Efficient Bayes-Adaptive Reinforcement Learning using Sample-Based Search

Arthur Guez, David Silver and Peter Dayan

Kevin Xie

Motivation

Solve MDP efficiently when we <u>don’t know dynamics</u>

- Exploration vs Exploitation Trade-off
- RL typically doesn’t value exploration
- Saw some “pure exploration” examples last lecture

High Level Approach

- Define our objective: bayes-optimal policy
- Reformulate problem
  - (original MDP w/ unknown dynamics)
  - -> (much more complicated MDP w/ known dynamics) (BAMDP)
- Monte-Carlo-Tree-Search to solve new MDP
- Approximations to make things tractable
  - This is the original contribution

High Level Approach

- **Define our objective: bayes-optimal policy**
- Reformulate problem
  - (original MDP w/ unknown dynamics)
  - -> (much more complicated MDP w/ known dynamics) (BAMDP)
- Monte-Carlo-Tree-Search to solve new MDP
- Approximations to make things tractable
  - This is the original contribution

Background: Formal Problem Statement

Original MDP: $M = \langle S, A, \tilde{\mathcal{P}}, \mathcal{R}, \gamma \rangle$

Dynamics prior: $P(\mathcal{P})$

Formal Problem Statement

Original MDP: $M = \langle S, A, \mathcal{P}, \mathcal{R}, \gamma \rangle$

Dynamics prior: $P(\mathcal{P})$

History: $h_t = s_1 a_1 s_2 a_2 \dots a_{t-1} s_t$

Bayes Update: $P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P})P(\mathcal{P})$

EE Policy: $\tilde{\pi} : S \times \mathcal{H} \times A \to [0, 1]$

Takes history into account

Formal Problem Statement

Original MDP: $M = \langle S, A, \mathcal{P}, \mathcal{R}, \gamma \rangle$

Dynamics prior: $P(\mathcal{P})$

History: $h_t = s_1 a_1 s_2 a_2 \ldots a_{t-1} s_t$

Bayes Update: $P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P}) P(\mathcal{P})$

EE Policy: $\tilde{\pi} : S \times \mathcal{H} \times A \rightarrow [0, 1]$

Objective: Maximize expected return under $P(\mathcal{P})$ prior

(Expected) expected discounted return

Objective: Maximize expected return under $P(\mathcal{P})$ prior

Expected return v starting at s after seeing history h:

$$v(s, h, \tilde{\pi}) = \mathbb{E}^{\tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, h_0 = h\right]$$

Expectation over all dynamics models

(Expected) expected discounted return

$$
\begin{aligned}
v(s, h, \tilde{\pi}) &= \mathbb{E}^{\tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, h_0 = h\right] \\
&= \int_{\mathcal{P}} \mathrm{d}\mathcal{P}\, P(\mathcal{P}\,|h)\, \boxed{\mathbb{E}^{\tilde{\pi}}_{M(\mathcal{P})}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, h_0 = h\right]}
\end{aligned}
$$

V in Original MDP with fixed P + history

Probability of dynamics given history

Recursive definition

Starting to look like a MDP

$$
\begin{aligned}
v(s,h,\tilde{\pi}) &= \mathbb{E}^{\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t r_t | s_0 = s, h_0 = h\right] \\
&= \int_{\mathcal{P}} \mathrm{d}\mathcal{P}\, P(\mathcal{P}\,|h)\, \mathbb{E}^{\tilde{\pi}}_{M(\mathcal{P})}\left[\sum_{t=0}^{\infty}\gamma^t r_t | s_0 = s, h_0 = h\right] \\
&= \sum_{a_0 \in A} \mathcal{R}(s,a_0) + \gamma \sum_{s' \in S} \tilde{\pi}(s,h,a_0) v(s', h a_0 s', \tilde{\pi}) \bar{\mathcal{P}}(s,a_0,s',h)
\end{aligned}
$$

Marginal dynamics over posterior: $\bar{\mathcal{P}}(s,a,s',h) \equiv \int_{\mathcal{P}} \mathrm{d}\mathcal{P}\, P(\mathcal{P}\,|h)\, \mathcal{P}(s,a,s')$

Bayes-optimal Policy

Definition 1 *Given S, A, $\mathcal{R}$, γ, and a prior distribution $P(\mathcal{P})$ over the dynamics of the* MDP *M, let*

$$v^*(s, \emptyset) = \sup_{\tilde{\pi} \in \tilde{\Pi}} v(s, \emptyset, \tilde{\pi}). \tag{15}$$

Martin (1967, Thm. 3.2.1) shows that there exists a strategy $\tilde{\pi}^* \in \tilde{\Pi}$ that achieves that expected return (i.e., $v(s, \emptyset, \tilde{\pi}^*) = v^*(s, \emptyset)$). Any such EE strategy $\tilde{\pi}^*$ is called a ***Bayes-optimal policy.***[2]

How to compute $\tilde{\pi}^* \in \tilde{\Pi}$?

High Level Approach

- Define our objective: bayes-optimal policy
- **Reformulate problem**
    - (original MDP w/ unknown dynamics)
    - -> (much more complicated MDP w/ known dynamics) (**BAMDP**)
- Monte-Carlo-Tree-Search to solve new MDP
- Approximations to make things tractable
    - This is the original contribution

Bayes-Adaptive MDP (BAMDP)

Original MDP, don't know dynamics, find bayes-optimal policy

BAMDP, know dynamics, find optimal policy

Proposition 1 (Silver, 1963; Martin, 1967) *The optimal policy of the* BAMDP *is the Bayes-optimal policy, as defined in Definition 1.*

Bayes-Adaptive MDP (BAMDP)

Original MDP: $M = \langle S, A, \mathcal{P}, \mathcal{R}, \gamma \rangle$

$S^+ = S \times \mathcal{H}$ ← Add history to state

$\mathcal{R}^+(\langle s, h \rangle, a) = R(s, a)$ ← Reward unchanged

Bayes-Adaptive MDP (BAMDP)

$$S^+ = S \times \mathcal{H}$$

$$\mathcal{R}^+(\langle s, h\rangle, a) = R(s, a)$$

Dynamics Posterior: $P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P})P(\mathcal{P})$

$$\mathcal{P}^+(\langle s, h\rangle, a, \langle s', h'\rangle) = \mathbb{1}[h' = has'] \int_{\mathcal{P}} \mathcal{P}(s, a, s')P(\mathcal{P}|h)\, \mathrm{d}\mathcal{P}$$

Augmented dynamics follow marginalized belief $\bar{\mathcal{P}}(s, a_0, s', h)$

Similar to Model-based RL,
But exploration built-in through history

Bayes-Adaptive MDP (BAMDP)

$$S^+ = S \times \mathcal{H}$$

$$\mathcal{R}^+(\langle s, h \rangle, a) = R(s, a)$$

Dynamics Posterior: $P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P})P(\mathcal{P})$

$$\mathcal{P}^+(\langle s, h \rangle, a, \langle s', h' \rangle) = \mathbb{1}[h' = has'] \int_{\mathcal{P}} \mathcal{P}(s, a, s') P(\mathcal{P}|h) \, \mathrm{d}\mathcal{P}$$

BAMDP: $M^+ = \langle S^+, A, \mathcal{P}^+, \mathcal{R}^+, \gamma \rangle$

We “know” $\mathcal{P}^+$ (by construction)

Bayes-Adaptive MDP (BAMDP)

$$S^{+} = S \times \mathcal{H}$$ |H| Explosion->Intractable

$$\mathcal{R}^{+}(\langle s, h \rangle, a) = R(s, a)$$

More Intractable

Dynamics Posterior: $P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P})P(\mathcal{P})$

$$\mathcal{P}^{+}(\langle s, h \rangle, a, \langle s', h' \rangle) = \mathbb{1}[h' = has'] \int_{\mathcal{P}} \mathcal{P}(s, a, s')P(\mathcal{P}|h)\,\mathrm{d}\mathcal{P}$$

BAMDP: $M^{+} = \langle S^{+}, A, \mathcal{P}^{+}, \mathcal{R}^{+}, \gamma \rangle$

Even More Intractable

We “know” $\mathcal{P}^{+}$ (by construction)

Main Contribution

- Bayes-adaptive monte carlo planner (BAMCP) Algorithm:
- Use MCTS with UCT rule to solve the BAMDP (BA-UCT)
  - Focus on promising branches $S^{+} = S \times \mathcal{H}$
- Introduce tricks for computational efficiency
  - Root Sampling $P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P})P(\mathcal{P})$
  - Lazy Sampling
  - Rollout Policy Learning

BAMCP = BA-UCT + 3 tricks

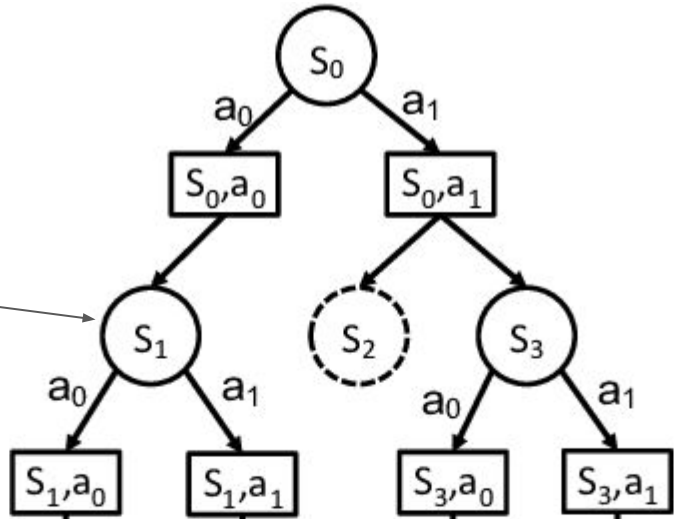
Main Contribution

- Bayes-adaptive monte carlo planner (BAMCP) Algorithm:
- **Use MCTS with UCT rule to solve the BAMDP (BA-UCT)**
    - Focus on promising branches $S^{+} = S \times \mathcal{H}$
- Introduce tricks for computational efficiency
    - Root Sampling
    - Lazy Sampling
    - Rollout Policy Learning

$$P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P})P(\mathcal{P})$$

BAMCP = BA-UCT + 3 tricks

BA-UCT: MCTS with UCB



State visitation: $N(\langle s, h \rangle)$

State action visitation: $N(\langle s, h \rangle, a)$

Q-value estimate: $Q(\langle s, h \rangle, a)$

UCT selection rule: $\text{argmax}_a\, Q(\langle s, h \rangle, a) + c\sqrt{\log(N(\langle s, h \rangle))/N(\langle s, h \rangle, a)}$

Exploit Explore

Rollout policy: $a \sim \pi_{ro}(\langle s, h \rangle, \cdot)$ The monte carlo part

BA-UCT

One simulation in the tree search:

- Simulate given starting state node and fixed dynamics

$$
\begin{array}{l}
\boxed{\textbf{procedure } \texttt{Simulate(} \langle s,h\rangle,\mathcal{P},d)} \\
\quad \textbf{if } \gamma^d R_{\max} < \epsilon \textbf{ then } \textbf{return } 0 \\
\quad \textbf{if } N(\langle s,h\rangle) = 0 \textbf{ then} \\
\quad\quad \textbf{for all } a \in A \textbf{ do} \\
\quad\quad\quad N(\langle s,h\rangle,a) \leftarrow 0, \\
\quad\quad\quad Q(\langle s,h\rangle,a)) \leftarrow 0 \\
\quad\quad \textbf{end} \\
\quad\quad a \sim \pi_{ro}(\langle s,h\rangle,\cdot) \\
\quad\quad s' \sim \mathcal{P}(s,a,\cdot) \\
\quad\quad r \leftarrow \mathcal{R}(s,a) \\
\quad\quad R \leftarrow r + \gamma\ \texttt{Rollout}(\langle s', has'\rangle,\mathcal{P},d) \\
\quad\quad N(\langle s,h\rangle) \leftarrow 1,\ N(\langle s,h\rangle,a) \leftarrow 1 \\
\quad\quad Q(\langle s,h\rangle,a) \leftarrow R \\
\quad\quad \textbf{return } R \\
\quad \textbf{end} \\
\quad a \leftarrow \operatorname*{argmax}_b Q(\langle s,h\rangle,b) + c\sqrt{\frac{\log(N(\langle s,h\rangle))}{N(\langle s,h\rangle,b)}} \\
\quad s' \sim \mathcal{P}(s,a,\cdot) \\
\quad r \leftarrow \mathcal{R}(s,a) \\
\quad R \leftarrow r + \gamma\ \texttt{Simulate}(\langle s', has'\rangle,\mathcal{P},d{+}1) \\
\quad N(\langle s,h\rangle) \leftarrow N(\langle s,h\rangle) + 1 \\
\quad N(\langle s,h\rangle,a) \leftarrow N(\langle s,h\rangle,a) + 1 \\
\quad Q(\langle s,h\rangle,a) \leftarrow Q(\langle s,h\rangle,a) + \frac{R - Q(\langle s,h\rangle,a)}{N(\langle s,h\rangle,a)} \\
\quad \textbf{return } R \\
\textbf{end procedure}
\end{array}
$$

BA-UCT

One simulation in the tree search:

- Simulate given starting state node and fixed dynamics
- If state node is unexplored:
    - Init counters

procedure `Simulate`($\langle s,h\rangle, \mathcal{P}, d$)
  if $\gamma^d R_{\max} < \epsilon$ **then return** 0
  if $N(\langle s,h\rangle) = 0$ **then**
    for all $a \in A$ **do**
      $N(\langle s,h\rangle, a) \leftarrow 0,$
      $Q(\langle s,h\rangle, a)) \leftarrow 0$
    end
    $a \sim \pi_{ro}(\langle s,h\rangle, \cdot)$
    $s' \sim \mathcal{P}(s,a,\cdot)$
    $r \leftarrow \mathcal{R}(s,a)$
    $R \leftarrow r + \gamma$ `Rollout`$(\langle s', has'\rangle, \mathcal{P}, d)$
    $N(\langle s,h\rangle) \leftarrow 1,\ N(\langle s,h\rangle, a) \leftarrow 1$
    $Q(\langle s,h\rangle, a) \leftarrow R$
    return R
  end
  $a \leftarrow \underset{b}{\operatorname{argmax}}\, Q(\langle s,h\rangle, b) + c\sqrt{\frac{\log(N(\langle s,h\rangle))}{N(\langle s,h\rangle, b)}}$
  $s' \sim \mathcal{P}(s,a,\cdot)$
  $r \leftarrow \mathcal{R}(s,a)$
  $R \leftarrow r + \gamma$ `Simulate`$(\langle s', has'\rangle, \mathcal{P}, d{+}1)$
  $N(\langle s,h\rangle) \leftarrow N(\langle s,h\rangle) + 1$
  $N(\langle s,h\rangle, a) \leftarrow N(\langle s,h\rangle, a) + 1$
  $Q(\langle s,h\rangle, a) \leftarrow Q(\langle s,h\rangle, a) + \frac{R - Q(\langle s,h\rangle, a)}{N(\langle s,h\rangle, a)}$
  return R
end procedure

BA-UCT

One simulation in the tree search:

- Simulate given starting state node and fixed dynamics
- If state node is unexplored:
  - Init counters
  - Rollout to get R sample

procedure Simulate($\langle s, h\rangle, \mathcal{P}, d$)

  if $\gamma^d R_{\max} < \epsilon$ **then return** 0

  if $N(\langle s, h\rangle) = 0$ **then**

    for all $a \in A$ **do**

      $N(\langle s, h\rangle, a) \leftarrow 0,$

      $Q(\langle s, h\rangle, a)) \leftarrow 0$

    end

    $a \sim \pi_{ro}(\langle s, h\rangle, \cdot)$

    $s' \sim \mathcal{P}(s, a, \cdot)$

    $r \leftarrow \mathcal{R}(s, a)$

    $R \leftarrow r + \gamma$ Rollout($\langle s', has'\rangle, \mathcal{P}, d$)

    $N(\langle s, h\rangle) \leftarrow 1,\ N(\langle s, h\rangle, a) \leftarrow 1$

    $Q(\langle s, h\rangle, a) \leftarrow R$

    return R

  end

  $a \leftarrow \operatorname*{argmax}_b Q(\langle s, h\rangle, b) + c\sqrt{\frac{\log(N(\langle s,h\rangle))}{N(\langle s,h\rangle,b)}}$

  $s' \sim \mathcal{P}(s, a, \cdot)$

  $r \leftarrow \mathcal{R}(s, a)$

  $R \leftarrow r + \gamma$ Simulate($\langle s', has'\rangle, \mathcal{P}, d+1$)

  $N(\langle s, h\rangle) \leftarrow N(\langle s, h\rangle) + 1$

  $N(\langle s, h\rangle, a) \leftarrow N(\langle s, h\rangle, a) + 1$

  $Q(\langle s, h\rangle, a) \leftarrow Q(\langle s, h\rangle, a) + \frac{R - Q(\langle s,h\rangle,a)}{N(\langle s,h\rangle,a)}$

  return R

end procedure

BA-UCT

One simulation in the tree search:

- Simulate given starting state node and fixed dynamics
- If state node is unexplored:
  - Init counters
  - Rollout to get R sample
  - Update N, Q=R
  - Return R
- Else:

procedure $\texttt{Simulate}(\langle s,h\rangle,\mathcal{P},d)$
  if $\gamma^d R_{\max} < \epsilon$ **then return** 0
  if $N(\langle s,h\rangle)=0$ **then**
    for all $a \in A$ **do**
      $N(\langle s,h\rangle,a) \leftarrow 0,$
      $Q(\langle s,h\rangle,a)) \leftarrow 0$
    end
    $a \sim \pi_{ro}(\langle s,h\rangle,\cdot)$
    $s' \sim \mathcal{P}(s,a,\cdot)$
    $r \leftarrow \mathcal{R}(s,a)$
    $R \leftarrow r+\gamma\,\texttt{Rollout}(\langle s',has'\rangle,\mathcal{P},d)$
    $N(\langle s,h\rangle) \leftarrow 1,\ N(\langle s,h\rangle,a) \leftarrow 1$
    $Q(\langle s,h\rangle,a) \leftarrow R$
    return R
  end
  $a \leftarrow \underset{b}{\operatorname{argmax}}\, Q(\langle s,h\rangle,b)+c\sqrt{\frac{\log(N(\langle s,h\rangle))}{N(\langle s,h\rangle,b)}}$
  $s' \sim \mathcal{P}(s,a,\cdot)$
  $r \leftarrow \mathcal{R}(s,a)$
  $R \leftarrow r+\gamma\,\texttt{Simulate}(\langle s',has'\rangle,\mathcal{P},d{+}1)$
  $N(\langle s,h\rangle) \leftarrow N(\langle s,h\rangle)+1$
  $N(\langle s,h\rangle,a) \leftarrow N(\langle s,h\rangle,a)+1$
  $Q(\langle s,h\rangle,a) \leftarrow Q(\langle s,h\rangle,a)+\frac{R-Q(\langle s,h\rangle,a)}{N(\langle s,h\rangle,a)}$
  return R
end procedure

BA-UCT

One simulation in the tree search:

- Simulate given starting state node and fixed dynamics

- If state node is unexplored:
    - Init counters
    - Rollout to get R sample
    - Update N, Q=R
    - Return R

- Else:
    - Select action node with UCB rule
    - Sample next state

procedure `Simulate`($\langle s, h\rangle, \mathcal{P}, d$)
  if $\gamma^d R_{\max} < \epsilon$ **then return** 0
  if $N(\langle s, h\rangle) = 0$ **then**
    for all $a \in A$ **do**
      $N(\langle s, h\rangle, a) \leftarrow 0,$
      $Q(\langle s, h\rangle, a)) \leftarrow 0$
    end
    $a \sim \pi_{ro}(\langle s, h\rangle, \cdot)$
    $s' \sim \mathcal{P}(s, a, \cdot)$
    $r \leftarrow \mathcal{R}(s, a)$
    $R \leftarrow r + \gamma$ `Rollout`$(\langle s', has'\rangle, \mathcal{P}, d)$
    $N(\langle s, h\rangle) \leftarrow 1,\ N(\langle s, h\rangle, a) \leftarrow 1$
    $Q(\langle s, h\rangle, a) \leftarrow R$
    return R
  end
  $a \leftarrow \underset{b}{\operatorname{argmax}}\, Q(\langle s, h\rangle, b) + c\sqrt{\frac{\log(N(\langle s,h\rangle))}{N(\langle s,h\rangle,b)}}$
  $s' \sim \mathcal{P}(s, a, \cdot)$
  $r \leftarrow \mathcal{R}(s, a)$
  $R \leftarrow r + \gamma$ `Simulate`$(\langle s', has'\rangle, \mathcal{P}, d{+}1)$
  $N(\langle s, h\rangle) \leftarrow N(\langle s, h\rangle) + 1$
  $N(\langle s, h\rangle, a) \leftarrow N(\langle s, h\rangle, a) + 1$
  $Q(\langle s, h\rangle, a) \leftarrow Q(\langle s, h\rangle, a) + \frac{R - Q(\langle s,h\rangle,a)}{N(\langle s,h\rangle,a)}$
  return R
end procedure

BA-UCT

One simulation in the tree search:

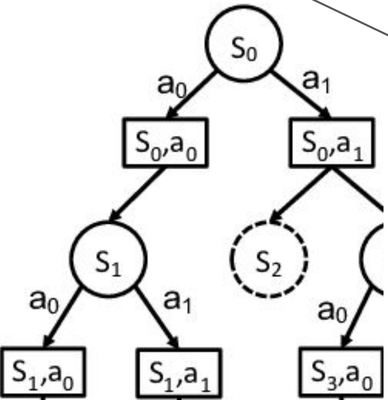
- Simulate given starting state node and fixed dynamics

- If state node is unexplored:
  - Init counters
  - Rollout to get R sample
  - Update N, Q=R
  - Return R

- Else:
  - Select action node with UCB rule
  - Sample next state
  - Simulate next state (recursive, will end in rollout) -> R
  - Update N, Q as sample average
  - Return R

Expand state leaf node

Traverse down tree



procedure Simulate($\langle s,h\rangle, \mathcal{P}, d$)

  if $\gamma^d R_{\max} < \epsilon$ **then** **return** 0

  if $N(\langle s,h\rangle) = 0$ **then**

    for all $a \in A$ **do**

      $N(\langle s,h\rangle, a) \leftarrow 0,$

      $Q(\langle s,h\rangle, a)) \leftarrow 0$

    end

    $a \sim \pi_{ro}(\langle s,h\rangle, \cdot)$

    $s' \sim \mathcal{P}(s,a,\cdot)$

    $r \leftarrow \mathcal{R}(s,a)$

    $R \leftarrow r + \gamma$ Rollout$(\langle s', has'\rangle, \mathcal{P}, d)$

    $N(\langle s,h\rangle) \leftarrow 1,\ N(\langle s,h\rangle, a) \leftarrow 1$

    $Q(\langle s,h\rangle, a) \leftarrow R$

    return R

  end

  $a \leftarrow \operatorname*{argmax}_b Q(\langle s,h\rangle, b) + c\sqrt{\frac{\log(N(\langle s,h\rangle))}{N(\langle s,h\rangle, b)}}$

  $s' \sim \mathcal{P}(s,a,\cdot)$

  $r \leftarrow \mathcal{R}(s,a)$

  $R \leftarrow r + \gamma$ Simulate$(\langle s', has'\rangle, \mathcal{P}, d{+}1)$

  $N(\langle s,h\rangle) \leftarrow N(\langle s,h\rangle) + 1$

  $N(\langle s,h\rangle, a) \leftarrow N(\langle s,h\rangle, a) + 1$

  $Q(\langle s,h\rangle, a) \leftarrow Q(\langle s,h\rangle, a) + \frac{R - Q(\langle s,h\rangle, a)}{N(\langle s,h\rangle, a)}$

  return R

end procedure

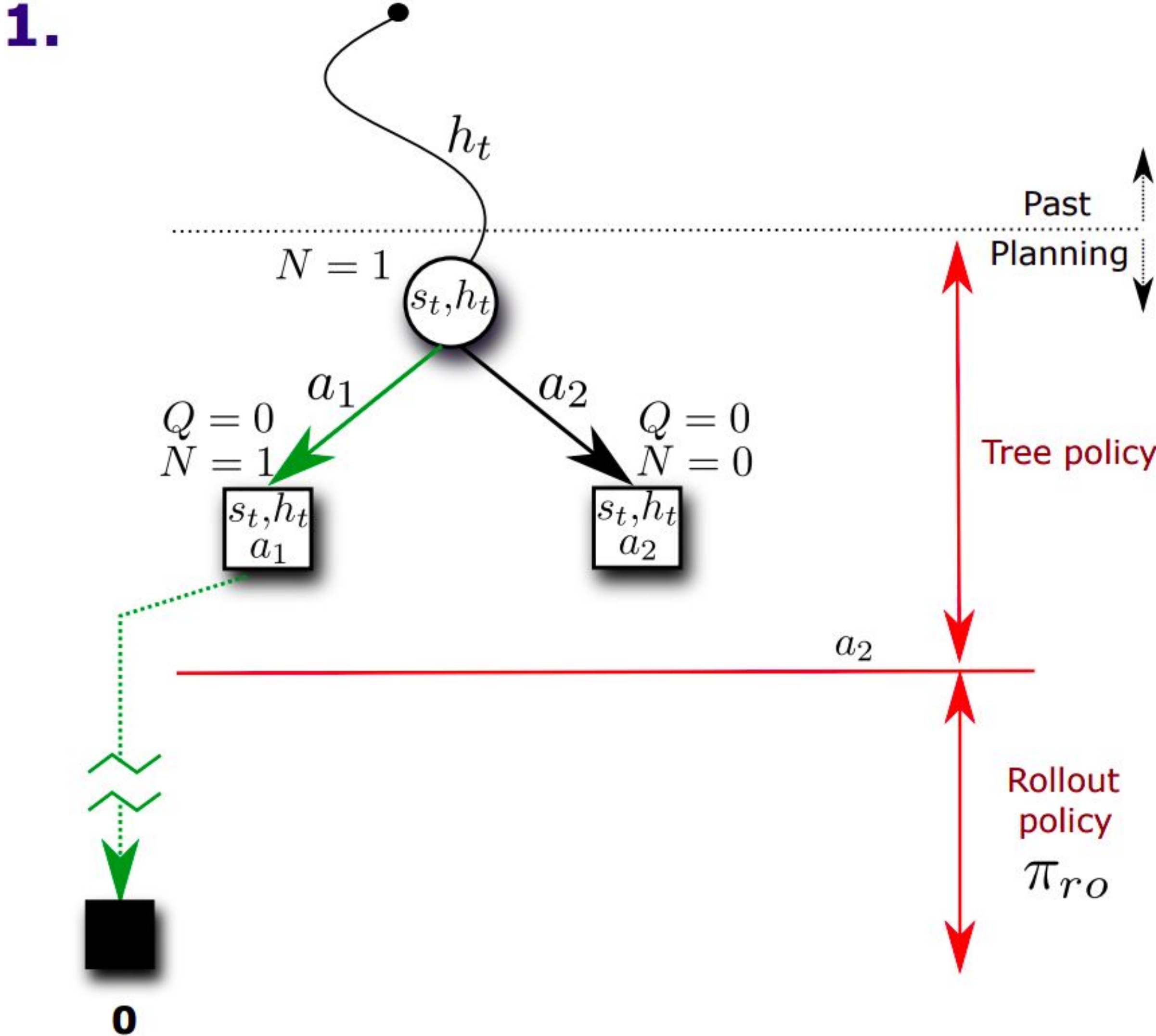
Example

Start at root with P1 sample

Chose a1 at root

Rollout a1 -> R=0

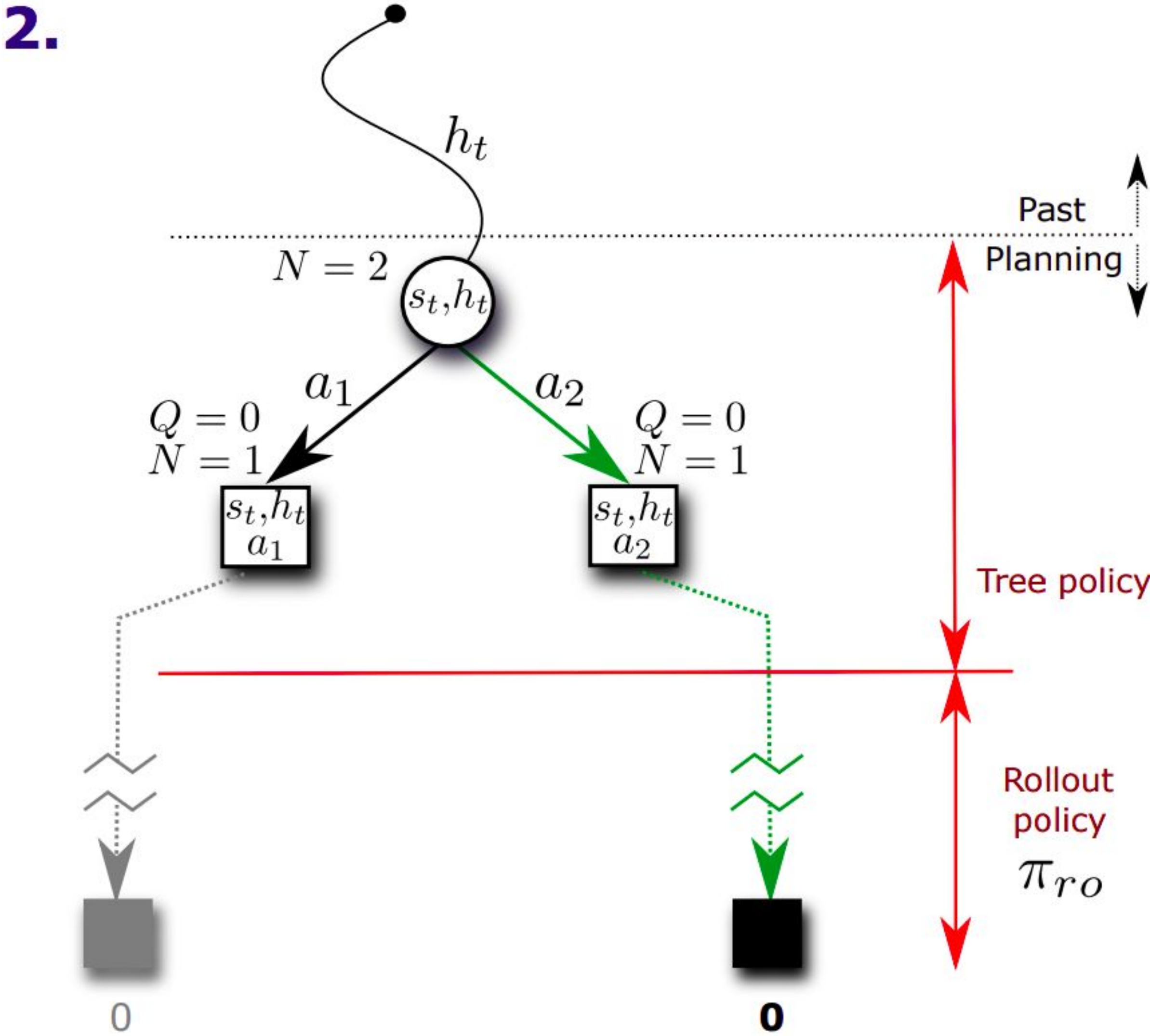
Q = R = 0



Start at root with P2 sample

Chose a2, since a1 already explored

Same thing happens (slight inconsistency with algorithm)



$$\text{argmax}_a\, Q(\langle s,h\rangle, a) + c\sqrt{\log(N(\langle s,h\rangle))/N(\langle s,h\rangle,a)}$$

Start at root with P3 sample

Chose a1 again

This time already visited,

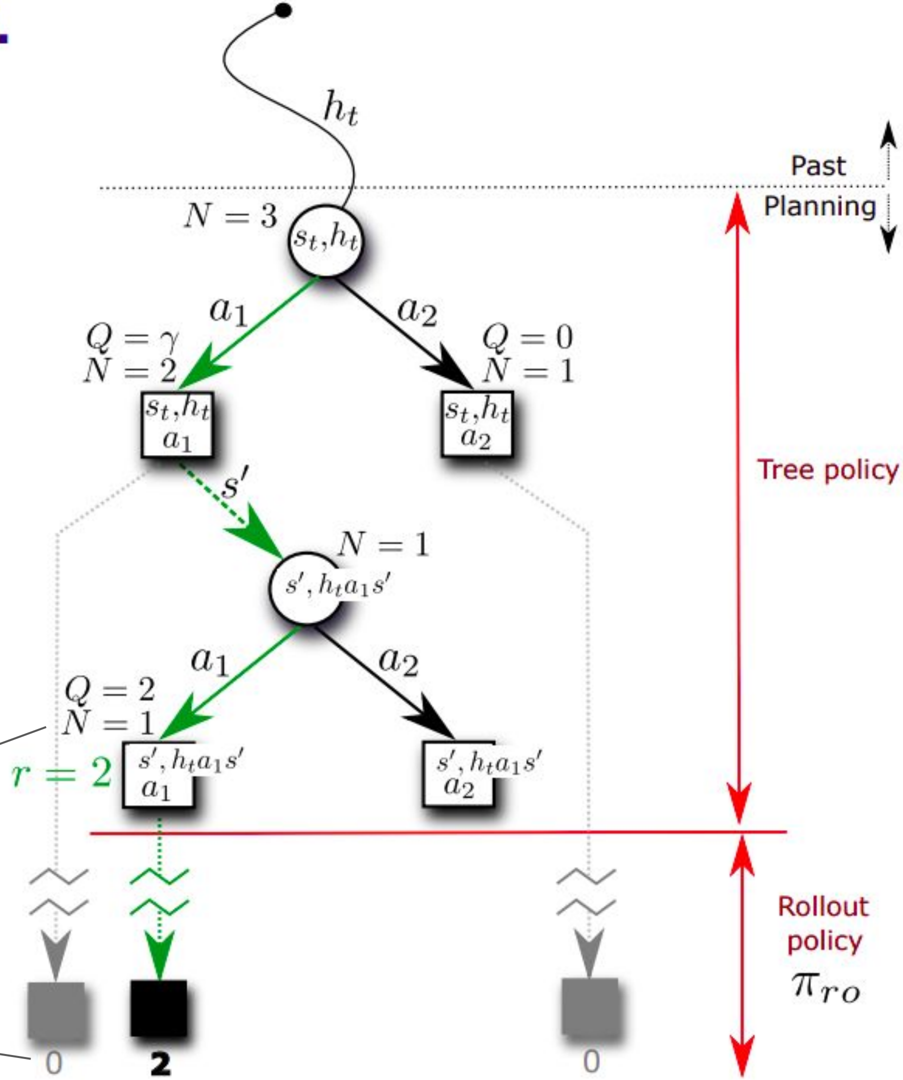
So sample next state s' and simulate it

Rollout from s' chose a1 got R=2

Q(s',a1) = R = 2

Q(s,a1) = 0.5 * (0 + Ɣ2)

3.



BA-UCT

One simulation in the tree search:

- Simulate given starting state node and fixed dynamics
- If state node is unexplored:
  - Init counters
  - Rollout to get R sample
  - Update N, Q=R
  - Return R
- Else:
  - Select action node with UCB rule
  - Sample next state
  - Simulate next state (recursive, will end in rollout) -> R
  - Update N, Q as sample average
  - Return R

$$P(\mathcal{P}|h_t) \propto P(h_t|\mathcal{P})P(\mathcal{P})$$

$$\int_{\mathcal{P}} \mathcal{P}(s,a,s')P(\mathcal{P}|h)\,\mathrm{d}\mathcal{P}$$

Recompute the marginal posterior dynamics?

procedure Simulate($\langle s,h\rangle, \mathcal{P}, d$)
  if $\gamma^d R_{\max} < \epsilon$ **then return** 0
  if $N(\langle s,h\rangle) = 0$ **then**
    for all $a \in A$ **do**
      $N(\langle s,h\rangle,a) \leftarrow 0$,
      $Q(\langle s,h\rangle,a)) \leftarrow 0$
    end
    $a \sim \pi_{ro}(\langle s,h\rangle,\cdot)$
    $s' \sim \mathcal{P}(s,a,\cdot)$
    $r \leftarrow \mathcal{R}(s,a)$
    $R \leftarrow r + \gamma\,\texttt{Rollout}(\langle s', has'\rangle, \boxed{\mathcal{P}}, d)$
    $N(\langle s,h\rangle) \leftarrow 1,\ N(\langle s,h\rangle,a) \leftarrow 1$
    $Q(\langle s,h\rangle,a) \leftarrow R$
    return R
  end
  $a \leftarrow \operatorname*{argmax}_b Q(\langle s,h\rangle,b) + c\sqrt{\frac{\log(N(\langle s,h\rangle))}{N(\langle s,h\rangle,b)}}$
  $s' \sim \mathcal{P}(s,a,\cdot)$
  $r \leftarrow \mathcal{R}(s,a)$
  $R \leftarrow r + \gamma\,\texttt{Simulate}(\langle s', has'\rangle, \boxed{\mathcal{P}}, d{+}1)$
  $N(\langle s,h\rangle) \leftarrow N(\langle s,h\rangle) + 1$
  $N(\langle s,h\rangle,a) \leftarrow N(\langle s,h\rangle,a) + 1$
  $Q(\langle s,h\rangle,a) \leftarrow Q(\langle s,h\rangle,a) + \frac{R - Q(\langle s,h\rangle,a)}{N(\langle s,h\rangle,a)}$
  return R
end procedure

Root Sampling

Sampling true $\mathcal{P}^+(\langle s, h\rangle, a, \langle s', h'\rangle) = \mathbb{1}[h' = has'] \int_{\mathcal{P}} \mathcal{P}(s, a, s') P(\mathcal{P}|h) \, \mathrm{d}\mathcal{P}$

at every step is intractable.

Instead, sample $\mathcal{P} \sim P(\mathcal{P}|h)$ once at root per simulation.

Distribution of histories equivalent w/ or w/o root sampling.
Intuitively, dynamics are filtered down paths that fit them.

Root Sampling - Not that bad

$\mathcal{P} \sim P(\mathcal{P}|h)$

Define $V(\langle s,h\rangle) = \max_{a\in A} Q(\langle s,h\rangle, a) \;\; \forall \langle s,h\rangle \in S \times \mathcal{H}$.

Theorem 1. *For all $\epsilon > 0$ (the numerical precision, see Algorithm 1) and a suitably chosen c (e.g. $c > \frac{Rmax}{1-\gamma}$), from state $\langle s_t, h_t\rangle$, BAMCP constructs a value function at the root node that converges in probability to an ϵ'-optimal value function, $V(\langle s_t,h_t\rangle) \xrightarrow{p} V^*_{\epsilon'}(\langle s_t,h_t\rangle)$, where $\epsilon' = \frac{\epsilon}{1-\gamma}$. Moreover, for large enough $N(\langle s_t,h_t\rangle)$, the bias of $V(\langle s_t,h_t\rangle)$ decreases as $O(\log(N(\langle s_t,h_t\rangle))/N(\langle s_t,h_t\rangle))$. (Proof available in supplementary material)*

Converges asymptotically to bayes-optimal policy.

Root Sampling

$$\mathcal{P} \sim P(\mathcal{P}|h)$$

Lemma 1 $\mathcal{D}^{\pi}(h_T) = \tilde{\mathcal{D}}^{\pi}(h_T)$ *for all EE policies* $\pi : \mathcal{H} \rightarrow A$.

Distribution of histories equivalent w/ or w/o root sampling.

Key insight: prob(P|h) is prop. to prob(P ends up at node h)

$$\begin{aligned}
P(\mathcal{P}\,|has') &= P(has'|\,\mathcal{P})P(\mathcal{P})/P(has') \\
&= P(h|\,\mathcal{P})P(\mathcal{P})\,\mathcal{P}(s,a,s')/P(has') \\
&= P(\mathcal{P}\,|h)P(h)\,\mathcal{P}(s,a,s')/P(has') \\
&\propto P(\mathcal{P}\,|h)\,\mathcal{P}(s,a,s') \\
&= \tilde{P}_h(\mathcal{P})\,\mathcal{P}(s,a,s') \\
&= \tilde{P}_{ha}(\mathcal{P})\,\mathcal{P}(s,a,s') \\
&= \tilde{P}_{has'}(\mathcal{P}),
\end{aligned}$$

Intuitively, dynamics are filtered down paths that fit them.

Rest of BAMCP

Root sampling

$$
\begin{array}{l}
\textbf{procedure } \texttt{Search}(\ \langle s,h\rangle\ ) \\
\quad \textbf{repeat} \\
\quad\quad \mathcal{P} \sim P(\mathcal{P}|h) \\
\quad\quad \texttt{Simulate}(\langle s,h\rangle, \mathcal{P}, 0) \\
\quad \textbf{until } \texttt{Timeout}() \\
\quad \textbf{return } \underset{a}{\operatorname{argmax}}\ Q(\langle s,h\rangle, a) \\
\textbf{end procedure}
\end{array}
$$

BAMCP:

1. Search -> a
2. Execute a in MDP
3. Add transition to h
4. Repeat

$$
\begin{array}{l}
\textbf{procedure } \texttt{Rollout}(\langle s,h\rangle, \mathcal{P}, d\ ) \\
\quad \textbf{if } \gamma^d R_{\max} < \epsilon \textbf{ then} \\
\quad\quad \textbf{return } 0 \\
\quad \textbf{end} \\
\quad a \sim \pi_{ro}(\langle s,h\rangle, \cdot) \\
\quad s' \sim \mathcal{P}(s,a,\cdot) \\
\quad r \leftarrow \mathcal{R}(s,a) \\
\quad \textbf{return} \\
\quad r + \gamma \texttt{Rollout}(\langle s', has'\rangle, \mathcal{P}, d+1) \\
\textbf{end procedure}
\end{array}
$$

Lazy Sampling

- Simple Idea: If dynamics parameterization factorized, only sample factors autoregressively as they are needed.

$$\theta_{s,a}$$

$$P(\Theta|h) = \int_\phi P(\Theta|\phi, h)P(\phi|h)$$

$$\begin{aligned} P(\Theta|\phi, h) = & P(\theta_{s_1,a_1}|\phi, h) \\ & P(\theta_{s_2,a_2}|\Theta_1, \phi, h) \\ & \vdots \\ & P(\theta_{s_T,a_T}|\Theta_{T-1}, \phi, h) \\ & P(\Theta \setminus \Theta_T|\Theta_T, \phi, h) \end{aligned}$$

Imagine infinite grid world!

?	?	?	?
?	?	v	?
?	o	<	?
?	?	?	?

Rollout Policy Learning

- Simple Idea: Train the rollout policy through model-free Q-learning with the true samples from the real MDP.

(s_t, a_t, r_t, s_{t+1}) observed,

$$Q_{ro}(s_t, a_t) \leftarrow Q_{ro}(s_t, a_t) + \alpha(r_t + \gamma \max_a Q_{ro}(s_{t+1}, a) - Q_{ro}(s_t, a_t)),$$

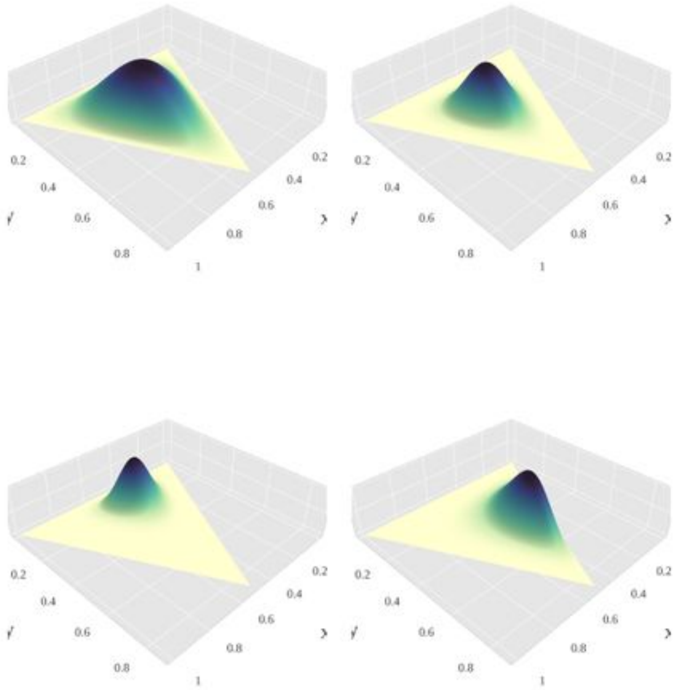
Epsilon-greedy rollout policy

$$\pi_{ro}(s, a) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A|} & \text{if } a = \operatorname{argmax}_{a'} Q_{ro}(s, a') \\ \frac{\epsilon}{|A|} & \text{otherwise,} \end{cases}$$

Experiments

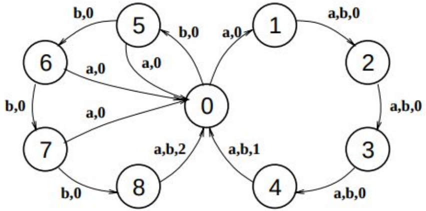
- Double-loop: |S|=9
- Grid5: |S|=5x5
- Grid10: |S|=10x10
- Maze: |S|=264

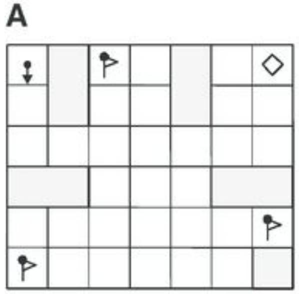
- Infinite Grid: |S| infinite
  - (R unknown)



Dirichlet-Multinomial Prior

Baselines





BFS3: Similar to BA-UCT but doesn't use MC rollouts

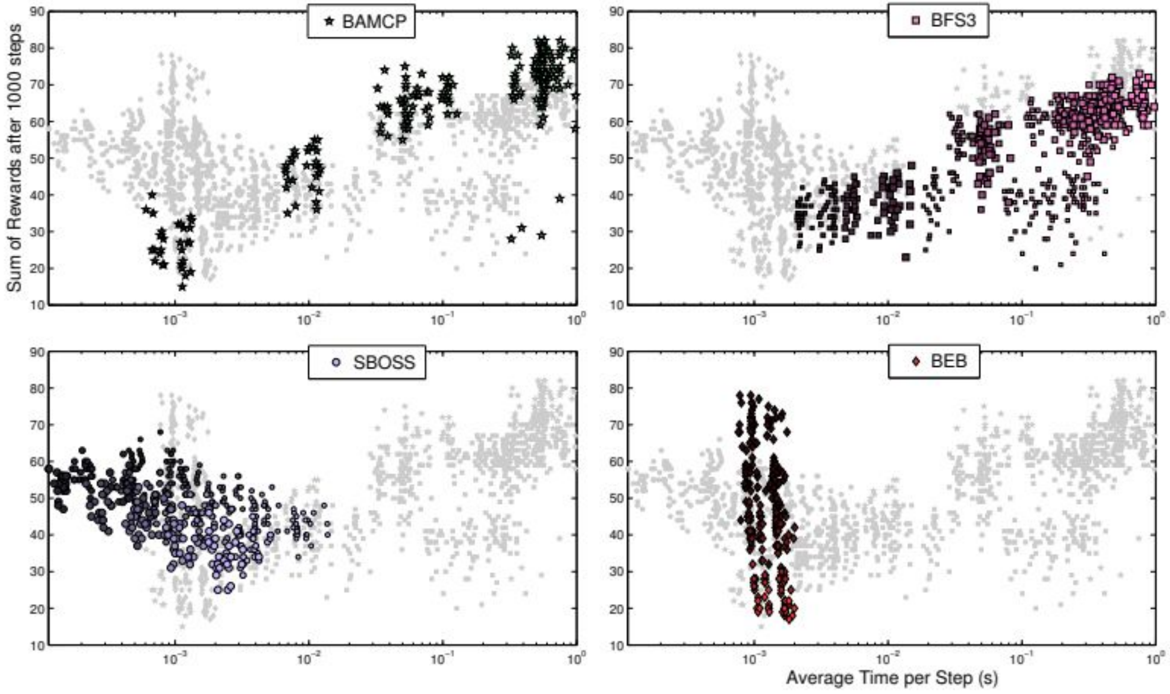
SBOSS: Sample K times from posterior and plan in averaged MDP

BEB: Plan with posterior mean + exploration bonus

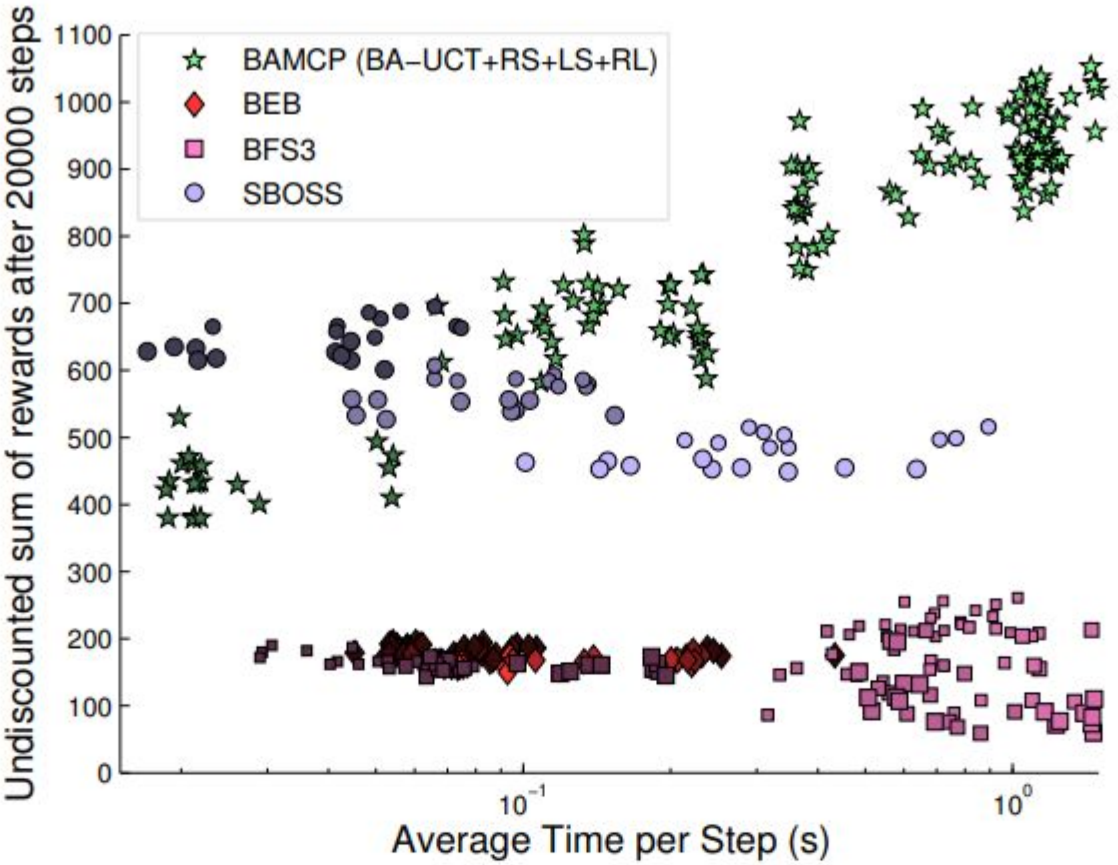
	Double-loop	Grid5	Grid10	Dearden's Maze
BAMCP	**387.6 ± 1.5**	**72.9 ± 3**	**32.7 ± 3**	**965.2 ± 73**
BFS3 [2]	382.2 ± 1.5	66 ± 5	10.4 ± 2	240.9 ± 46
SBOSS [5]	371.5 ± 3	59.3 ± 4	21.8 ± 2	671.3 ± 126
BEB [17]	386 ± 0	67.5 ± 3	10 ± 1	184.6 ± 35
Bayesian DP* [22]	377 ± 1	-	-	-
Bayes VPI+MIX* [8]	326 ± 31	-	-	817.6 ± 29
IEQL+* [19]	264 ± 1	-	-	269.4 ± 1
QL Boltzmann*	186 ± 1	-	-	195.2 ± 20

Bayesian Q-learning approaches (model-free)

Grid 5


BAMCP
BFS3
SBOSS
BEB
Sum of Rewards after 1000 steps
Average Time per Step (s)

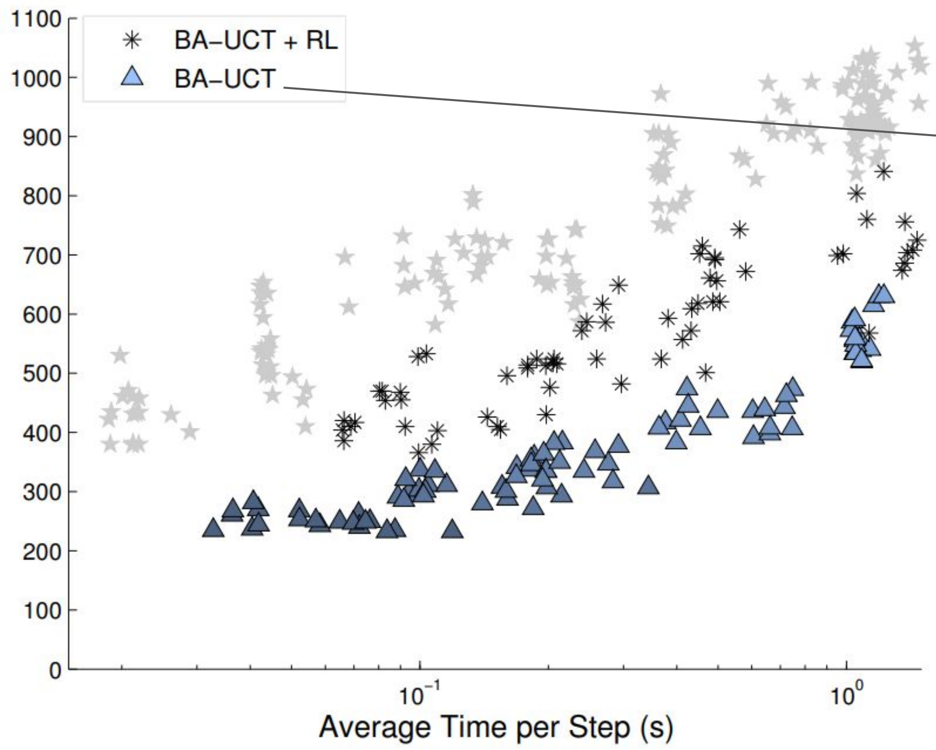
Maze

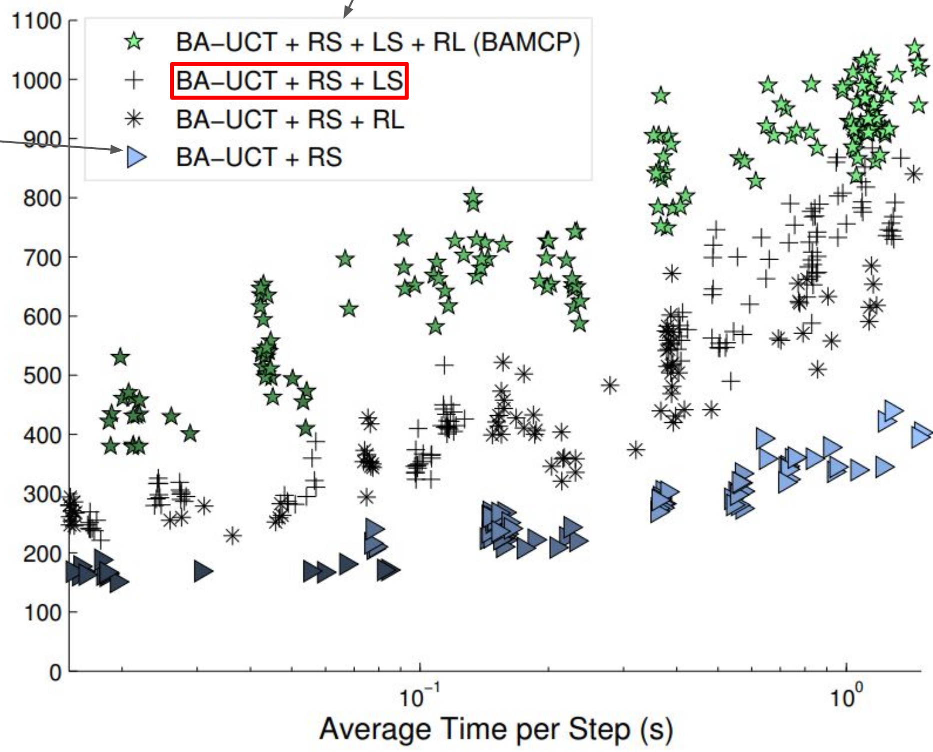


BAMCP (BA–UCT+RS+LS+RL)
BEB
BFS3
SBOSS
Undiscounted sum of rewards after 20000 steps
1100
1000
900
800
700
600
500
400
300
200
100
0
10^{-1}
10^{0}
Average Time per Step (s)

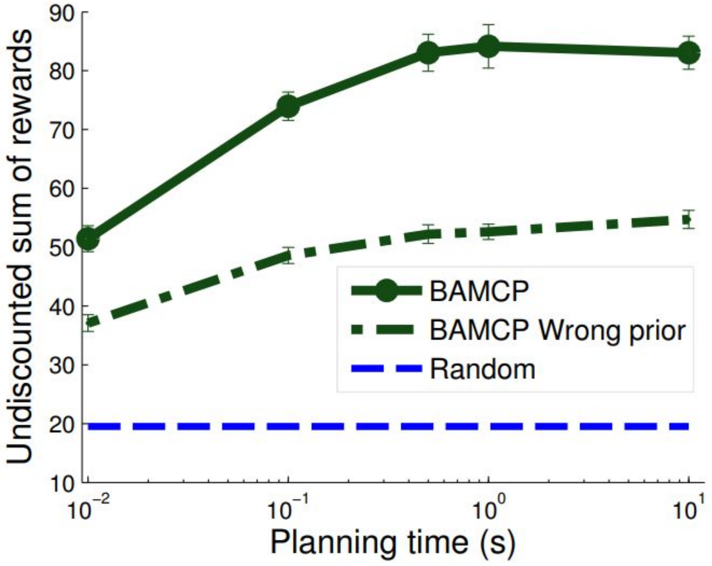
Ablation (Maze)

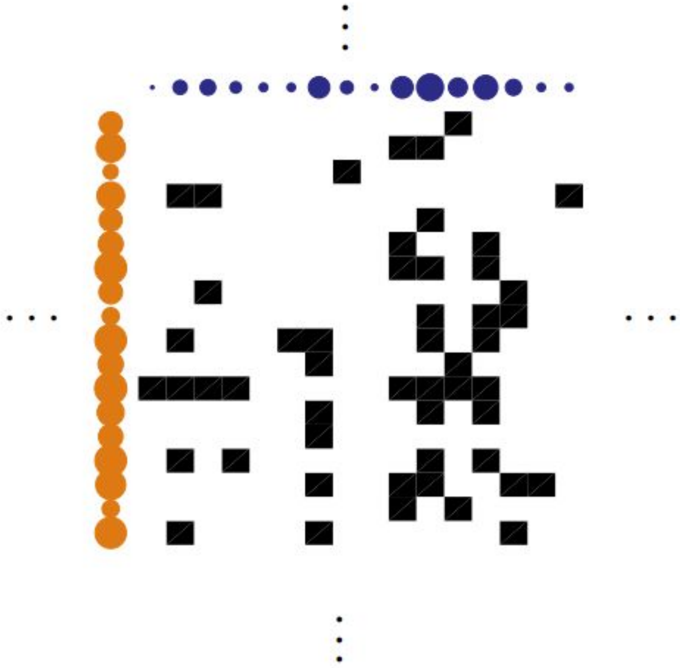
RS does worse on wall-clock?
To be fair, LS only possible when using RS.





Infinite Grid World (Requires LS)

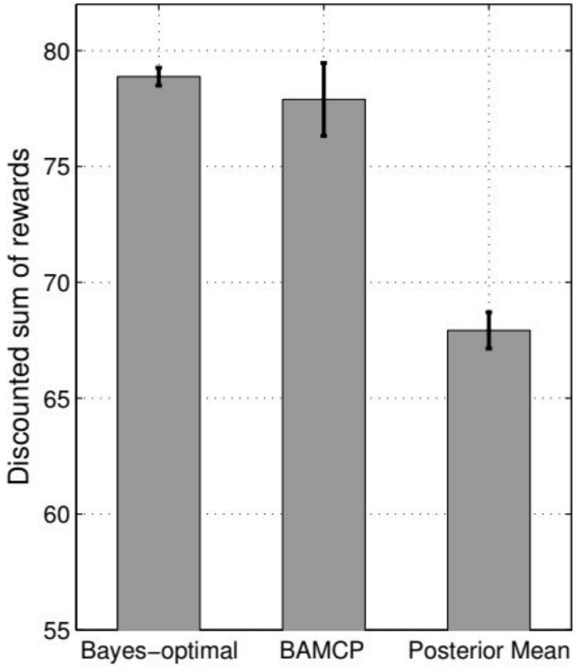




Probability of reward for each row and column sampled separately.
Exact inference not possible, uses MCMC.
Infinite state space -> Lazy sampling is a must

BAMCP vs Bayes-optimal on Bandit Problems

- 8-armed bernoulli bandits
- No “dynamics”
- Just posterior over return probabilities
- BAMCP converges to Bayes-optimal
- Using posterior mean does not



(b)

Limitations

- MCTS still restricted to discrete A and S
- Root sampling converges asymptotically but seems to hurt on wall clock (without adding LS)
- Still requires efficient posterior inference (every single simulate step)

Conclusion

1. Bayes-optimal EE policy: 1 way to formalize “optimal” exploration
    a. when dynamics unknown
2. Can compute 1 by solving the BAMDP
    a. which is a MDP with known dynamics
3. BAMDP rollouts expensive
    a. b/c history explosion + marginal posterior
4. Efficient search with MCTS + Root Sampling
    a. + Lazy Sampling and Rollout Policy Learning
5. Root sampling converges asymptotically to bayes-optimal

Questions to think about

- What's the point of tracking the visitation counts? $N(\langle s, h \rangle, a)$
- Why would BAMDP encourage policy to explore?
- What does root sampling do and why did we need it?
- When does lazy sampling hurt efficiency of BAMCP?

Root Sampling - Not that bad

$\mathcal{P} \sim P(\mathcal{P}|h)$

Define $V(\langle s, h\rangle) = \max_{a \in A} Q(\langle s, h\rangle, a) \ \forall \langle s, h\rangle \in S \times \mathcal{H}$.

Theorem 1. *For all $\epsilon > 0$ (the numerical precision, see Algorithm 1) and a suitably chosen c (e.g. $c > \frac{Rmax}{1-\gamma}$), from state $\langle s_t, h_t\rangle$, BAMCP constructs a value function at the root node that converges in probability to an ϵ'-optimal value function, $V(\langle s_t, h_t\rangle) \xrightarrow{p} V^*_{\epsilon'}(\langle s_t, h_t\rangle)$, where $\epsilon' = \frac{\epsilon}{1-\gamma}$. Moreover, for large enough $N(\langle s_t, h_t\rangle)$, the bias of $V(\langle s_t, h_t\rangle)$ decreases as $O(\log(N(\langle s_t, h_t\rangle))/N(\langle s_t, h_t\rangle))$. (Proof available in supplementary material)*

Converges asymptotically to bayes-optimal policy.

Theorem 6 *Consider a finite-horizon MDP with rewards scaled to lie in the $[0, 1]$ interval. Let the horizon of the MDP be D, and the number of actions per state be K. Consider algorithm UCT such that the bias terms of UCB1 are multiplied by D. Then the bias of the estimated expected payoff, $\overline{X}_n$, is $O(\log(n)/n)$. Further, the failure probability at the root converges to zero at a polynomial rate as the number of episodes grows to infinity.*

Start at root with P4 sample.

Chose a1 at root b/c higher Q.

Happens to land in same state s’.

Chose a2 at s’ due to N=0.

Rollout get’s R = 2ɣ^2.

...