action
- Gets reward if pole is balanced up right

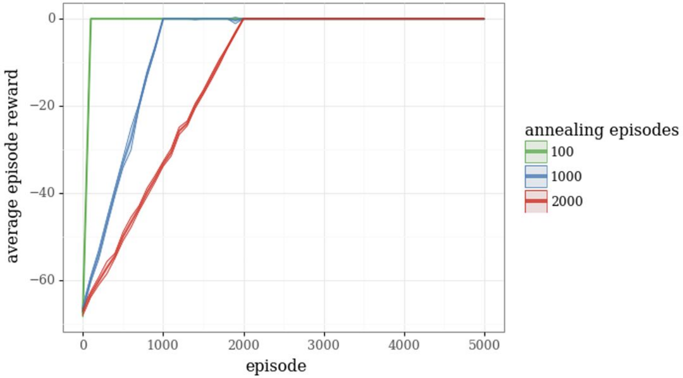


Figure 16: DQN with ϵ-greedy exploration simply learns to stay motionless.

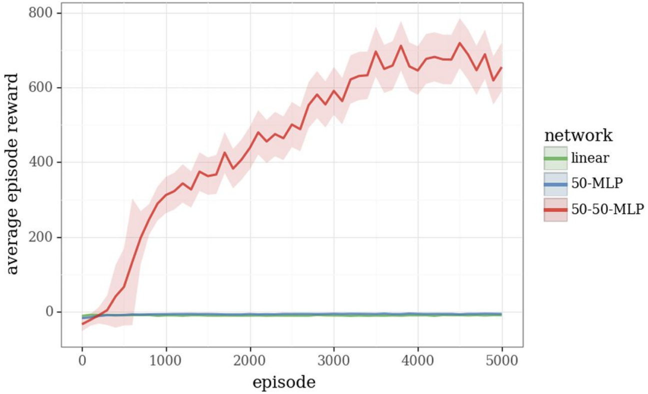


Figure 17: RLSVI with 2-layer neural network is able to learn a near-optimal policy.

Thanks for listening!

And happy to hear any questions and feedbacks :)