Blazing the trails before beating the path: Sample-efficient Monte-Carlo planning

jean-bastien.grill@inria.fr, michal.valko@inria.fr, and munos@google.com

Inria informatics mathematics

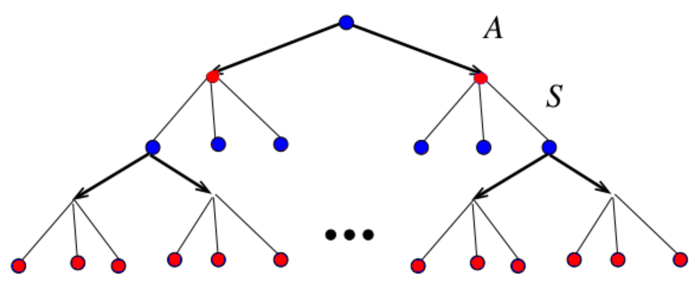
Google DeepMind

Setting

Goal: Local planning:

- Environment is **stochastic**.
- Planning is **local**: we only care about the best action for the state we are in rather than full policy.
- Contruct look-ahead tree to compute the best action.

We assume A actions and S next states.



$\mathcal{V}[s]$: rewards if you play optimally from s.

- `MAX` nodes: $\mathcal{V}[s] = \max_{s' \text{ child of } s} \mathcal{V}[s']$.
- `AVG` nodes: $\mathcal{V}[s] = r(s) + \gamma \sum_{s' \text{ child of } s} p(s'|s)\mathcal{V}[s']$.

- We have access to a **generative model:**

$$\text{average node } s \longrightarrow \begin{array}{ll} r_s & \text{reward sample s.t. } \mathbb{E}r = r(s) \\ y_s & \text{next state sample} \sim p(\cdot|s) \end{array}$$

- Don't access the full the transition or reward probability law.

- We want PAC guarantees.

For any $\delta > 0, \varepsilon > 0$, we compute $v(\delta, \varepsilon)$ such that

$$\mathbb{P}\left[|v(\delta,\varepsilon) - \mathcal{V}[s_0]| < \varepsilon\right] > 1 - \delta$$

Algorithm: Average Node

Input: m, ε
Initialization: {Only executed on first call}
SampledNodes $\leftarrow \emptyset$,
$r \leftarrow 0$
Run:
if $\varepsilon \geq 1/(1-\gamma)$ **then**
  Output: 0
end if
if $|\text{SampledNodes}| > m$ **then**
  ActiveNodes $\leftarrow$ SampledNodes$(1:m)$
else
  while $|\text{SampledNodes}| < m$ **do**
    $\tau \leftarrow$ {new sample of next state}
    SampledNodes.append(τ)
    $r \leftarrow r +$[new sample of reward]
  end while
  ActiveNodes $\leftarrow$ SampledNodes
end if {At this point, $|\text{ActiveNodes}| = m$}
for all unique nodes $s \in$ ActiveNodes **do**
  $k \leftarrow$ #occurrences of s in ActiveNodes
  $\nu \leftarrow$ call s with parameters $(k, \varepsilon/\gamma)$
  $\mu \leftarrow \mu + \nu k/m$
end for
Output: $\gamma\mu + r/|\text{SampledNodes}|$

Algorithm: Maximum Node

Input: m, ε
$\mathcal{L} \leftarrow$ all children of the node
$\ell \leftarrow 1$
while $|\mathcal{L}| > 1$ and $\mathtt{U} \geq (1-\eta)\varepsilon$ **do**
  $\mathtt{U} \leftarrow \frac{2}{1-\gamma}\sqrt{\frac{\log(K\ell/(\delta\varepsilon)) + \gamma/(\eta-\gamma) + \lambda + 1}{\ell}}$
  for $b \in \mathcal{L}$ **do**
    $\mu_b \leftarrow$ call b with $(\ell, \mathtt{U}\eta/(1-\eta))$
  end for
  $\mathcal{L} \leftarrow \left\{b : \mu_b + \frac{2\mathtt{U}}{1-\eta} \geq \sup_j \left[\mu_j - \frac{2\mathtt{U}}{1-\eta}\right]\right\}$
  $\ell \leftarrow \ell + 1$
end while
if $|\mathcal{L}| > 1$ **then**
  Output: $\mu \leftarrow \max_{b\in\mathcal{L}} \mu_b$
else $\{\, \mathcal{L} = \{b^\star\} \,\}$
  $b^\star \leftarrow \arg\max_{b\in\mathcal{L}} \mu_b$
  $\mu \leftarrow$ call $b^\star$ with $(m, \eta\varepsilon)$
  Output: μ
end if

Contributions

We provide an algorithm `TrailBlazer` with two **Sample complexity** bounds.

$n(\delta, \varepsilon)$: **Sample complexity**, the number of calls to the generative model.
S is the size of the MDP.

Theorem 1. *There exists $C > 0$ such that for all $\varepsilon > 0$ and $\delta > 0$, with probability $1-\delta$, the sample-complexity of `TrailBlazer` (the number of calls to the generative model before the algorithm terminates) is*

$$n(\varepsilon,\delta) \leq C(1/\varepsilon)^{\max\left(2, \frac{\log(N\kappa)}{\log(1/\gamma)} + o(1)\right)} \left(\log(1/\delta) + \log(1/\varepsilon)\right)^\alpha,$$

where $\alpha = 5$ when $\log(N\kappa)/\log(1/\gamma) \geq 2$ and $\alpha = 3$ otherwise.

- High probability sample complexity bound.
- The bound is polynomial in $1/\varepsilon$.
- The quantity $\kappa \in [1, A]$ is problem dependent. It measures the branching factor of the set of important states.
- This is small improvement over `StOP`
- It should be compared to the S-dependent sample complexity of uniform planning:

$$\mathcal{O}\left((1/\varepsilon)^{2+\frac{\log(AS)}{\log(1/\gamma)}}\right)$$

Theorem 2. *If d is finite then there exists $C > 0$ such that for all $\varepsilon > 0$ and $\delta > 0$ the expected sample complexity of `TrailBlazer` satisfies*

$$\mathbb{E}\left[n(\varepsilon,\delta)\right] \leq C\frac{\left(\log(1/\delta) + \log(1/\varepsilon)\right)^3}{\varepsilon^{2+d}}.$$

- This bound is independent with the size of the MDP.
- It holds in expectation only.
- It should be compared to the S-independent sample complexity of uniform planning:

$$\mathcal{O}\left((1/\varepsilon)^{\log(1/\varepsilon)/\log(1/\gamma)}\right)$$

- Like κ, the quantity d is problem dependent. Unlike κ, the quantity d may not exist.
- When d exists: first polynomial S independent bound.

Example for d=0

- The **gap** of a node is the difference in value between the best and second best action.
- Low gap refer to hard problems.
- $\Delta(s) := s \to$ gap of s' with probability $p(s'|s)$.
- The following assumption measure the number of low gap nodes.

Assumption 1. *$\exists a, b > 0$ s.t. for all average node s and $t > 0$*

$$\mathbb{P}\left[\Delta(s) < t\right] < at^{2+b}$$

Theorem 3. *Under Assumption 1, $d = 0$.*

When $d = 0$, the **Sample complexity** is of order $(1/\varepsilon)^2$ which is the same order as **Monte Carlo sampling**.

Key Ideas

- Tree-based algorithm
- Delicate treatment of uncertainty
- Refining few paths

Analysis

$\Delta_{\to s}(s')$: The difference of the sum of discounted rewards stating from s' between an agent playing optimally and one playing first the action toward s and then optimally.

Definition 1 (near-optimality). *We say that a node s of depth h is* near-optimal, *if for all $h' < h$*

$$\Delta_{\to s}(s_{h'}) \leq 12\frac{\gamma^{h-h'}}{1-\gamma} \quad \textit{or} \quad \textit{the action from } s_{h'} \textit{ to } s \textit{ is optimal}$$

with $s_{h'}$ the ancestor of s of depth h'. Let $\mathcal{N}_h$ be the set of all near-optimal nodes of depth h.

Definition 2. *We define $\kappa \in [1, K]$ as the smallest number such that*

$$\exists C\, \forall h, \quad |\mathcal{N}_h| \leq C(N\kappa)^h.$$

- There are at most $(AS)^h$ nodes of depth h thus $\kappa \leq A$.
- With probability $1-\delta$, `TrailBlazer` only explore near-optimal nodes.

Definition 3. *We define $d \geq 0$ as the smallest d such that there exists $a > 0$ for which for all $h_m > 0$*

$$\sup_{h \leq h_m} \mathbb{E}\left[K^{-h}\prod_{h'=0}^{h-1}\left(\frac{\mathbb{1}\left(\Delta_{\to S^h}\left(S^h_{h'}\right) \leq \gamma^{h-h'}/(1-\gamma)\right)}{\max\left(\Delta_{\to S^h}\left(S^h_{h'}\right), \gamma^{h_m-h'}\right)^2} + OPT^h_{h'}\right)\right] \leq a\gamma^{-dh_m}$$

With S^h: Random node of depth h chosen according to transition probabilities.
$S^h_{h'}$: `MAX`-node parent of S^h of depth h'.
$OPT^h_{h'}$: 1 if the action at $S^h_{h'}$ to S^h is optimal else 0.
If no such d exists, we set $d = \infty$.

- It also takes into account the difficulty to identify the near-optimal paths.
- d is higher when low gap nodes are concentrated.

References

Michael Kearns, Yishay Mansour, and Andrew Y. Ng. *A sparse sampling algorithm for near- optimal planning in large Markov decision processes.* ICAI, 1999.

Thomas J Walsh, Sergiu Goschin, and Michael L Littman. *Integrating sample-based planning and model-based reinforcement learning.* AAAI, 2010.

Balazs Szorenyi, Gunnar Kedenburg, and Remi Munos. *Optimistic planning in Markov decision processes using a generative model.* NIPS, 2014.