Learning is Planning
Near Bayes-optimal RL via MCTS
by Asmuth & Littman

Ben Haanstra

November 2, 2012
Points of Interest

- Exploitation vs. Exploration Dilemma
- Bayes-optimal action selection
  *approximate and near-optimal*
- Monte Carlo Tree Search
- Strengths and Weaknesses
- Future Work
Preliminaries
Learning is Planning Paper

MDPs

- Have some problem \( M = \langle S, A, T, R, \gamma \rangle \)
- Find a policy \( \pi \) that scores high on objective \( J \)

\[
J(\pi, M) \triangleq E \left\{ \sum_{t=0}^{T} \gamma^{t} r_{t} \right\} \quad M, s = s_{t}, a_{t} \sim \pi
\]

- A Q-function, \( Q : S \times A \rightarrow \mathbb{R} \), for \( \pi \) on \( M \)

\[
Q_{\pi}^{M}(s, a) = \sum_{s'} T_{s,a,s'}^{M} [ R_{s,a,s'}^{M} + \gamma V_{\pi}^{M}(s', a') ]
\]

- With optimal \( \pi^* \)

\[
Q_{\pi^*}^{M}(s, a) = Q_{*}^{M}(s, a) = \sum_{s'} T_{s,a,s'}^{M} [ R_{s,a,s'}^{M} + \gamma \max_{a'} Q_{*}^{M}(s', a') ]
\]
Reinforcement Learning

- To optimize $J(\pi, M)$ we need both $\pi$ and $M$.

- Challenge: don’t know the true $M_*$ (hence we don’t know $\pi_*$)

- Approach: Reinforcement Learning
  We learn from interaction with $M_*$
  - Model-free: Q-learning, SARSA
  - Model-based: Dyna-Q, R-max

- Contrast this with Planning: $M_*$ is given!
  E.g. Value iteration
Typically, myopic, heuristic and semi-random ASMs are used when learning $Q_*$. They intent to ensure ‘sufficient’ exploration.

Hence, performance during learning is often poor.

Preferably, we wish to only select actions that are good or will lead to good actions (i.e. are informative).

This means that we need to properly balance explorative and exploitative actions.
Naturally, $M_\star \in \mathcal{M}$, the space of MDPs.

Idea: expand our MDP by incorporating the interaction history $h$ in the state, $S^+ = S \times H$. This $h$ corresponds to our knowledge of $M_\star$.

For hypothetical histories, we can sample interactions from $M \sim \mathcal{M}$. Insight: actions implicitly provide us information!

Using some prior $\phi : \mathcal{M} \times H \to \mathbb{R}$, we weigh $M$:

$$Q_\star(h, s, a) = \int_{M \in \mathcal{M}} \phi(M|h) \left[ \sum_{s'} T^M_{s,a,s'} \left[ R^M_{s,a,s'} + \gamma \max_{a'} Q_\star(\langle h \cup (s, a, r, s') \rangle, s', a') \right] \right] dM$$
Both $\phi$ and $M$ are 'known'.

This allows us to plan $Q_*$ in this augmented MDP.

Hence, "learning is planning"
Keeping track of histories is tedious.

Idea: introduce a statistic that replaces them. I.e. a prior probability distribution $b$ over $M$.

$\phi(M|h)$ then corresponds with the posterior $b(M|h)$:

$$b(M|h) = \eta b(h|M)b(M)$$

With a convenient prior (FDM), updating $b$ is easy:

$$b_{t+1} = \tau(b_t, \langle s, a, r, s' \rangle)$$

Optimal Q-function, where $o = \langle s, a, r, s' \rangle$:

$$Q_*(b, s, a) = \int b(M) \sum_{s'} T^M_{s,a,s'} \left[ R^M_{s,a,s'} + \gamma \max_{a'} Q_*(\tau(b, o), s', a') \right] dM$$
Approximate and near Bayesian RL

- Exact Bayesian RL is generally intractable

- Approximate Bayes-optimality: 
  Estimated $Q(b, s, a)$ is within $\epsilon$ of true value
  Algorithms: BEETLE

- Near Bayes-optimal: 
  Approx BO for all but a small number of steps
  Algorithms: BEB, VBR, BFS3, BCMP

- Other approaches: 
  VPI, Thompson sampling/Bayesian DP, BOSS, SBOSS
Monte Carlo Tree Search

- Generate trajectories using a (generative) model
- Action-values are estimated by evaluating trajectories
- Many ideas: Sparse Sampling, UCT, Bayesian Sparse Sampling, Forward Sparse Sampling, and now BFS3
### Algoritms

**Input:** state $s$, max depth $d$, #trajectories $t$, MDP $M$  
**Output:** estimated value for state $s$  

for $t$ times do  
\[ FSSS-Rollout(s, d, 0, M) \]  
\[ \hat{V}(s) \leftarrow \max_a U_d(s, a) \]  
return $\hat{V}(s)$  

**Algorithm 1:** FSSS($s, d, t, M$)

---

**Input:** state $s$, history $h$, depth $d$, #trajectories $t$, prior $\phi$  
**Output:** action to take in state $s$  

if $\langle s, h \rangle \in$ solved-belief-states then  
\[ \text{return } \pi(\langle s, h \rangle) \]  
foreach $a \in A$ do  
\[ \text{for } C \text{ times do} \]  
\[ \langle s', h', r \rangle \sim T_{\phi}(\langle s, h, a \rangle) \]  
\[ q(a) \leftarrow q(a) + \frac{1}{C} [r + \gamma FSSS(\langle s', h', d, t, M \rangle)] \]  
solved-belief-states $\leftarrow$ solved-belief-states $\cup \{\langle s, h \rangle\}$  
\[ \pi(\langle s, h \rangle) \leftarrow \arg\max_a q(a) \]  
return $\pi(\langle s, h \rangle)$  

**Algorithm 3:** BFS3($s, h, d, t, \phi$)

---

**Input:** state $s$, max depth $d$, current depth $l$, MDP $M$  
if $Terminal(s)$ then  
\[ U_d(s) = L_d(s) = 0 \]  
return  
if $d = l$ then  
\[ U_d(s) \leftarrow U_d(s) + r/C \]  
foreach $a \in A$ do  
\[ \text{for } C \text{ times do} \]  
\[ s', r \sim T_{\phi}(s, a), R_{M}(s, a) \]  
\[ \text{Count}_d(s, a, s') \leftarrow \text{Count}_d(s, a, s') + 1 \]  
\[ \text{Children}_d(s, a) \leftarrow \text{Children}_d(s, a) \cup \{s'\} \]  
\[ R_d(s, a) \leftarrow R_d(s, a) + r/C \]  
if $\neg \text{Visited}_{d+1}(s')$ then  
\[ U_{d+1}(s'), L_{d+1}(s') = V_{\max}, V_{\min} \]  
Bellman-backup($s, d$)  
\[ a \leftarrow \arg\max_a U_d(s, a) \]  
\[ s' \leftarrow \arg\max_{s'} (U_{d+1}(s') - L_{d+1}(s')) \cdot \text{Count}_d(s, a, s') \]  
FSSS-Rollout($s', d, l + 1, M$)  
Bellman-backup($s, d$)  
return  

**Algorithm 2:** FSSS-Rollout($s, d, l, M$)
Strengths and Weaknesses

- What parts do you think are strong / weak?
- Do you have better / other ideas?
Future work

- General paths of improvement:
  1. Different priors
  2. Inference
  3. Sampling / Lazy sampling
  4. Search tree expansion

- More interesting: many $M \in \mathcal{M}$ share the same $\pi^*$. MBBRL implicitly accounts for this, but its approximations do so in a 'blind' manner.