Recent Advances in Batch Mode Reinforcement Learning

Synthesizing Artificial Trajectories

R. Fonteneau\(^{(1)}\), S.A. Murphy\(^{(2)}\), L. Wehenkel\(^{(1)}\), D. Ernst\(^{(1)}\)

\(^{(1)}\) University of Liège, Belgium – \(^{(2)}\) University of Michigan, USA

GRASCOMP's Day, November 3th, 2011
Outline

- **Batch Mode Reinforcement Learning**
  - Reinforcement Learning & Batch Mode Reinforcement Learning
  - Formalization, Objectives, Main Difficulties & Usual Approach

- **A New Approach: Synthesizing Artificial Trajectories**
  - Artificial Trajectories
  - Estimating the Performances of Policies
  - Computing Bounds & Inferring Safe Policies
  - Sampling Strategies
  - Connexion to Classic Batch Mode Reinforcement Learning

- **Conclusions**
Batch Mode Reinforcement Learning
Reinforcement Learning

Examples of rewards:

- Reinforcement Learning (RL) aims at finding a policy maximizing received rewards by interacting with the environment.
Batch Mode Reinforcement Learning

- All the available information is contained in a **batch collection of data**
- Batch mode RL aims at computing a (near-)optimal policy from this collection of data

Finite collection of trajectories of the agent
Formalization

- System dynamics: \( x_{t+1} = f(x_t, u_t, w_t) \quad \forall t \in \{0, \ldots, T - 1\} \)

- Reward function: \( r_t = \rho(x_t, u_t, w_t) \quad \forall t \in \{0, \ldots, T - 1\} \)

- Performance of a policy \( h : \{0, \ldots, T - 1\} \times \mathcal{X} \to \mathcal{U} \)

  \begin{itemize}
  \item Expected T-stage return: \( J^h(x_0) = \mathbb{E}_{w_0, \ldots, w_{T-1} \sim p_W(\cdot)}[R^h(x_0)] \)
  \item Value-at-risk: \( J_{VAR}^{h,(b,c)}(x_0) = \begin{cases} 
  -\infty & \text{if } P(R^h(x_0) < b) > c \\
  J^h(x_0) & \text{otherwise}.
  \end{cases} \)
  \end{itemize}

\[
R^h(x_0) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t), w_t)
\]

\[
w_t \sim p_W(\cdot) \quad x_{t+1} = f(x_t, h(t, x_t), w_t) \quad \forall t \in \{0, \ldots, T - 1\}
\]
Formalization

- The system dynamics, reward function and disturbance probability distribution are unknown.
- Instead, we have access to a sample of one-step system transitions:

\[ \mathcal{F}_n = \left\{ (x^l, u^l, r^l, y^l) \right\}_{l=1}^n \]

\[ y^l = f(x^l, u^l, w^l) \]

\[ r^l = \rho(x^l, u^l, w^l) \]

\[ w^l \sim p_W(\cdot) \]
Objectives

● Main goal: **Finding a "good" policy**

● Many associated subproblems:
  − Evaluating the performance of a given policy
  − Computing performance guarantees and safe policies
  − Generating additional sample transitions
  − ...
Main Difficulties & Usual Approach

Main Difficulties

- Functions are unknown (and not accessible to simulation)
- The state-space and/or the action space are large or continuous
- Highly stochastic environments

Usual Approach

- To combine dynamic programming with function approximators (neural networks, regression trees, SVM, linear regression over basis functions, etc)
- Function approximators have two main roles:
  - To offer a concise representation of state-action value function for deriving value / policy iteration algorithms
  - To generalize information contained in the finite sample

Remaining Challenges

- The black box nature of function approximators may have some unwanted effects: hazardous generalization, difficulties to compute performance guarantees, unefficient use of optimal trajectories, no straightforward sampling strategies,...
A New Approach: Synthesizing Artificial Trajectories
Artificial Trajectories

- Artificial trajectories are *(ordered) sequences of elementary pieces of trajectories:*

\[
\left[ (x^{l_0}, u^{l_0}, r^{l_0}, y^{l_0}), \ldots, (x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, y^{l_{T-1}}) \right] \in \mathcal{F}_n^T
\]

\[
l_t \in \{1, \ldots, n\}, \quad \forall t \in \{1, \ldots, T - 1\}
\]
Estimating the Performances of Policies

Expected Return

- If the system dynamics and the reward function were accessible to simulation, then **Monte Carlo estimation** would allow estimating the performance of $h$

- We propose an approach that mimics Monte Carlo (MC) estimation by rebuilding $p$ **artificial trajectories** from one-step system transitions

- These artificial trajectories are built so as to **minimize the discrepancy (using a distance metric $\Delta$) with a classical MC sample** that could be obtained by simulating the system with the policy $h$; each one step transition is used at most once

- We average the cumulated returns over the $p$ artificial trajectories to obtain the **Model-free Monte Carlo estimator** (MFMC) of the expected return of $h$:

$$M^h_p (\mathcal{F}_n, x_0) = \frac{1}{p} \sum_{i=1}^{p} \sum_{t=0}^{T-1} r^{i,t}$$
Estimating the Performances of Policies

Monte Carlo Estimator

- Illustration with $p=3$, $T=4$

\[ M_3^h(x_0) = \left( \frac{l_0^1 + l_1^1 + l_2^1 + l_3^1}{3} \right) + \left( \frac{l_0^2 + l_1^2 + l_2^2 + l_3^2}{3} \right) + \left( \frac{l_0^3 + l_1^3 + l_2^3 + l_3^3}{3} \right) \]

MODEL OR SIMULATOR REQUIRED!
Estimating the Performances of Policies

Model-free Monte Carlo Estimator

- Illustration with $p=3$, $T=4$

\[
\mathcal{M}_3^h (\mathcal{F}_n, x_0) = \frac{\left( r_{l_0}^1 + r_{l_1}^1 + r_{l_2}^1 + r_{l_3}^1 \right) + \left( r_{l_0}^2 + r_{l_1}^2 + r_{l_2}^2 + r_{l_3}^2 \right) + \left( r_{l_0}^3 + r_{l_1}^3 + r_{l_2}^3 + r_{l_3}^3 \right)}{3}
\]
Estimating the Performances of Policies

Additional Assumptions

Assumption: Lipschitz continuity of the functions $f$, $\rho$ and $h$.
\[
\forall (x, x', u, u', w) \in \mathcal{X}^2 \times \mathcal{U}^2 \times \mathcal{W},
\]
\[
\| f(x, u, w) - f(x', u', w) \|_\mathcal{X} \leq L_f (\| x - x' \|_\mathcal{X} + \| u - u' \|_\mathcal{U}),
\]
\[
| \rho(x, u, w) - \rho(x', u', w) | \leq L_\rho (\| x - x' \|_\mathcal{X} + \| u - u' \|_\mathcal{U}),
\]
\[
\| h(t, x) - h(t, x') \|_\mathcal{U} \leq L_h \| x - x' \|_\mathcal{X}, \forall t \in \{0, \ldots, T - 1\}
\]

Definition (Distance Metric $\Delta$)
\[
\forall (x, x', u, u') \in \mathcal{X}^2 \times \mathcal{U}^2,
\Delta((x, u), (x', u')) = \| x - x' \|_\mathcal{X} + \| u - u' \|_\mathcal{U}.
\]

Definition ($k-$Dispersion)
\[
\alpha_k(\mathcal{P}_n) = \sup_{(x, u) \in \mathcal{X} \times \mathcal{U}} \Delta^\mathcal{P}_k (x, u),
\]

where $\Delta^\mathcal{P}_k (x, u)$ denotes the distance of $(x, u)$ to its $k-$th nearest neighbor (using the distance metric $\Delta$) in the $\mathcal{P}_n$ sample.
Estimating the Performances of Policies

Theoretical Results

Theorem  (Bias Bound for $\mathcal{M}_p^h \left( \tilde{F}_n, x_0 \right)$)

\[ |J^h(x_0) - E_{p, \mathcal{P}_n}^h(x_0)| \leq C \alpha_{pT} (\mathcal{P}_n) \]

\[ \text{with } C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} (L_f (1 + L_h))^i \]

Theorem  (Variance Bound for $\mathcal{M}_p^h \left( \tilde{F}_n, x_0 \right)$)

\[ V_{p, \mathcal{P}_n}^h(x_0) \leq \left( \frac{\sigma_{R^h}(x_0)}{\sqrt{p}} + 2C \alpha_{pT} (\mathcal{P}_n) \right)^2 \]

\[ \text{with } C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} (L_f (1 + L_h))^i \]
Estimating the Performances of Policies

Experimental Results

\[
x_{t+1} = \sin \left( \frac{\pi}{2}(x_t + u_t + w_t) \right) \\
\rho(x_t, u_t, w_t) = \frac{1}{2\pi} e^{-\frac{1}{2}(x_t^2 + u_t^2)} + w_t \\
h(t, x) = -\frac{x}{2} \quad x_0 = -0.5 \\
\mathcal{W} = \left[ -\frac{\epsilon}{2}, \frac{\epsilon}{2} \right] 
\]
Consider again the $p$ artificial trajectories that were rebuilt by the MFMC estimator.

The Value-at-Risk of the policy $h$ can be straightforwardly estimated as follows:

$$
\tilde{j}^{h,(b,c)}_{\text{VaR}}(x_0) = \begin{cases} 
-\infty & \text{if } \frac{1}{p} \sum_{i=1}^{p} \mathbb{I}\{r^i < b\} > c, \\
\mathcal{M}^h(\mathcal{F}_n, x_0) & \text{otherwise}
\end{cases}
$$

$$
r^i = \sum_{t=0}^{T-1} r^i_t
$$

$b \in \mathbb{R}$

$c \in [0, 1]$
Deterministic Case: Computing Bounds

Lower Bound from a Single Trajectory

Proposition (Lower Bound from any Artificial Trajectory)

Let \([ (x_{lt}, u_{lt}, r_{lt}, y_{lt}) ]_{t=0}^{T-1} \) be any artificial trajectory. Then,

\[
J^h(x_0) \geq \sum_{t=0}^{T-1} r_{lt} - \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta \left( (y_{lt-1}, h(t, y_{lt-1})), (x_{lt}, u_{lt}) \right)
\]

where

\[
L_{Q_{T-t}} = L_{\rho} \sum_{i=0}^{T-t-1} \left( L_f (1 + L_h) \right)^i
\]

and \( y_{l-1} = x_0 \).
Deterministic Case: Computing Bounds

Maximal Bounds

Definition (Maximal Lower Bound)

\[
L^h(\mathcal{F}_n, x_0) = \max_{[(x^t, u^t, r_t, y^t)]_{t=0}^{T-1} \in \mathcal{F}_n^T} \sum_{t=0}^{T-1} r_t
\]

\[
- \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta \left( (y^{t-1}, h(t, y^{t-1})), (x^t, u^t) \right)
\]

Definition (Minimal Upper Bound)

\[
U^h(\mathcal{F}_n, x_0) = \min_{[(x^t, u^t, r_t, y^t)]_{t=0}^{T-1} \in \mathcal{F}_n^T} \sum_{t=0}^{T-1} r_t
\]

\[
+ \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta \left( (y^{t-1}, h(t, y^{t-1})), (x^t, u^t) \right)
\]
Deterministic Case: Computing Bounds

Tightness of Maximal Bounds

**Proposition (Tightness of the Bounds)**

\[ \exists C_b > 0 : \quad J^h(x_0) - L^h(\mathcal{F}_n, x_0) \leq C_b \alpha_1(\mathcal{P}_n) \]
\[ U^h(\mathcal{F}_n, x_0) - J^h(x_0) \leq C_b \alpha_1(\mathcal{P}_n) \]

where \( \alpha_1(\mathcal{P}_n) \) denotes the 1–dispersion of the sample of system transitions \( \mathcal{F}_n \).
Inferring Safe Policies
From Lower Bounds to Cautious Policies

- Consider the set of open-loop policies:

\[ \Pi = \{ \pi : \{0, \ldots, T-1\} \rightarrow \mathcal{U} \} \]

- For such policies, bounds can be computed in a similar way

- We can then search for a specific policy for which the associated lower bound is maximized:

\[ \hat{\pi}^*_{\mathcal{F}_n, x_0} \in \arg \max_{\pi \in \Pi} L^\pi (\mathcal{F}_n, x_0) \]

- A \( O(T n^2) \) algorithm for doing this: the CGRL algorithm (Cautious approach to Generalization in RL)
Inferring Safe Policies

Convergence

**Theorem  (Convergence of $\hat{\pi}^*_{F_n, x_0}$)**

Let $\mathcal{J}^*(x_0)$ be the set of optimal open-loop policies:

$$\mathcal{J}^*(x_0) = \arg\max_{\pi \in \Pi} J^\pi(x_0),$$

and let us suppose that $\mathcal{J}^*(x_0) \neq \Pi$ (if $\mathcal{J}^*(x_0) = \Pi$, the search for an optimal policy is indeed trivial). We define

$$\epsilon(x_0) = \min_{\pi \in \Pi \setminus \mathcal{J}^*(x_0)} \left\{ \left( \max_{\pi' \in \Pi} J^{\pi'}(x_0) \right) - J^\pi(x_0) \right\}.$$  

Then,

$$\left( C'_b \alpha^*(\mathcal{P}_n) < \epsilon(x_0) \right) \implies \hat{\pi}^*_{F_n, x_0} \in \mathcal{J}^*(x_0).$$
Inferring Safe Policies

Experimental Results

- The puddle world benchmark
## Inferring Safe Policies

### Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>CGRL</th>
<th>FQI (Fitted Q Iteration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The state space is</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>uniformly covered by</td>
<td>Initial state</td>
<td>Initial state</td>
</tr>
<tr>
<td>the sample</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>Information about the</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>Puddle area is removed</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Inferring Safe Policies

Bonus

Theorem (Optimal Policies computed from Optimal Trajectories)

Let $\pi^*_x \in \mathcal{J}^*(x_0)$ be an optimal open-loop policy. Let us assume that one can find in $\mathcal{F}_n$ a sequence of $T$ one-step system transitions

$$[(x^{l_0}, u^{l_0}, r^{l_0}, x^{l_1}), (x^{l_1}, u^{l_1}, r^{l_1}, x^{l_2}), \ldots, (x^{l_{T-1}}, u^{l_{T-1}}, r^{l_{T-1}}, x^{l_T})] \in \mathcal{F}_{n}^T$$

such that

$$x^{l_0} = x_0,$$

$$u^{l_t} = \pi^*_x(t) \quad \forall t \in \{0, \ldots, T - 1\}.$$ 

Let $\hat{\pi}^{*}_{\mathcal{F}_n, x_0}$ be such that

$$\hat{\pi}^{*}_{\mathcal{F}_n, x_0} \in \arg \max_{\pi \in \Pi} L^\pi(\mathcal{F}_n; x_0).$$

Then,

$$\hat{\pi}^{*}_{\mathcal{F}_n, x_0} \in \mathcal{J}^*(x_0).$$
**Sampling Strategies**

**An Artificial Trajectories Viewpoint**

- Given a sample of system transitions

\[ \mathcal{F}_n = \left\{ (x^l, u^l, r^l, y^l) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R} \times \mathcal{X} \right\}_{l=1}^{n} \]

How can we determine where to sample additional transitions?

- We define the set of candidate optimal policies:

\[ \Pi(\mathcal{F}, x_0) = \left\{ \pi \in \Pi \mid \forall \pi' \in \Pi, U^{\pi}(\mathcal{F}, x_0) \geq L^{\pi'}(\mathcal{F}, x_0) \right\} \]

- A transition \((x, u, r, y) \in \mathcal{X} \times \mathcal{U} \times \mathbb{R} \times \mathcal{X}\) is said compatible with \(\mathcal{F}\) if

\[ \forall(x^l, u^l, r^l, y^l) \in \mathcal{F}, \ (u^l = u) \implies \left\{ \begin{array}{c} |r - r^l| \leq L_p \| x - x^l \|_x, \\ \| y - y^l \|_x \leq L_f \| x - x^l \|_x \end{array} \right\} \]

and we denote by \(\mathcal{C}(\mathcal{F})\) the set of all such compatible transitions.
Sampling Strategies

An Artificial Trajectories Viewpoint

- Iterative scheme:

\[(x^{m+1}, u^{m+1}) \in \arg\min_{(x,u) \in X \times U} \left\{ \max_{(r,y) \in \mathbb{R} \times X} \delta^\pi(X_m \cup \{(x, u, r, y)\}, x_0) \right\} \]

with

\[\delta^\pi(X, x_0) = U^\pi(X, x_0) - L^\pi(X, x_0)\]

- Conjecture:

\[\exists m_0 \in \mathbb{N} \setminus \{0\} : \forall m \in \mathbb{N}, \left( m \geq m_0 \right) \implies \Pi(X_m, x_0) = \hat{F}^*(x_0)\]
Connexion to Classic Batch Mode RL

Towards a New Paradigm for Batch Mode RL

- FQI (evaluation mode) with k-NN:

\[(x_0, h(0, x_0))\]
Connexion to Classic Batch Mode RL
Towards a New Paradigm for Batch Mode RL

Proposition  \((k-\text{NN FQI-PE using Artificial Trajectories})\)

\[
\hat{J}_{\text{FQI}}^h(\mathcal{F}_n, x_0) = \frac{1}{kT} \sum_{i_0=1}^{k} \ldots \sum_{i_{T-1}=1}^{k} \left( r^{i_0} + r^{i_0,i_1} + \ldots + r^{i_0,i_1,\ldots,i_{T-1}} \right).
\]

where the set of rebuilt artificial trajectories

\[
\left\{ \left[ (x^{i_0}, u^{i_0}, r^{i_0}, y^{i_0}), \ldots, (x^{i_0,\ldots,i_{T-1}}, u^{i_0,\ldots,i_{T-1}}, r^{i_0,\ldots,i_{T-1}}, y^{i_0,\ldots,i_{T-1}}) \right] \right\}
\]

is such that \(\forall t \in \{0, \ldots, T-1\}, \forall (i_0, \ldots, i_t) \in \{1, \ldots, k\}^{t+1}, \)

\[
\Delta \left( (x^{i_0,\ldots,i_{t-1}}, h(t, y^{i_0,\ldots,i_{t-1}})), (x^{i_0,\ldots,i_t}, u^{i_0,\ldots,i_t}) \right) \leq \alpha_k(\mathcal{P}_n).
\]
Conclusions

- Rebuilding artificial trajectories: a new approach for batch mode RL
- Several types of problems can be addressed
- Towards a new paradigm for developing new algorithms?

"Batch mode reinforcement learning based on the synthesis of artificial trajectories". R. Fonteneau, S.A. Murphy, L. Wehenkel and D. Ernst. Submitted.


