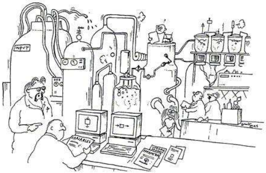
Open-Loop Optimistic Planning

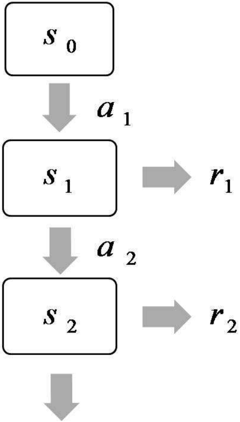
Sébastien Bubeck

joint work with Rémi Munos

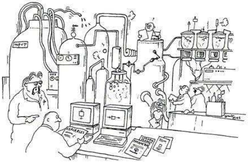
INRIA Lille, SequeL team

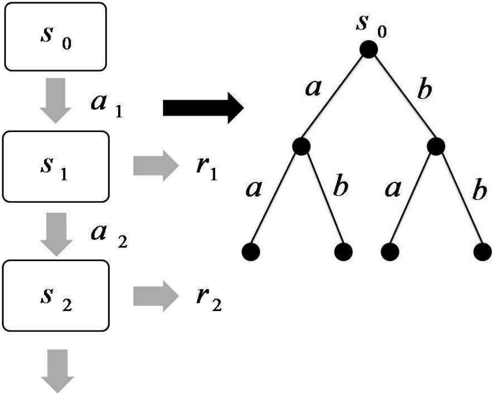
Reinforcement learning in very large spaces





Reinforcement learning in very large spaces with open-loop planning!





Summary of the talk

- Mathematical framework for open-loop planning.
- A simple planner: uniform planning.
- Minimax lower bound.
- An adaptive optimistic planner: OLOP (Open-Loop Optimistic Planning).
- Comparison with other planners.

Summary of the talk

- Mathematical framework for open-loop planning.
- A simple planner: uniform planning.
- Minimax lower bound.
- An adaptive optimistic planner: OLOP (Open-Loop Optimistic Planning).
- Comparison with other planners.

Summary of the talk

- Mathematical framework for open-loop planning.
- A simple planner: uniform planning.
- Minimax lower bound.
- An adaptive optimistic planner: OLOP (Open-Loop Optimistic Planning).
- Comparison with other planners.

Summary of the talk

- Mathematical framework for open-loop planning.
- A simple planner: uniform planning.
- Minimax lower bound.
- An adaptive optimistic planner: OLOP (Open-Loop Optimistic Planning).
- Comparison with other planners.

Summary of the talk

- Mathematical framework for open-loop planning.
- A simple planner: uniform planning.
- Minimax lower bound.
- An adaptive optimistic planner: OLOP (Open-Loop Optimistic Planning).
- Comparison with other planners.

Exploration in a stochastic and discounted environment

Parameters available to the agent: discount factor $\gamma \in (0, 1)$, finite set of actions A, number of rounds n.

Parameters unknown to the agent: the reward distributions (over $[0, 1]$) $\nu(a)$, $a \in A^*$, with mean $\mu(a)$.

For each episode $m \geq 1$; for each moment in the episode $t \geq 1$;

1. If n actions have already been performed then the agent outputs an action $a(n) \in A$ and the game stops.
2. The agent chooses an action $a_t^m \in A$.
3. The environment draws $Y_t^m \sim \nu(a_{1:t}^m)$ and the agent receives the reward $\gamma^t Y_t^m$.
4. The agent decides to either move the next moment $t + 1$ in the episode or to reset to its initial position and move the next episode $m + 1$.

Exploration in a stochastic and discounted environment

Parameters available to the agent: discount factor $\gamma \in (0, 1)$, finite set of actions A, number of rounds n.

Parameters unknown to the agent: the reward distributions (over $[0, 1]$) $\nu(a)$, $a \in A^*$, with mean $\mu(a)$.

For each episode $m \geq 1$; for each moment in the episode $t \geq 1$;

1. If n actions have already been performed then the agent outputs an action $a(n) \in A$ and the game stops.
2. The agent chooses an action $a_t^m \in A$.
3. The environment draws $Y_t^m \sim \nu(a_{1:t}^m)$ and the agent receives the reward $\gamma^t Y_t^m$.
4. The agent decides to either move the next moment $t + 1$ in the episode or to reset to its initial position and move the next episode $m + 1$.

Exploration in a stochastic and discounted environment

Parameters available to the agent: discount factor $\gamma \in (0, 1)$, finite set of actions A, number of rounds n.

Parameters unknown to the agent: the reward distributions (over $[0, 1]$) $\nu(a)$, $a \in A^*$, with mean $\mu(a)$.

For each episode $m \geq 1$; for each moment in the episode $t \geq 1$;

1. If n actions have already been performed then the agent outputs an action $a(n) \in A$ and the game stops.
2. The agent chooses an action $a_t^m \in A$.
3. The environment draws $Y_t^m \sim \nu(a_{1:t}^m)$ and the agent receives the reward $\gamma^t Y_t^m$.
4. The agent decides to either move the next moment $t + 1$ in the episode or to reset to its initial position and move the next episode $m + 1$.

Exploration in a stochastic and discounted environment

Parameters available to the agent: discount factor $\gamma \in (0, 1)$, finite set of actions A, number of rounds n.

Parameters unknown to the agent: the reward distributions (over $[0, 1]$) $\nu(a)$, $a \in A^*$, with mean $\mu(a)$.

For each episode $m \geq 1$; for each moment in the episode $t \geq 1$;

1. If n actions have already been performed then the agent outputs an action $a(n) \in A$ and the game stops.
2. The agent chooses an action $a_t^m \in A$.
3. The environment draws $Y_t^m \sim \nu(a_{1:t}^m)$ and the agent receives the reward $\gamma^t Y_t^m$.
4. The agent decides to either move the next moment $t + 1$ in the episode or to reset to its initial position and move the next episode $m + 1$.

Exploration in a stochastic and discounted environment

Parameters available to the agent: discount factor $\gamma \in (0, 1)$, finite set of actions A, number of rounds n.

Parameters unknown to the agent: the reward distributions (over $[0, 1]$) $\nu(a)$, $a \in A^*$, with mean $\mu(a)$.

For each episode $m \geq 1$; for each moment in the episode $t \geq 1$;

1. If n actions have already been performed then the agent outputs an action $a(n) \in A$ and the game stops.
2. The agent chooses an action $a_t^m \in A$.
3. The environment draws $Y_t^m \sim \nu(a_{1:t}^m)$ and the agent receives the reward $\gamma^t Y_t^m$.
4. The agent decides to either move the next moment $t + 1$ in the episode or to reset to its initial position and move the next episode $m + 1$.

Exploration in a stochastic and discounted environment

Parameters available to the agent: discount factor $\gamma \in (0, 1)$, finite set of actions A, number of rounds n.

Parameters unknown to the agent: the reward distributions (over $[0, 1]$) $\nu(a)$, $a \in A^*$, with mean $\mu(a)$.

For each episode $m \geq 1$; for each moment in the episode $t \geq 1$;

1. If n actions have already been performed then the agent outputs an action $a(n) \in A$ and the game stops.
2. The agent chooses an action $a_t^m \in A$.
3. The environment draws $Y_t^m \sim \nu(a_{1:t}^m)$ and the agent receives the reward $\gamma^t Y_t^m$.
4. The agent decides to either move the next moment $t + 1$ in the episode or to reset to its initial position and move the next episode $m + 1$.

Exploration in a stochastic and discounted environment

Parameters available to the agent: discount factor $\gamma \in (0, 1)$, finite set of actions A, number of rounds n.

Parameters unknown to the agent: the reward distributions (over $[0, 1]$) $\nu(a)$, $a \in A^*$, with mean $\mu(a)$.

For each episode $m \geq 1$; for each moment in the episode $t \geq 1$;

1. If n actions have already been performed then the agent outputs an action $a(n) \in A$ and the game stops.
2. The agent chooses an action $a_t^m \in A$.
3. The environment draws $Y_t^m \sim \nu(a_{1:t}^m)$ and the agent receives the reward $\gamma^t Y_t^m$.
4. The agent decides to either move the next moment $t + 1$ in the episode or to reset to its initial position and move the next episode $m + 1$.

Simple regret

- **Goal:** find the optimal immediate action.
- Define the value of a sequence of actions $a \in A^h$ as:

$$V(a) = \sup_{u \in A^\infty : u_{1:h} = a} \sum_{t \geq 1} \gamma^t \mu(u_{1:t}).$$

- Define the simple regret of a planner as

$$r_n = \max_{a \in A} V(a) - V(a(n)).$$

Simple regret

- **Goal:** find the optimal immediate action.
- Define the value of a sequence of actions $a \in A^h$ as:

$$V(a) = \sup_{u \in A^\infty : u_{1:h} = a} \sum_{t \geq 1} \gamma^t \mu(u_{1:t}).$$

- Define the simple regret of a planner as

$$r_n = \max_{a \in A} V(a) - V(a(n)).$$

Simple regret

- **Goal:** find the optimal immediate action.
- Define the value of a sequence of actions $a \in A^h$ as:

$$V(a) = \sup_{u \in A^\infty : u_{1:h} = a} \sum_{t \geq 1} \gamma^t \mu(u_{1:t}).$$

- Define the simple regret of a planner as

$$r_n = \max_{a \in A} V(a) - V(a(n)).$$

Uniform planning

- Let $H \in \mathbb{N}$ be the largest integer such that $HK^H \leq n$.
- For each sequence of actions $a \in A^H$, allocate one episode (of length H) to estimate the value of the sequence a. That is, receive $Y_t^a \sim \nu(a_{1:t})$, $1 \leq t \leq H$ (drawn independently).
- Compute, for all $a \in A^h$, $h \leq H$,

$$\widehat{\mu}(a) = \frac{1}{K^{H-h}} \sum_{b \in A^H : b_{1:h} = a} Y_h^b.$$

- Compute, for all $a \in A^H$, $\widehat{V}(a) = \sum_{t=1}^{H} \gamma^t \widehat{\mu}(a_{1:t})$.
- Let $a(n) \in A$ be the first action of the sequence $\arg\max_{a \in A^H} \widehat{V}(a)$.

Uniform planning

- Let $H \in \mathbb{N}$ be the largest integer such that $HK^H \leq n$.
- For each sequence of actions $a \in A^H$, allocate one episode (of length H) to estimate the value of the sequence a. That is, receive $Y_t^a \sim \nu(a_{1:t})$, $1 \leq t \leq H$ (drawn independently).
- Compute, for all $a \in A^h$, $h \leq H$,

$$\widehat{\mu}(a) = \frac{1}{K^{H-h}} \sum_{b \in A^H : b_{1:h} = a} Y_h^b .$$

- Compute, for all $a \in A^H$, $\widehat{V}(a) = \sum_{t=1}^{H} \gamma^t \widehat{\mu}(a_{1:t})$.
- Let $a(n) \in A$ be the first action of the sequence $\arg\max_{a \in A^H} \widehat{V}(a)$.

Uniform planning

- Let $H \in \mathbb{N}$ be the largest integer such that $HK^H \leq n$.
- For each sequence of actions $a \in A^H$, allocate one episode (of length H) to estimate the value of the sequence a. That is, receive $Y_t^a \sim \nu(a_{1:t})$, $1 \leq t \leq H$ (drawn independently).
- Compute, for all $a \in A^h$, $h \leq H$,

$$\widehat{\mu}(a) = \frac{1}{K^{H-h}} \sum_{b \in A^H : b_{1:h} = a} Y_h^b.$$

- Compute, for all $a \in A^H$, $\widehat{V}(a) = \sum_{t=1}^{H} \gamma^t \widehat{\mu}(a_{1:t})$.
- Let $a(n) \in A$ be the first action of the sequence $\arg\max_{a \in A^H} \widehat{V}(a)$.

Uniform planning

- Let $H \in \mathbb{N}$ be the largest integer such that $HK^H \leq n$.
- For each sequence of actions $a \in A^H$, allocate one episode (of length H) to estimate the value of the sequence a. That is, receive $Y_t^a \sim \nu(a_{1:t})$, $1 \leq t \leq H$ (drawn independently).
- Compute, for all $a \in A^h$, $h \leq H$,

$$\widehat{\mu}(a) = \frac{1}{K^{H-h}} \sum_{b \in A^H : b_{1:h} = a} Y_h^b .$$

- Compute, for all $a \in A^H$, $\widehat{V}(a) = \sum_{t=1}^{H} \gamma^t \widehat{\mu}(a_{1:t})$.
- Let $a(n) \in A$ be the first action of the sequence $\arg\max_{a \in A^H} \widehat{V}(a)$.

Uniform planning

- Let $H \in \mathbb{N}$ be the largest integer such that $HK^H \leq n$.
- For each sequence of actions $a \in A^H$, allocate one episode (of length H) to estimate the value of the sequence a. That is, receive $Y_t^a \sim \nu(a_{1:t})$, $1 \leq t \leq H$ (drawn independently).
- Compute, for all $a \in A^h$, $h \leq H$,

$$\widehat{\mu}(a) = \frac{1}{K^{H-h}} \sum_{b \in A^H : b_{1:h} = a} Y_h^b.$$

- Compute, for all $a \in A^H$, $\widehat{V}(a) = \sum_{t=1}^{H} \gamma^t \widehat{\mu}(a_{1:t})$.
- Let $a(n) \in A$ be the first action of the sequence $\arg\max_{a \in A^H} \widehat{V}(a)$.

Regret bound for uniform planning

Theorem

Uniform planning satisfies:

$$\mathbb{E}r_n = \begin{cases} \tilde{O}\left(n^{-\frac{\log 1/\gamma}{\log K}}\right) & \textit{if } \gamma\sqrt{K} > 1, \\ \tilde{O}\left(n^{-\frac{1}{2}}\right) & \textit{if } \gamma\sqrt{K} \leq 1. \end{cases}$$

Minimax lower bound

Theorem

Any agent satisfies:

$$\sup_{\nu} \mathbb{E} r_n = \begin{cases} \Omega\left(n^{-\frac{\log 1/\gamma}{\log K}}\right) & \text{if } \gamma\sqrt{K} > 1, \\ \Omega\left(n^{-\frac{1}{2}}\right) & \text{if } \gamma\sqrt{K} \leq 1. \end{cases}$$

OLOP (Open-Loop Optimistic Planning)

Let $L = \lceil \log n / (2 \log 1/\gamma) \rceil$ and M be the largest integer such that $ML \leq n$.

For each episode $m = 1, 2, \ldots, M$;

1. The agent computes the B–values at time $m-1$ for sequences of actions in A^L and chooses

$$a^m \in \operatorname*{argmax}_{a \in A^L} B_a(m-1).$$

2. The environment draws the sequence of rewards $Y_t^m \sim \nu(a_{1:t}^m)$, $t = 1, \ldots, L$.

Return an action that has been the most played: $a(n) = \mathrm{argmax}_{a \in A} T_a(M)$.

OLOP (Open-Loop Optimistic Planning)

Let $L = \lceil \log n/(2 \log 1/\gamma) \rceil$ and M be the largest integer such that $ML \leq n$.

For each episode $m = 1, 2, \dots, M$;

1. The agent computes the B–values at time $m-1$ for sequences of actions in A^L and chooses

$$a^m \in \operatorname*{argmax}_{a \in A^L} B_a(m-1).$$

2. The environment draws the sequence of rewards $Y_t^m \sim \nu(a_{1:t}^m)$, $t = 1, \dots, L$.

Return an action that has been the most played: $a(n) = \operatorname{argmax}_{a \in A} T_a(M)$.

OLOP (Open-Loop Optimistic Planning)

Let $L = \lceil \log n / (2 \log 1/\gamma) \rceil$ and M be the largest integer such that $ML \leq n$.

For each episode $m = 1, 2, \ldots, M$;

1. The agent computes the B–values at time $m - 1$ for sequences of actions in A^L and chooses

$$a^m \in \operatorname*{argmax}_{a \in A^L} B_a(m-1).$$

2. The environment draws the sequence of rewards $Y_t^m \sim \nu(a_{1:t}^m)$, $t = 1, \ldots, L$.

Return an action that has been the most played: $a(n) = \mathrm{argmax}_{a \in A} T_a(M)$.

OLOP (Open-Loop Optimistic Planning)

Let $L = \lceil \log n / (2 \log 1/\gamma) \rceil$ and M be the largest integer such that $ML \leq n$.

For each episode $m = 1, 2, \ldots, M$;

1. The agent computes the B–values at time $m - 1$ for sequences of actions in A^L and chooses

$$a^m \in \underset{a \in A^L}{\operatorname{argmax}} B_a(m-1).$$

2. The environment draws the sequence of rewards $Y_t^m \sim \nu(a_{1:t}^m)$, $t = 1, \ldots, L$.

Return an action that has been the most played: $a(n) = \operatorname{argmax}_{a \in A} T_a(M)$.

OLOP (Open-Loop Optimistic Planning)

Let $L = \lceil \log n/(2 \log 1/\gamma) \rceil$ and M be the largest integer such that $ML \leq n$.

For each episode $m = 1, 2, \ldots, M$;

1. The agent computes the B–values at time $m-1$ for sequences of actions in A^L and chooses

$$a^m \in \operatorname*{argmax}_{a \in A^L} B_a(m-1).$$

2. The environment draws the sequence of rewards $Y_t^m \sim \nu(a_{1:t}^m)$, $t = 1, \ldots, L$.

Return an action that has been the most played: $a(n) = \operatorname{argmax}_{a \in A} T_a(M)$.

Definition of B–values

For any $1 \leq h \leq L$, for any $a \in A^h$, let

- $T_a(m) = \sum_{s=1}^{m} \mathbb{1}\{a_{1:h}^s = a\}$,
- $\widehat{\mu}_a(m) = \frac{1}{T_a(m)} \sum_{s=1}^{m} Y_h^s \mathbb{1}\{a_{1:h}^s = a\}$,
- $U_a(m) = \sum_{t=1}^{h} \left(\gamma^t \widehat{\mu}_{a_{1:t}}(m) + \gamma^t \sqrt{\frac{2 \log M}{T_{a_{1:t}}(m)}} \right) + \frac{\gamma^{h+1}}{1-\gamma}$.

Now for sequences $a \in A^L$, let

$$B_a(m) = \inf_{1 \leq h \leq L} U_{a_{1:h}}(m).$$

Definition of B–values

For any $1 \leq h \leq L$, for any $a \in A^h$, let

- $T_a(m) = \sum_{s=1}^{m} \mathbb{1}\{a_{1:h}^s = a\}$,
- $\widehat{\mu}_a(m) = \frac{1}{T_a(m)} \sum_{s=1}^{m} Y_h^s \mathbb{1}\{a_{1:h}^s = a\}$,
- $U_a(m) = \sum_{t=1}^{h} \left(\gamma^t \widehat{\mu}_{a_{1:t}}(m) + \gamma^t \sqrt{\frac{2 \log M}{T_{a_{1:t}}(m)}} \right) + \frac{\gamma^{h+1}}{1-\gamma}$.

Now for sequences $a \in A^L$, let

$$B_a(m) = \inf_{1 \leq h \leq L} U_{a_{1:h}}(m).$$

Definition of B–values

For any $1 \leq h \leq L$, for any $a \in A^h$, let

- $T_a(m) = \sum_{s=1}^m \mathbb{1}\{a_{1:h}^s = a\}$,
- $\widehat{\mu}_a(m) = \frac{1}{T_a(m)} \sum_{s=1}^m Y_h^s \mathbb{1}\{a_{1:h}^s = a\}$,
- $U_a(m) = \sum_{t=1}^h \left(\gamma^t \widehat{\mu}_{a_{1:t}}(m) + \gamma^t \sqrt{\frac{2 \log M}{T_{a_{1:t}}(m)}} \right) + \frac{\gamma^{h+1}}{1-\gamma}$.

Now for sequences $a \in A^L$, let

$$B_a(m) = \inf_{1 \leq h \leq L} U_{a_{1:h}}(m).$$

Definition of B–values

For any $1 \leq h \leq L$, for any $a \in A^h$, let

- $T_a(m) = \sum_{s=1}^{m} \mathbb{1}\{a^s_{1:h} = a\}$,
- $\widehat{\mu}_a(m) = \frac{1}{T_a(m)} \sum_{s=1}^{m} Y^s_h \mathbb{1}\{a^s_{1:h} = a\}$,
- $U_a(m) = \sum_{t=1}^{h} \left(\gamma^t \widehat{\mu}_{a_{1:t}}(m) + \gamma^t \sqrt{\frac{2 \log M}{T_{a_{1:t}}(m)}} \right) + \frac{\gamma^{h+1}}{1-\gamma}$.

Now for sequences $a \in A^L$, let

$$B_a(m) = \inf_{1 \leq h \leq L} U_{a_{1:h}}(m).$$

Definition of B–values

For any $1 \leq h \leq L$, for any $a \in A^h$, let

- $T_a(m) = \sum_{s=1}^{m} \mathbb{1}\{a_{1:h}^s = a\}$,
- $\widehat{\mu}_a(m) = \frac{1}{T_a(m)} \sum_{s=1}^{m} Y_h^s \mathbb{1}\{a_{1:h}^s = a\}$,
- $U_a(m) = \sum_{t=1}^{h} \left(\gamma^t \widehat{\mu}_{a_{1:t}}(m) + \gamma^t \sqrt{\frac{2 \log M}{T_{a_{1:t}}(m)}} \right) + \frac{\gamma^{h+1}}{1-\gamma}$.

Now for sequences $a \in A^L$, let

$$B_a(m) = \inf_{1 \leq h \leq L} U_{a_{1:h}}(m).$$

Regret bound for OLOP

Define $\kappa_c \in [1, K]$ as the branching factor of the set of sequences in A^h that are $c\frac{\gamma^{h+1}}{1-\gamma}$-optimal, where $c > 0$, i.e.

$$\kappa_c = \limsup_{h\to\infty} \left| \left\{ a \in A^h : V(a) \geq V - c\frac{\gamma^{h+1}}{1-\gamma} \right\} \right|^{1/h}.$$

Theorem

For any $\kappa' > \kappa_2$, OLOP satisfies:

$$\mathbb{E}r_n = \begin{cases} \tilde{O}\left(n^{-\frac{\log 1/\gamma}{\log \kappa'}}\right) & \text{if } \gamma\sqrt{\kappa'} > 1, \\ \tilde{O}\left(n^{-\frac{1}{2}}\right) & \text{if } \gamma\sqrt{\kappa'} \leq 1. \end{cases}$$

Regret bound for OLOP

Define $\kappa_c \in [1, K]$ as the branching factor of the set of sequences in A^h that are $c\frac{\gamma^{h+1}}{1-\gamma}$-optimal, where $c > 0$, i.e.

$$\kappa_c = \limsup_{h\to\infty} \left| \left\{ a \in A^h : V(a) \geq V - c\frac{\gamma^{h+1}}{1-\gamma} \right\} \right|^{1/h}.$$

Theorem

For any $\kappa' > \kappa_2$, OLOP satisfies:

$$\mathbb{E}r_n = \begin{cases} \tilde{O}\left(n^{-\frac{\log 1/\gamma}{\log \kappa'}}\right) & \text{if } \gamma\sqrt{\kappa'} > 1, \\ \tilde{O}\left(n^{-\frac{1}{2}}\right) & \text{if } \gamma\sqrt{\kappa'} \leq 1. \end{cases}$$

Comparison with Zooming Algorithm, HOO, UCB-Air (case $\gamma\sqrt{K} > 1$)