Scalable Coordinated Exploration in Concurrent Reinforcement Learning

Maria Dimakopoulou, Ian Osband, Benjamin Van Roy

Stanford University

DeepMind

Motivation

We consider a team of reinforcement learning agents that concurrently learn to operate in a common environment, such as **a farm of robots learning how to carry out a task**.

Google Brain robot farm.

A **larger number of robots** can gather and share **larger volumes of data** that enable each one to learn faster. The **benefits to scale are most dramatic** if the robots explore the environment in a **coordinated fashion**.

Coordinated Exploration

Efficient coordination among agents greatly accelerates learning. There are three necessary properties:
Adaptivity: Adapt as data becomes available to make effective use of new information.
Commitment: Maintain the intent to carry out action sequences that span multiple periods.
Diversity: Divide-and-conquer learning opportunities.

Contribution

We propose **generalized seed sampling** for concurrent reinforcement learning that:

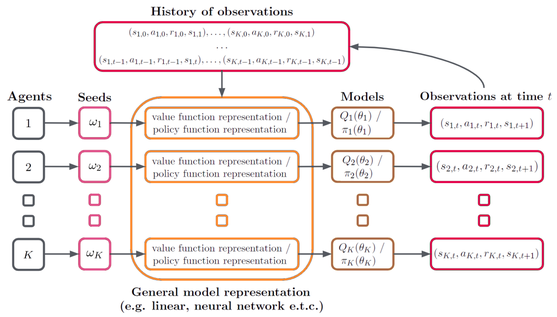
1. satisfies the three necessary properties for efficient coordinated exploration, **adaptivity**, **commitment**, **diversity**.
2. admits **complex generalization** based on randomized value functions to address practical problems that typically pose **enormous state spaces**.

Problem Formulation

- K agents that operate in parallel in identical environments and collaborate to achieve a common goal.
- Agents share data with one another in real time and have access to a common buffer $\mathcal{D}$ with (s,a,r,s') observations.
- Agents generalize across enormous state space $\mathcal{S}$ and action space $\mathcal{A}$ with feature representation $\Phi:\mathcal{S}\times\mathcal{A}\to\mathbb{R}^d$.
- Each agent k uses state-action value function $Q_k(\theta):\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ parameterized by θ (e.g., neural network with input $\Phi(s,a)$ and weights θ).
- Agents have prior beliefs over the parameter θ, such as the expectation, $\bar{\theta}$, or the level of uncertainty, λ, on θ.
- Agents update their state-action value functions in real-time based on all observations made by their peers.

Generalized Seed Sampling

Generalized seed sampling offers a framework for designing scalable and efficient coordinated exploration algorithms in concurrent reinforcement learning.



Lemma 1. Consider the data $(X,y)=(\{x_j\}_{j=1}^N,\{y_j\}_{j=1}^N)$, where $y_j=\theta^{*\top}x_j+\epsilon_j$, with iid $\epsilon_j\sim\mathcal{N}(0,v)$. Let $f_\theta=\theta^\top x$, $\hat{\theta}\sim\mathcal{N}(\bar{\theta},\lambda I)$ and $z_j\sim\mathcal{N}(0,v)$. Then, the solution to $\text{argmin}_\theta\left(\frac{1}{v}\sum_j(y_j+z_j-f_\theta(x_i))^2+\frac{1}{\lambda}\|\theta-\hat{\theta}\|_2^2\right)$ is a sample from the posterior of θ^* given (X,y).

Each agent fits a model to a randomly perturbed prior and randomly perturbed observations to generate approximate posterior samples of the state action value function ($\simeq$ Thompson sampling). The seed that each agent samples at the beginning provides the source of randomness.

- The **independent seeds** help **diversify** the exploratory effort among agents.
- The fact that the agent maintains a **fixed seed** throughout learning leads to a sufficient degree of **commitment**.
- At each time period, agent k obtains an **approximate posterior sample** θ_k for the model parameters and uses state-action value function $Q_k(\theta_k)$ (**Lemma 1**). Hence, the agent **adapts** in real-time to new, high-dimensional information.

Seed Least Squares Value Iteration

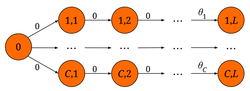
Seeds of agent k: $z_{k,j}\sim\mathcal{N}(0,v)$ and $\hat{\theta}_k\sim\mathcal{N}(\bar{\theta},\lambda I)$.
Before each action, agent k performs LSVI:
$\tilde{\theta}_H\leftarrow 0$
for $h=H-1,H-2,\dots,0$
$\tilde{\theta}_h\leftarrow\text{argmin}_\theta\left(\frac{1}{v}\sum_{j\in\mathcal{D}}\left(r_j+\gamma\max_{a\in\mathcal{A}}\tilde{Q}_{k,\tilde{\theta}_{h+1}}(s'_j,a)+z_{k,j}-\tilde{Q}_{k,\theta}(s_j,a_j)\right)^2+\frac{1}{\lambda}\|\theta-\hat{\theta}_k\|^2\right)$
$\theta_k\leftarrow\tilde{\theta}_0$

Seed Temporal Differencing

Seeds of agent k: $z_{k,j}\sim\mathcal{N}(0,v)$ and $\hat{\theta}_k\sim\mathcal{N}(\bar{\theta},\lambda I)$.
Before each action, agent k performs GD on minibatch $\mathcal{J}\in\mathcal{D}$:
$\mathcal{L}(\theta)\leftarrow\frac{1}{v}\sum_{j\in\mathcal{J}}\left(r_j+\gamma\max_{a\in\mathcal{A}}\tilde{Q}_{k,\tilde{\theta}}(s'_j,a)+z_{k,j}-\tilde{Q}_{k,\theta}(s_j,a_j)\right)^2+\frac{1}{\lambda}\|\theta-\hat{\theta}_k\|^2$
$\theta_k\leftarrow\theta_k-\alpha\nabla_\theta\mathcal{L}(\theta_k)$

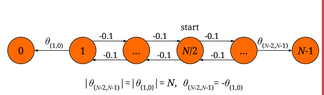
Sanity Checks

We use the "Parallel Chains" & "Bipolar Chain" toy domains.



$\theta_c\sim N(0,\sigma_c^2)$, $\sigma_1^2<\dots<\sigma_c^2$

Parallel chains example.

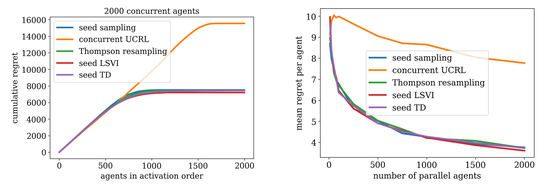


$|\theta_{(N-2,N-1)}|=|\theta_{(1,0)}|=N$, $\theta_{(N-2,N-1)}=-\theta_{(1,0)}$

Bipolar chain example.

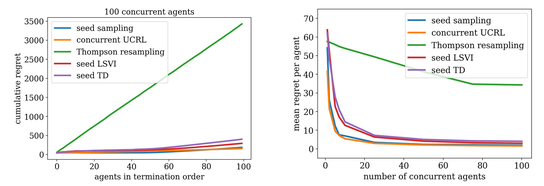
We compare tabular seed sampling, concurrent UCRL, Thompson resampling, seed LSVI and seed TD. Seed LSVI and seed TD use linear representation of the state-action value function with one-hot encoded features.

1. Parallel Chains: Tabular seed sampling, seed LSVI, seed TD (independent seeds) and Thompson resampling (independent MDPs) diversify. Concurrent UCRL makes the agents gather the same data.



Performance of algorithms in the parallel chains problem.

2. Bipolar Chain: Seed sampling, seed LSVI, seed TD (fixed seed) and concurrent UCRL (optimism) commit to explore the endpoints of the chain. Thompson resampling makes the agents dither.



Performance of algorithms in the bipolar chain problem.

Seed sampling algorithms with generalization **adapt, diversify and commit**. Even with a completely uninformative prior, they **perform as well as** the very informed **tabular seed sampling** designed for tabular settings.

Seed Bootstrap

Rather than adding explicit noise to the target values, train models (with LSVI/TD) on **bootstrap samples**. The seed of agent k is $\hat{\theta}_k\sim\mathcal{N}(\bar{\theta},\lambda I)$ and $z_{k,j}\sim$Bernoulli specifying if observation j will be used by agent k **throughout learning**.

Seed Policy Gradient

Seeding principles apply for policy function approximation. Each agent k defines a policy function $\tilde{\pi}_k(s,a,\theta)$ and before an action, it uses the buffer of observations $\mathcal{D}$ and its reward perturbation seeds $z_k\sim\mathcal{N}(0,v)$ to perform policy gradient.

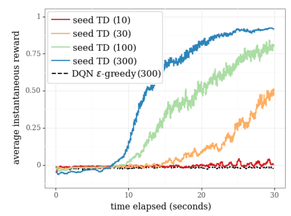
Seed Ensemble

When the number of parallel agents is large, instead of having each one of the K agents fit a separate model (e.g. K separate neural networks), we can have **a smaller ensemble of E models** to decrease computational cost. Each model e is initialized with $\hat{\theta}_e\sim\mathcal{N}(\bar{\theta},\lambda)$ and trained on the buffer $\mathcal{D}$ with reward perturbations $z_{e,j}\sim\mathcal{N}(0,v)$. The seed of agent k is a randomly drawn index of a model from the ensemble.

Cartpole: Swing-Up & Center

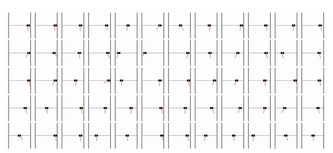
Multiple robots are given their own cart-pole to play with for **30 seconds**.
Goal: learn to swing-up and balance the pole upright while centering the cart – the only rewarding state.

- Due to the curse of dimensionality, tabular approaches are intractable with multiple continuous state variables.
- Due to the highly sparse reward structure, deep and coordinated exploration is necessary.
- **Only generalized seed sampling scales in intractable state spaces and achieves coordinated exploration.**

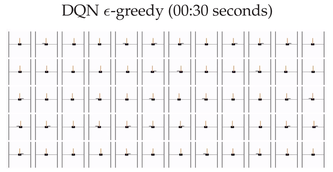


Generalized seed sampling and DQN ϵ-greedy after 30 seconds of learning.

As the number of parallel agents grows, generalized seed sampling is able crack complex tasks very quickly.



DQN ϵ-greedy (00:30 seconds)



Generalized seed sampling (00:30 seconds)

Demos

A team of mice explore a maze for a large round of cheese.

Generalized seed sampling in 'cart-pole: swing-up and center'.