Thompson Sampling based Monte-Carlo Planning in POMDPs

Aijun Bai[1] Feng Wu[2] Zongzhang Zhang[3] Xiaoping Chen[1]

[1]University of Science & Technology of China

[2]University of Southampton

[3]National University of Singapore

June 24, 2014

Table of Contents

Monte-Carlo tree search

- Online planning method
- Finds near-optimal policies for MDPs and POMDPs
- Builds a best-first search tree using Monte-Carlo samplings
- Without explicitly knowing the underlying models in advance

MCTS procedure

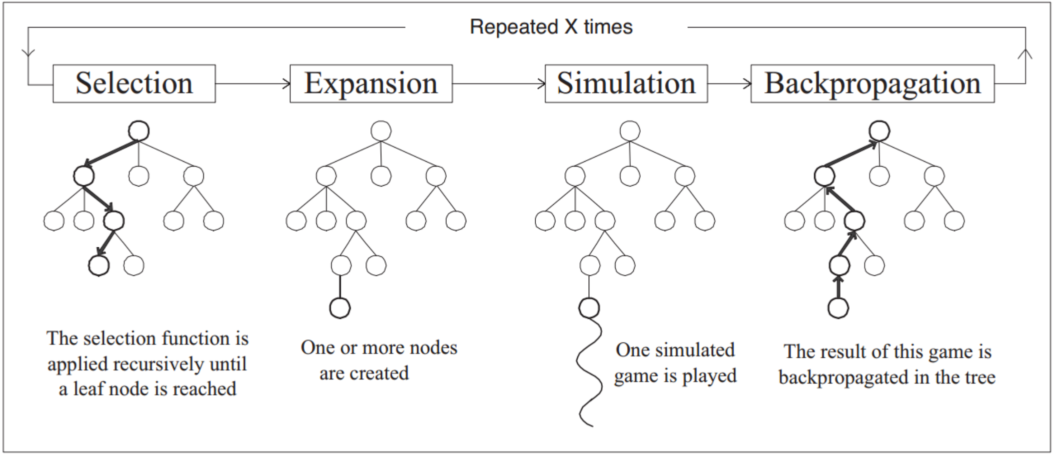


Figure 1 : Outline of Monte-Carlo tree search [Chaslot et al., 2008].

Resulting asymmetric search tree

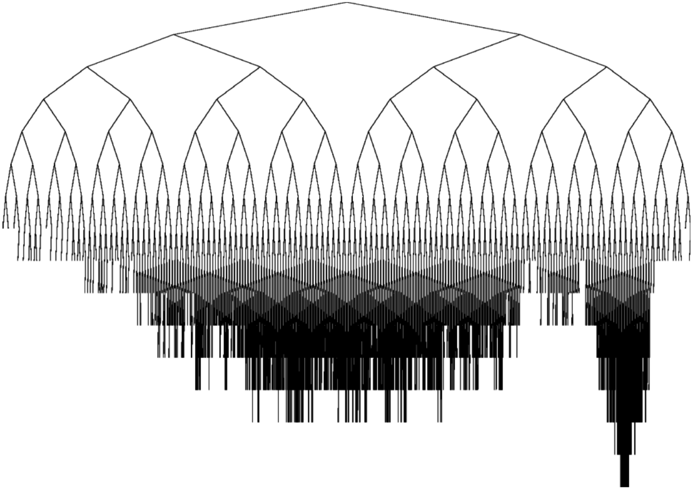


Figure 2 : An example of resulting asymmetric search tree [Coquelin and Munos, 2007].

The *exploration vs. exploitation* dilemma

- A fundamental problem in MCTS:
  1. Must not only exploit by selecting the action that currently seems best
  2. Should also keep exploring for possible higher future outcomes
- Can be seen as a multi-armed bandit problem (MAB)
  1. A set of actions: A
  2. An unknown stochastic reward function $R(a) := X_a$
- Cumulative regret (CR):

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T}(X_{a^*} - X_{a_t})\right] \tag{1}$$

- Minimize CR by trading off between exploration and exploitation

UCB1 heuristics

- POMCP algorithm [Silver and Veness, 2010]:

$$\mathrm{UCB1}(h,a) = \bar{Q}(h,a) + c\sqrt{\frac{\log N(h)}{N(h,a)}} \tag{2}$$

  - $\bar{Q}(h,a)$ is the mean outcome of applying action a in history h
  - $N(h,a)$ is the visitation count of action a following h
  - $N(h) = \sum_{a \in A} N(h,a)$ is the overall count
  - c is the exploration constant

Balancing between CR and SR in MCTS

- Simple regret (SR):

$$r_n = \mathbb{E}\left[X_{a^*} - X_{\bar{a}}\right] \tag{3}$$

  where $\bar{a} = \operatorname{argmax}_{a \in A} \bar{X}_a$
- Makes more sense for pure exploration
- A recently growing understanding: balance between CR and SR [Feldman and Domshlak, 2012]
  1. Does not collect a real reward when searching the tree
  2. Good to grow the tree more accurately by exploiting the current tree

Thompson sampling

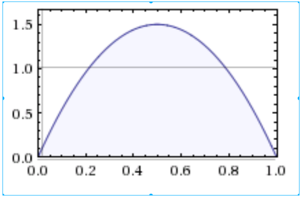
- Select an action based on its posterior probability of being optimal

$$P(a) = \int \mathbf{1}\left[a = \operatorname*{argmax}_{a'} \mathbb{E}\left[X_{a'} \mid \theta_{a'}\right]\right] \prod_{a'} P_{a'}(\theta_{a'} \mid Z)\,\mathrm{d}\boldsymbol{\theta} \tag{4}$$

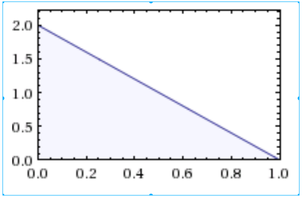
  1. θ_a specifies the unknown distribution of X_a
  2. $\boldsymbol{\theta} = (\theta_{a_1}, \theta_{a_2}, \dots)$ is a vector of all hidden parameters
- Can efficiently be approached by sampling method
  1. Sample a set of hidden parameters θ_a
  2. Select the action with highest expectation $\mathbb{E}\left[X_{a'} \mid \theta_{a'}\right]$

An example of Thompson sampling

- 2-armed bandit: a and b
- Bernoulli reward distributions
- Hidden parameters p_a and p_b
- Prior distributions:
  - $p_a \sim Uniform(0, 1)$
  - $p_b \sim Uniform(0, 1)$
- History: a, 1, b, 0, a, 0
- Posterior distributions:
  - $p_a \sim Beta(2, 2)$
  - $p_b \sim Beta(1, 2)$
- Sample p_a and p_b
- Compare $\mathbb{E}[X_a \mid p_a]$ and $\mathbb{E}[X_b \mid p_b]$



(a) $Beta(2, 2)$.



(b) $Beta(1, 2)$.

Figure 3 : Posterior distributions.

Motivation

- Thompson sampling
  1. Theoretically achieves asymptotic optimality for MABs in terms of CR
  2. Empirically has competitive and even better performance comparing with state-of-the-art in terms of CR and SR
- Seems to be a promising approach for the challenge of balancing CR and SR

Contribution

- A complete Bayesian approach for online Monte-Carlo planning in POMDPs
  1. Maintain the posterior reward distribution of applying an action
  2. Use Thompson sampling to guide the action selection

Bayesian modeling and inference

- $X_{b,a}$: the immediate reward of performing action a in belief b
- A finite set of possible immediate rewards: $\mathcal{I} = \{r_1, r_2, \ldots, r_k\}$
- $X_{b,a} \sim Multinomial(p_1, p_2, \ldots, p_k)$
  1. $p_i = \sum_{s \in S} \mathbf{1}[R(s,a) = r_i] b(s)$
  2. $\sum_{i=1}^{k} p_i = 1$
- $(p_1, p_2, \ldots, p_k) \sim Dirichlet(\boldsymbol{\psi}_{b,a})$, where $\boldsymbol{\psi}_{b,a} = (\psi_{b,a,r_1}, \psi_{b,a,r_2}, \ldots, \psi_{b,a,r_k})$
- Observing r:

$$\psi_{b,a,r} \leftarrow \psi_{b,a,r} + 1 \tag{5}$$

Bayesian modeling and inference

- $X_{s,b,\pi}$: the cumulative reward of following policy π in joint state $\langle s, b \rangle$
- $X_{s,b,\pi} \sim \mathcal{N}(\mu_{s,b}, 1/\tau_{s,b})$ (according to CLT on Markov chains)
- $(\mu_{s,b}, \tau_{s,b}) \sim NormalGamma(\mu_{s,b,0}, \lambda_{s,b}, \alpha_{s,b}, \beta_{s,b})$
- Observing v:

$$\mu_{s,b,0} = \frac{\lambda_{s,b}\mu_{s,b,0} + v}{\lambda_{s,b} + 1} \tag{6}$$

$$\lambda_{s,b} = \lambda_{s,b} + 1 \tag{7}$$

$$\alpha_{s,b} = \alpha_{s,b} + \frac{1}{2} \tag{8}$$

$$\beta_{s,b} = \beta_{s,b} + \frac{1}{2}\left(\frac{\lambda_{s,b}(v - \mu_{s,b,0})^2}{\lambda_{s,b} + 1}\right) \tag{9}$$

Bayesian modeling and inference

- $X_{b,\pi}$: the cumulative reward of following policy π in belief b
- $X_{b,\pi}$ follows a mixture of Normal distributions:

$$f_{X_{b,\pi}}(x) = \sum_{s \in S} b(s) f_{X_{s,b,\pi}}(x) \tag{10}$$

- $X_{b,a,\pi}$: the cumulative reward of applying a in belief b and following policy π

$$X_{b,a,\pi} = X_{b,a} + \gamma X_{b',\pi} \tag{11}$$

- Expectation of $X_{b,a,\pi}$:

$$\mathbb{E}[X_{b,a,\pi}] = \mathbb{E}[X_{b,a}] + \gamma \sum_{o \in O} \mathbf{1}[b' = \zeta(b,a,o)]\Omega(o \mid b,a)\mathbb{E}[X_{b',\pi}] \tag{12}$$

Bayesian modeling and inference

- $\Omega(\cdot \mid b, a) \sim Dirichlet(\boldsymbol{\rho}_{b,a})$
- $\boldsymbol{\rho}_{b,a} = (\rho_{b,a,o_1}, \rho_{b,a,o_2}, \dots)$
- Observing a transition $(b, a) \to o$:

$$\rho_{b,a,o} \leftarrow \rho_{b,a,o} + 1 \tag{13}$$

Thompson sampling based action selection

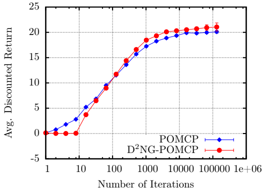
- Decision node with belief b
- Sample a set of parameters:
  1. $\{w_{b,a,o}\} \sim Dirichlet(\boldsymbol{\rho}_{b,a})$
  2. $\{w_{b,a,r}\} \sim Dirichlet(\boldsymbol{\psi}_{b,a})$
  3. $\{\mu_{s',b'}\} \sim NormalGamma(\mu_{s',b',0}, \lambda_{s',b'}, \alpha_{s',b'}, \beta_{s',b'})$, where $b' = \zeta(b, a, o)$
- Select action with highest expectation — sampled $\tilde{Q}$ value:

$$\tilde{Q}(b,a) = \sum_{r \in \mathcal{I}} w_{b,a,r} r + \gamma \sum_{o \in O} \mathbf{1}[b' = \zeta(b,a,o)] w_{b,a,o} \sum_{s' \in S} \mu_{s',b'} b'(s') \tag{14}$$

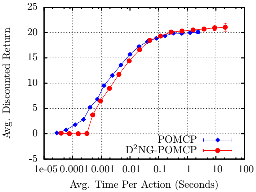
Experiments

- D^{2}NG-POMCP: Dirichlet-Dirichlet-NormalGamma partially observable Monte-Carlo planning
- *RockSample* and *PocMan* domains
- Evaluation:
  1. Run the algorithms for a number of iterations for current belief
  2. Apply the best action based on the resulting action-values
  3. Repeat until terminating conditions (goal state or maximal number of steps)
  4. Report the total discounted reward and the average time usage per action

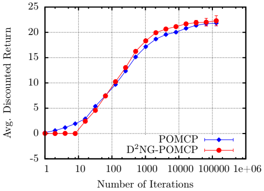
Experimental results



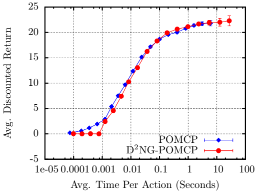
(a) RS[7, 8].



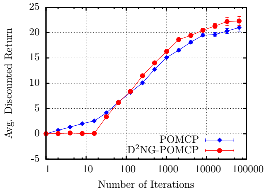
(b) RS[7, 8].



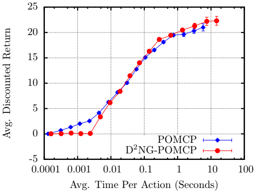
(c) RS[11, 11].



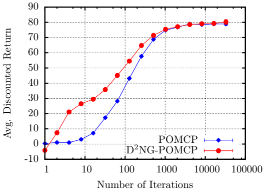
(d) RS[11, 11].



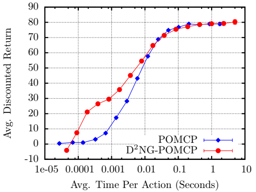
(e) RS[15, 15].



(f) RS[15, 15].



(g) PocMan.



(h) PocMan.

Figure 4 : Performance of D^{2}NG-POMCP in RockSample and PocMan

Discussion

- The total computation time is linear with the total number of simulations
- Require more computation time than POMCP, due to the time-consuming operations of sampling from various distributions
- Can obtain better performance in terms of computational complexity, if the simulations are expensive
- Expected to have lower sample complexity

Conclusion and future work

- A Bayesian MCTS algorithm: D^2NG-POMCP
  1. Maintain the posterior distribution of cumulative reward
  2. Select action using Thompson sampling
  3. Balance between CR and SR
- Future work
  1. Our assumptions of distributions in principle only hold in the limit
  2. Extend these assumptions to more realistic distributions
  3. Test our algorithm on real-world applications

References I

Chaslot, G., Bakkes, S., Szita, I. and Spronck, P. (2008).
Monte-Carlo Tree Search: A New Framework for Game AI.
In Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, October 22-24, 2008, Stanford, California, USA, (Darken, C. and Mateas, M., eds), The AAAI Press.

Coquelin, P.-A. and Munos, R. (2007).
Bandit algorithms for tree search.
In Uncertainty in Artificial Intelligence.

Feldman, Z. and Domshlak, C. (2012).
Simple regret optimization in online planning for Markov decision processes.
In AAAI Conference on Artificial Intelligence.

Silver, D. and Veness, J. (2010).
Monte-Carlo planning in large POMDPs.
In Advances in Neural Information Processing Systems pp. 2164–2172,.