Near-optimal Reinforcement Learning in Factored MDPs

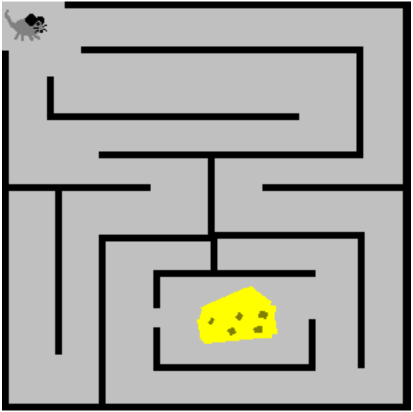
Ian Osband Benjamin Van Roy

Mangement Science and Engineering
Stanford University
iosband@stanford.edu

INFORMS 2014

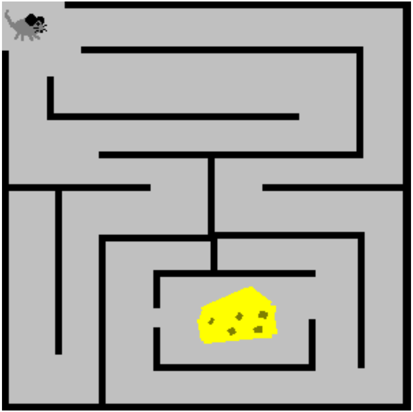
Table of contents

A mouse in a maze



- Simple model:
  "Mice love cheese".
- Put a mouse and some cheese together in a maze.
- How should the mouse get as much cheese as possible?

A mouse in a maze



- Simple model:
  "Mice love cheese".
- Put a mouse and some cheese together in a maze.
- How should the mouse get as much cheese as possible?
- Do **not** provide a map.

A mouse in a maze

- The mouse faces a sequential decision problem.
- At each timestep t the mouse must choose an **action**:
  $a_t \in \{\text{move up, down, left, right, eat}\}$.
- This choice of action will influence both:
  its immediate **reward** r_t (how much cheese it ate)
  its **state** at the following time step s_{t+1}.
- The mouse's goal is to maximize its cumulative rewards through time, not just the reward in any single timestep t.
- We can model this problem as a ***Markov Decision Process***.

Markov Decision Process

- An **agent** taking actions in an **enivronment** M (the MDP).
- **State** s_t encodes the relevant data on the environment.
- **Action** a_t is chosen by the agent.
- The agent receives a **reward** $r_t \sim R^M(s_t, a_t)$.
- The state **transitions** according to $s_{t+1} \sim P^M(s_t, a_t)$.

Maze as an MDP

- **Agent** $\leftrightarrow$ Mouse.
- **Environment** $\leftrightarrow$ Maze + Cheese.
- **State** $\leftrightarrow$ Position of the mouse.
- **Reward** $\leftrightarrow$ 1 if mouse eats cheese, 0 otherwise.
- **Transition** $\leftrightarrow$ Movement blocked by walls.

MDP notation

- Finite horizon MDP $M = (\mathcal{S}, \mathcal{A}, R^M, P^M, \tau, \rho)$.

- **Policy** μ is a function mapping each state $s \in \mathcal{S}$ and $i = 1, \ldots, \tau$ to an action $a \in \mathcal{A}$.

- For each MDP M and policy μ, we define a **value function**:

$$V^M_{\mu,i}(s) := \mathbb{E}_{M,\mu}\left[\sum_{j=i}^{\tau} \overline{R}^M(s_j, a_j) \middle| s_i = s\right].$$

- A policy μ is **optimal** for the MDP M if $V^M_{\mu,i}(s) = \max_{\mu'} V^M_{\mu',i}(s)$ for all $s \in \mathcal{S}$ and $i = 1, \ldots, \tau$.

- We write μ^M **as the optimal policy for** M.

Reinforcement learning

Reinforcement learning

- Agent interacts with an MDP just as before.
- **BUT** the agent is uncertain over dynamics R^M and P^M.
- The agent observes the outcomes of the states and actions it visits and so can learn about the MDP through time.
- Fundamental tradeoff:

Reinforcement learning

- Agent interacts with an MDP just as before.
- **BUT** the agent is uncertain over dynamics R^M and P^M.
- The agent observes the outcomes of the states and actions it visits and so can learn about the MDP through time.
- Fundamental tradeoff: **exploration versus exploitation**.

Reinforcement learning

- Agent interacts with an MDP just as before.
- **BUT** the agent is uncertain over dynamics R^M and P^M.
- The agent observes the outcomes of the states and actions it visits and so can learn about the MDP through time.
- Fundamental tradeoff: **exploration versus exploitation**.
- *The mouse does not know the maze the first time.*

Reinforcement learning

- Repeated episodes of length τ, initial distribution ρ.
  Episode k starts at $t_k := (k-1)\tau + 1$.

- **RL algorithm** is a sequence $\{\pi_k\}_{\mathbb{N}}$ of functions mapping H_{t_k} to a probability distribution $\pi_k(H_{t_k})$ over policies.

- We define the **regret** over episode k wrt the MDP M^*:

$$\Delta_k := \sum_{\mathcal{S}} \rho(s)(V^{M^*}_{\mu^*,1}(s) - V^{M^*}_{\mu_k,1}(s))$$

- And the regret to time T of algorithm π:

$$\mathrm{Regret}(T, \pi, M^*) := \sum_{k=1}^{\lceil T/\tau \rceil} \Delta_k.$$

Why do we care?

- ~~Mice need to get more cheese!~~
- MDPs are great models for sequential decision problems.
- **BUT** we rarely know the appropriate R^M, P^M exactly.
- We want algorithms that will give us performance close to that of the unknown optimal controller for the unknown system!

Why do we care?

- ~~Mice need to get more cheese!~~
- MDPs are great models for sequential decision problems.
- **BUT** we rarely know the appropriate R^M, P^M exactly.
- We want algorithms that will give us performance close to that of the unknown optimal controller for the unknown system!
- Examples: *healthcare, robotics, agriculture, finance and more.*

Algorithms for RL

- The **ϵ-greedy** approach:
  Maintain estimate $\hat{M}$ for M^*. With probability ϵ choose a random policy otherwise use $\mu^{\hat{M}}$ certainty equivalent.

- **Bayes-optimal** strategies:
  Mantain a posterior ϕ for M^*, choose $\mu \in \arg\max_\mu \mathbb{E}[V_\mu^{M^*}]$.

- **Optimism in the face of uncertainty**:
  Maintain a confidence set $\mathcal{M}_k$ that contains M^* with high probability. Choose $\mu_k \in \arg\max_\mu \max_{M \in \mathcal{M}_k} V_\mu^{M_k}$.

- **Posterior sampling**:
  Mantain a posterior ϕ for M^*. Every episode sample $M_k \sim \phi$ and choose μ^{M_k} which is optimal for that sample.

Existing regret bounds

- Naive exploration (ϵ-greedy, Boltzmann) generally take **exponentially** long in $|\mathcal{S}|, |\mathcal{A}|$ to learn the optimal policy.
- Bayes-optimal strategy usually computationally **intractable**.
- Optimism and posterior sampling are closely linked [1].
- **State of the art** regret bounds $\tilde{O}(|\mathcal{S}|\sqrt{|\mathcal{A}|T})$ attained by UCRL2 [2] (optimism) and PSRL [3] (sampling).
- Close to fundamental lower bounds $\Omega(\sqrt{|\mathcal{S}||\mathcal{A}|T})$.

Standard proof outline

Write $V^*_{*,1}$ for $V^{M^*}_{\mu^{M^*},1}$ and similarly k for M_k. The episode regret:

$$\Delta_k = V^*_{*1} - V^*_{k,1} = \left(V^*_1 * - V^k_{k,1}\right) + \left(V^k_{k,1} - V^*_{k,1}\right).$$

Using M_k chosen optimistically, the first term is ≤ 0 with high probability. For posterior sampling it is zero in expectation.

The remaining term can be **decomposed into the Bellman error**:

$$\left(V^k_{k,1} - V^*_{k,1}\right) = \sum_{i=1}^{\tau} \left(\mathcal{T}^k_{k,i} - \mathcal{T}^*_{k,i}\right) V^k_{k,i+1} + \sum_{i=1}^{\tau} d_{t_k+i}.$$

where d_i is a bounded martingale difference and $\mathcal{T}^M_\mu$ is the Bellman operator. The contribution from bounded martingale differences is zero in expectation and bounded $O(\sqrt{T})$ with high probability.

Standard proof outline (cont.)

The Bellman operator is defined

$$\mathcal{T}_{\mu}^{M} V(s) := \overline{R}^{M}(s, \mu(s)) + \sum_{s' \in \mathcal{S}} P^{M}(s'|s, \mu(s)) V(s').$$

Which, together with **Hölder's inequality** allows us to say:

$$\sum_{i=1}^{\tau} \left(\mathcal{T}_{k,i}^{k} - \mathcal{T}_{k,i}^{*} \right) V_{k,i+1}^{k} \leq \sum_{i=1}^{\tau} |\overline{R}^{k} - \overline{R}^{*}| + \frac{1}{2} \Psi_{k} \|P^{k} - P^{*}\|_{1}$$

Where $\Psi_k = max_{s,s'} V_{k,1}^k(s) - V_{k,1}^k(s')$ is the MDP span of M_k. Standard **concentration inequalities** [4] give the convergence of $R_k \to R^*$ and $P_k \to P^*$ at rate $\sqrt{\frac{1}{n}}$ and noting $\sum_{n=1}^{T} \sqrt{\frac{1}{T}} \leq 2\sqrt{T}$ completes the proof.

Problems with these bounds

- These bounds require $T = \Omega(|\mathcal{S}|^2|\mathcal{A}|)$ for guarantees.
- **But** many problems have **$|\mathcal{S}|$ and $|\mathcal{A}|$ large or infinite!**
- Mouse in maze $|\mathcal{S}| = 100, |\mathcal{A}| = 5 \implies |\mathcal{S}|^2|\mathcal{A}| = 50,000$.
- Factory line with 100 machines, each with 3 states, 3 actions:
  $|\mathcal{S}| = 3^{100}, |\mathcal{A}| = 3^{100} \implies |\mathcal{S}|^2|\mathcal{A}| = 3^{300} \simeq 10^{150}$
  Even lower bound $\sqrt{|\mathcal{S}||\mathcal{A}|T}$ requires $T = \Omega(|\mathcal{S}||\mathcal{A}|) \simeq 10^{100}$
- How do you even deal with continuous $\mathcal{S}, \mathcal{A}$?
  Discretisation $\rightarrow$ ***curse of dimensionality***.

Factored MDPs

A production line

At each step, the rewards and transition of each machine only depend upon its neighbours.

- 100 **distinct** machines, each with 3 states and 3 actions.
- Each machine only **directly** affected by its neighbours.
- How should you operate the production line?

A production line

At each step, the rewards and transition of each machine only depend upon its neighbours.

- 100 **distinct** machines, each with 3 states and 3 actions.
- Each machine only **directly** affected by its neighbours.
- How should you operate the production line?
- **Goal:** Exploit some low-dimensional structure.

Exploiting low-dimensional structure

- We know that over all MDPs regret $\Omega(\sqrt{|\mathcal{S}||\mathcal{A}|T})$.
- However, if we know that M has some **low-dimensional structure** we can exploit this to improve guarantees.
- Previous works [5] showed loose sample complexity bounds for RL on factored MDPs. We prove the first regret bounds which are close to optimal.
- We can exploit the graphical structure of a factored MDP to obtain regret bounds that scale with the **parameters of the MDP**, which may be **exponentially smaller than $|\mathcal{S}|$ or $|\mathcal{A}|$**.

Factored MDPs

Let $\mathcal{S} \times \mathcal{A} = \mathcal{X} = \mathcal{X}_1 \times .. \times \mathcal{X}_n$, $Z \subseteq [n]$ and $\mathcal{X}[Z] := \bigotimes_{i \in Z} \mathcal{X}_i$

Definition (Factored transition functions $P \in \mathcal{P} \subseteq \mathcal{P}_{\mathcal{X},\mathcal{S}}$)

$\mathcal{P}$ is factored over $\mathcal{X} = \mathcal{X}_1 \times .. \times \mathcal{X}_n$ and $\mathcal{S} = \mathcal{S}_1 \times .. \times \mathcal{S}_m$ with scopes $Z_1, ..Z_m \iff$, for all $P \in \mathcal{P}, x \in \mathcal{X}, s \in \mathcal{S}$ there exist,

$$P(s|x) = \prod_{i=1}^{m} P_i\left(s[i] \,\middle|\, x[Z_i]\right) \leftarrow \textbf{\textit{(conditional independence)}}$$

Definition (Factored reward functions $R \in \mathcal{R} \subseteq \mathcal{P}^{C,\sigma}_{\mathcal{X},\mathbb{R}}$)

$\mathcal{R}$ is factored over $\mathcal{X} = \mathcal{X}_1 \times .. \times \mathcal{X}_n$ with scopes $Z_1, ..Z_l \iff$, for all $R \in \mathcal{R}, x \in \mathcal{X}$ there exist,

$$\mathbb{E}[R(x)] = \sum_{i=1}^{l} \mathbb{E}\big[R_i(x[Z_i])\big]$$

*with **each** $r_i \sim R_i(x[Z_i])$ **and individually observed**.*

Production line as a factored MDP

- **Agent** $\leftrightarrow$ Production manager.
- **Environment** $\leftrightarrow$ Production line and outputs.
- **State** $\leftrightarrow$ Machine states $s = (s_1, .., s_{100}) \in \{1, 2, 3\}^{100}$.
- **Reward** $\leftrightarrow$ Dollar output from each machine $r = \sum_{i=1}^{100} r_j$.
- **Transition** $\leftrightarrow$ Evolution of each machine state.

Production line as a factored MDP

- **Agent** $\leftrightarrow$ Production manager.
- **Environment** $\leftrightarrow$ Production line and outputs.
- **State** $\leftrightarrow$ Machine states $s = (s_1, .., s_{100}) \in \{1, 2, 3\}^{100}$.
- **Reward** $\leftrightarrow$ Dollar output from each machine $r = \sum_{i=1}^{100} r_j$.
- **Transition** $\leftrightarrow$ Evolution of each machine state.
- **Factored** since over one timestep s_i only depends on its neighbours' states and actions (s_k, a_k) for $k \in \{i-1, i, i+1\}$.

Main results

Posterior Sampling for Reinforcement Learning

Input: Prior ϕ encoding $\mathcal{G}$, $t = 1$
for episodes $k = 1, 2, ..$ **do**
    sample $M_k \sim \phi(\cdot | H_t)$
    compute near-optimal $\mu_k = \Gamma(M_k, \sqrt{\tau / k})$ $\leftarrow$(ADP planner)
    for timesteps $j = 1, .., \tau$ **do**
        sample and apply $a_t = \mu_k(s_t, j)$
        observe $r_t^1, .., r_t^l$ and $s_{t+1}^1, .., s_{t+1}^m$
        $t = t + 1$
    end for
end for

UCRL-Factored

Input: Graph structure $\mathcal{G}$, confidence δ, $t = 1$
for episodes $k = 1, 2, ..$ **do**
    $d_t^{R_i} = 4\sigma^2 \log\left(4l|\mathcal{X}[Z_i^R]|k/\delta\right)$ for $i = 1, .., l$
    $d_t^{P_j} = 4|\mathcal{S}_j| \log\left(4m|\mathcal{X}[Z_j^P]|k/\delta\right)$ for $j = 1, .., m$
    $\mathcal{M}_k = \{M \mid \mathcal{G}, \overline{R}_i \in \mathcal{R}_t^i(d_t^{R_i}), P_j \in \mathcal{P}_t^j(d_t^{P_j}) \ \forall i, j\}$ $\leftarrow$(Confidence sets)
    compute near-optimistic $\mu_k = \tilde{\Gamma}(\mathcal{M}_k, \sqrt{\tau/k})$ $\leftarrow$(Optim. ADP planner)
    for timesteps $u = 1, .., \tau$ **do**
        sample and apply $a_t = \mu_k(s_t, u)$
        observe $r_t^1, .., r_t^l$ and $s_{t+1}^1, .., s_{t+1}^m$
        $t = t + 1$
    end for
end for

Main results

Theorem (Expected regret for PSRL in factored MDPs)

If the prior ϕ is the distribution of M^* and Ψ is the span of the optimal value function:

$$\mathbb{E}\left[\mathrm{Regret}(T, \pi_\tau^{\mathrm{PS}}, M^*)\right] = \tilde{O}\left(\sigma \sum_{i=1}^{d_1} \sqrt{|\mathcal{X}[Z_i^R]| T} + \mathbb{E}[\Psi] \sum_{j=1}^{d_2} \sqrt{|\mathcal{X}[Z_j^P]||\mathcal{S}_j| T}\right) \quad (1)$$

Theorem (High probability regret for UCRL-Factored)

If D is the diameter of M^*, then for any M^* can bound the regret of UCRL-Factored:

$$\mathrm{Regret}(T, \pi_\tau^{\mathrm{UC}}, M^*) = \tilde{O}\left(\sigma \sum_{i=1}^{d_1} \sqrt{|\mathcal{X}[Z_i^R]| T} + CD \sum_{j=1}^{d_2} \sqrt{|\mathcal{X}[Z_j^P]||\mathcal{S}_j| T}\right) \quad (2)$$

with probability at least $1 - \delta$

Discussion of results

- Close link between posterior sampling and optimism [1].
- Bounds in expected regret versus high probability. Different MDP complexity measures, span $\Psi \leq CD$ (diameter) $\leq C\tau$.
- PSRL more **statistically and computationally** efficient [3].
- Known structure $\mathcal{G}$ **exponential** $\rightarrow$ **polynomial** regret.
- Both algorithms require approximate MDP planning.
- **Near optimal** as m independent MDPs $\rightarrow \tilde{O}(mS\sqrt{AT})$.

Clean bounds in the symmetric case

Let $\mathcal{Q}$ be shorthand for the structure $\mathcal{G}$ such that $l+1=m$, $C=\sigma=1$, $|\mathcal{S}_i|=|\mathcal{X}_i|=K$ and $|Z_i^R|=|Z_i^P|=\zeta$ for all suitable i and write $J=K^\zeta$. In this case $\Psi, D \leq \tau$ trivially.

Corollary (Clean bounds for PSRL)

$$\mathbb{E}\left[\mathrm{Regret}(T, \pi_\tau^{\mathrm{PS}}, M^*)\right] \leq 15 m\tau\sqrt{JKT\log(2mJT)} \qquad (3)$$

Corollary (Clean bounds for UCRL-Factored)

$$\mathrm{Regret}(T, \pi_\tau^{\mathrm{UC}}, M^*) \leq 15 m\tau\sqrt{JKT\log(12mJT/\delta)} \qquad (4)$$

with probability at least $1-\delta$.

Bounds for the production line

- 100 different machines, each with 3 states and 3 actions.
- Transitions only depend on neighbours $\to$ **factored MDP**.
- $\mathcal{G}$-naive bounds $|\mathcal{S}|\sqrt{|\mathcal{A}|T} = 3^{250}\sqrt{T} \simeq 10^{120}\sqrt{T}$.
- Using $\mathcal{G}$ we obtain $100\sqrt{(9)^3 3T} \simeq 10^3\sqrt{T}$.
- In general, **bounds exponentially tighter** than $\mathcal{G}$-naive.

Key lemma

Lemma (Bounding factored deviations)

Let the transition function class $\mathcal{P} \subseteq \mathcal{P}_{\mathcal{X},\mathcal{S}}$ be factored over $\mathcal{X} = \mathcal{X}_1 \times .. \times \mathcal{X}_n$ and $\mathcal{S} = \mathcal{S}_1 \times .. \times \mathcal{S}_m$ with scopes $Z_1, ..Z_m$. Then, for any $P, \tilde{P} \in \mathcal{P}$ we may bound their L1 distance:

$$\|P(x) - \tilde{P}(x)\|_1 \leq \sum_{i=1}^{m} \|P_i(x[Z_i]) - \tilde{P}_i(x[Z_i])\|_1$$

Proof:

We begin with the simple claim that for any $\alpha_1, \alpha_2, \beta_1, \beta_2 \in (0, 1]$:

$$\begin{aligned} |\alpha_1\alpha_2 - \beta_1\beta_2| &= \alpha_2 \left|\alpha_1 - \frac{\beta_1\beta_2}{\alpha_2}\right| \\ &\leq \alpha_2 \left(|\alpha_1 - \beta_1| + \left|\beta_1 - \frac{\beta_1\beta_2}{\alpha_2}\right|\right) \\ &\leq \alpha_2 |\alpha_1 - \beta_1| + \beta_1 |\alpha_2 - \beta_2| \end{aligned}$$

Key lemma continued

We now consider the probability distributions $p, \tilde{p}$ over $\{1, .., d_1\}$ and $q, \tilde{q}$ over $\{1, .., d_2\}$. We let $Q = pq^T, \tilde{Q} = \tilde{p}\tilde{q}^T$ be the joint probability distribution over $\{1, .., d_1\} \times \{1, .., d_2\}$. Using the claim above we bound the L1 deviation $\|Q - \tilde{Q}\|_1$ by the deviations of their factors:

$$
\begin{aligned}
\|Q - \tilde{Q}\|_1 &= \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} |p_i q_j - \tilde{p}_i \tilde{q}_j| \\
&\leq \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} q_j |p_i - \tilde{p}_i| + \tilde{p}_i |q_j - \tilde{q}_j| \\
&= \|p - \tilde{p}\|_1 + \|q - \tilde{q}\|_1
\end{aligned}
$$

We conclude the proof by applying this m times to the factored transitions P and $\tilde{P}$.

Conclusions

- **Regret polynomial in the parameters** encoding the factored MDP, which may be **exponentially smaller than $|\mathcal{S}|$ or $|\mathcal{A}|$**.
- Near-optimal regret bounds and simple algorithms.
- Two algorithms based on **posterior sampling** and **optimism**.

Conclusions

- **Regret polynomial in the parameters** encoding the factored MDP, which may be **exponentially smaller than $|\mathcal{S}|$ or $|\mathcal{A}|$**.
- Near-optimal regret bounds and simple algorithms.
- Two algorithms based on **posterior sampling** and **optimism**.
- **BUT**:
  - Algorithms require access to approximate MDP planner.
  - You need to know $\mathcal{G}$ structure a priori.
  - How can you learn without episodic reset τ?
  - What about other large/continuous MDPs with different structure, for example linear-quadratic control? [6].

References

D. Russo and B. Van Roy.
Learning to optimize via posterior sampling.
CoRR, abs/1301.2609, 2013.

Thomas Jaksch, Ronald Ortner, and Peter Auer.
Near-optimal regret bounds for reinforcement learning.
The Journal of Machine Learning Research, 99:1563–1600, 2010.

Ian Osband, Daniel Russo, and Benjamin Van Roy.
(More) Efficient Reinforcement Learning via Posterior Sampling.
Advances in Neural Information Processing Systems, 2013.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger.
Inequalities for the L1 deviation of the empirical distribution.
Hewlett-Packard Labs, Tech. Rep, 2003.

Michael Kearns and Daphne Koller.
Efficient reinforcement learning in factored MDPs.
In *IJCAI*, volume 16, pages 740–747, 1999.

Ian Osband and Benjamin Van Roy.
Model-based reinforcement learning and the eluder dimension.
arXiv preprint arXiv:1406.1853, 2014.