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

Agent

Actions

Observations, Rewards

Environment

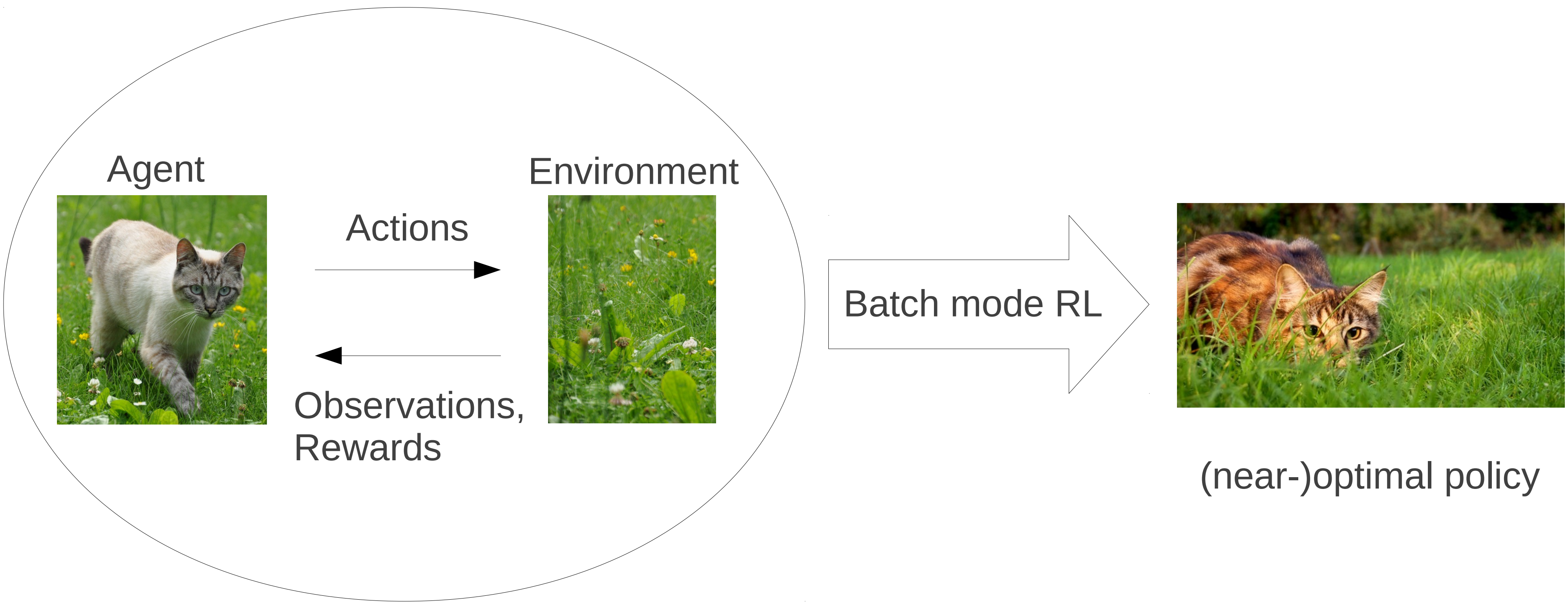
Examples of rewards:




- 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: $x_{t+1} = f(x_t, u_t, w_t) \qquad \forall t \in \{0, \ldots, T-1\}$
- Reward function: $r_t = \rho(x_t, u_t, w_t) \qquad \forall t \in \{0, \ldots, T-1\}$
- Performance of a policy $h : \{0, \ldots, T-1\} \times \mathcal{X} \rightarrow \mathcal{U}$
  - Expected T-stage return: $J^h(x_0) = \underset{w_0, \ldots, w_{T-1} \sim p_{\mathcal{W}}(.)}{\mathbb{E}} \left[R^h(x_0)\right]$
  - Value-at-risk: $J_{VaR}^{h,(b,c)}(x_0) = \begin{cases} -\infty & \text{if } P\left(R^h(x_0) < b\right) > c \\ J^h(x_0) & \text{otherwise .} \end{cases}$

$$b \in \mathbb{R}$$
$$c \in [0, 1[$$

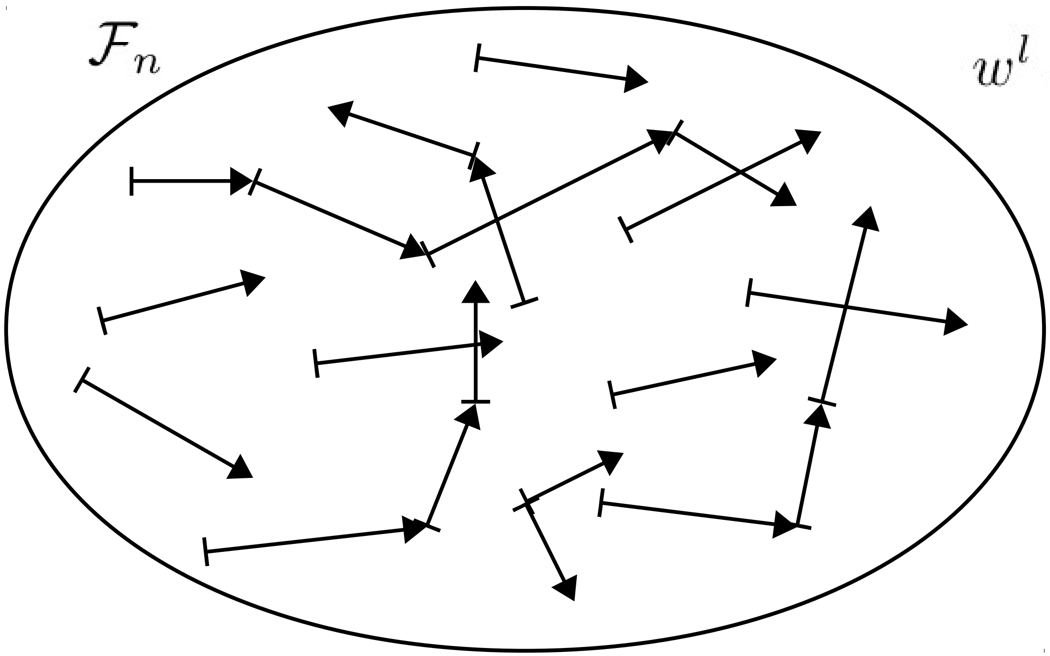
$$R^h(x_0) = \sum_{t=0}^{T-1} \rho(x_t, h(t, x_t), w_t)$$

$$w_t \sim p_{\mathcal{W}}(\cdot) \qquad x_{t+1} = f(x_t, h(t, x_t), w_t) \qquad \forall t \in \{0, \ldots, T-1\}$$

Formalization

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

$$\mathcal{F}_n = \left\{ \left(x^l, u^l, r^l, y^l \right) \right\}_{l=1}^{n} \qquad \forall l \in \{1, \ldots, n\}, \qquad \begin{aligned} y^l &= f\left(x^l, u^l, w^l\right) \\ r^l &= \rho\left(x^l, u^l, w^l\right) \\ w^l &\sim p_{\mathcal{W}}(\cdot) \end{aligned}$$



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,...

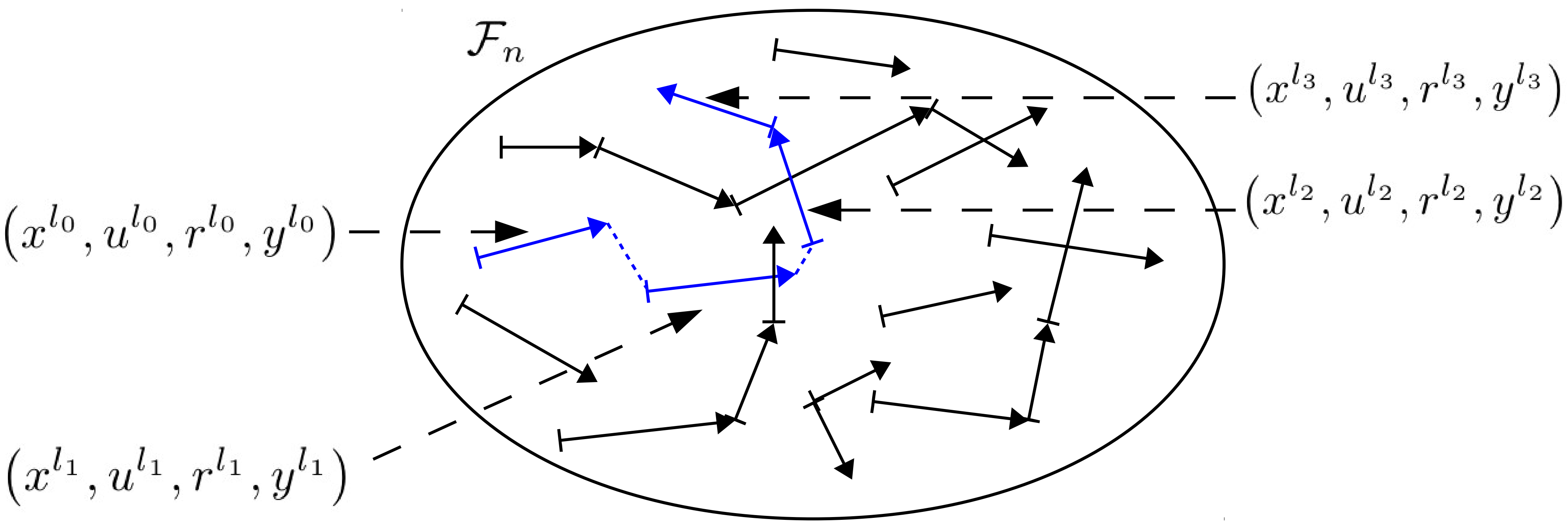
A New Approach: Synthesizing Artificial Trajectories

Artificial Trajectories

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

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

$$l_t \in \{1, \ldots, n\} \ , \qquad \forall t \in \{1, \ldots, T-1\}$$



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 Δ) 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:

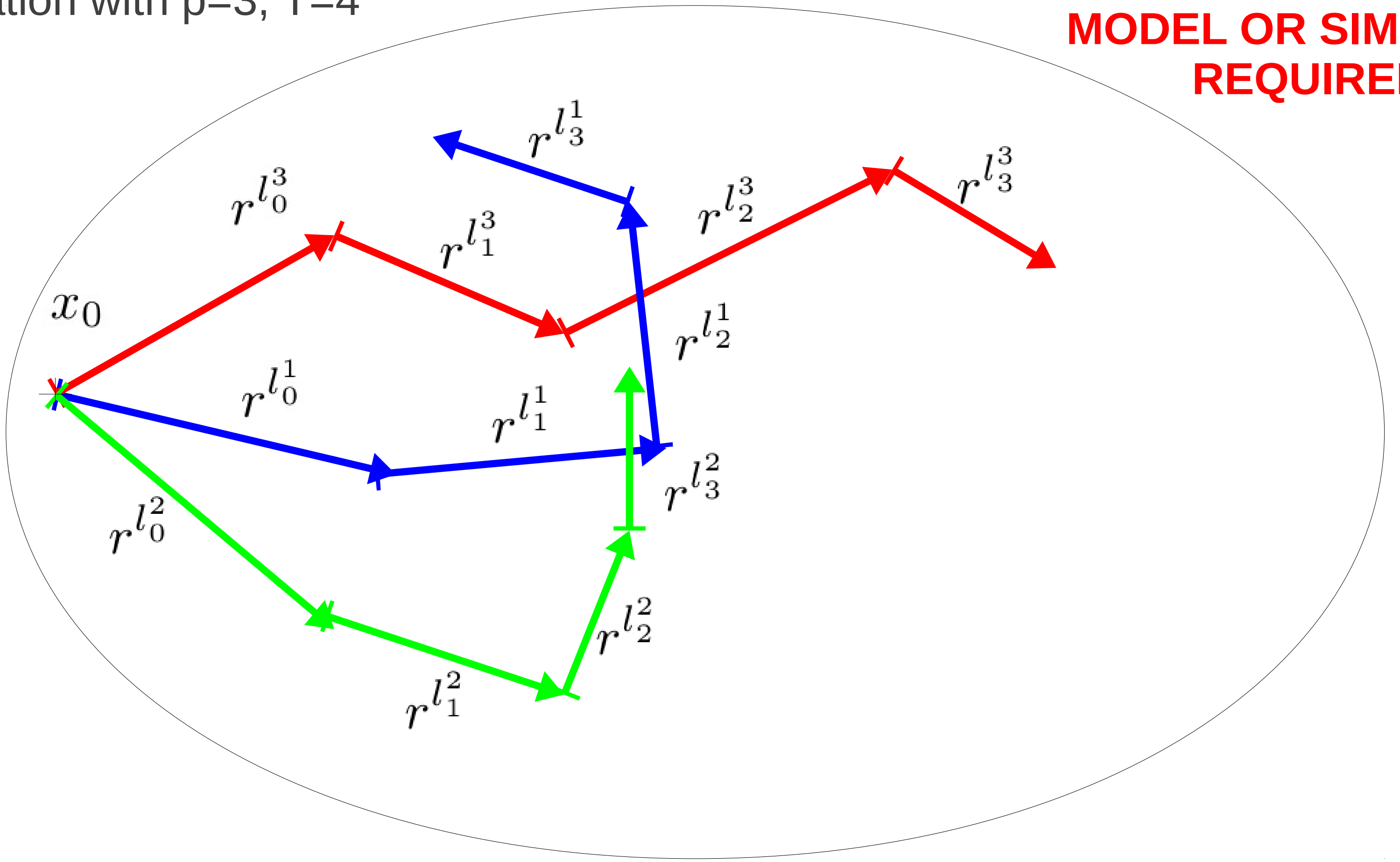
$$\mathfrak{M}_p^h(\mathcal{F}_n, x_0) = \frac{1}{p}\sum_{i=1}^{p}\sum_{t=0}^{T-1} r^{l_t^i}$$

Estimating the Performances of Policies

Monte Carlo Estimator

- Illustration with p=3, T=4

MODEL OR SIMULATOR REQUIRED !

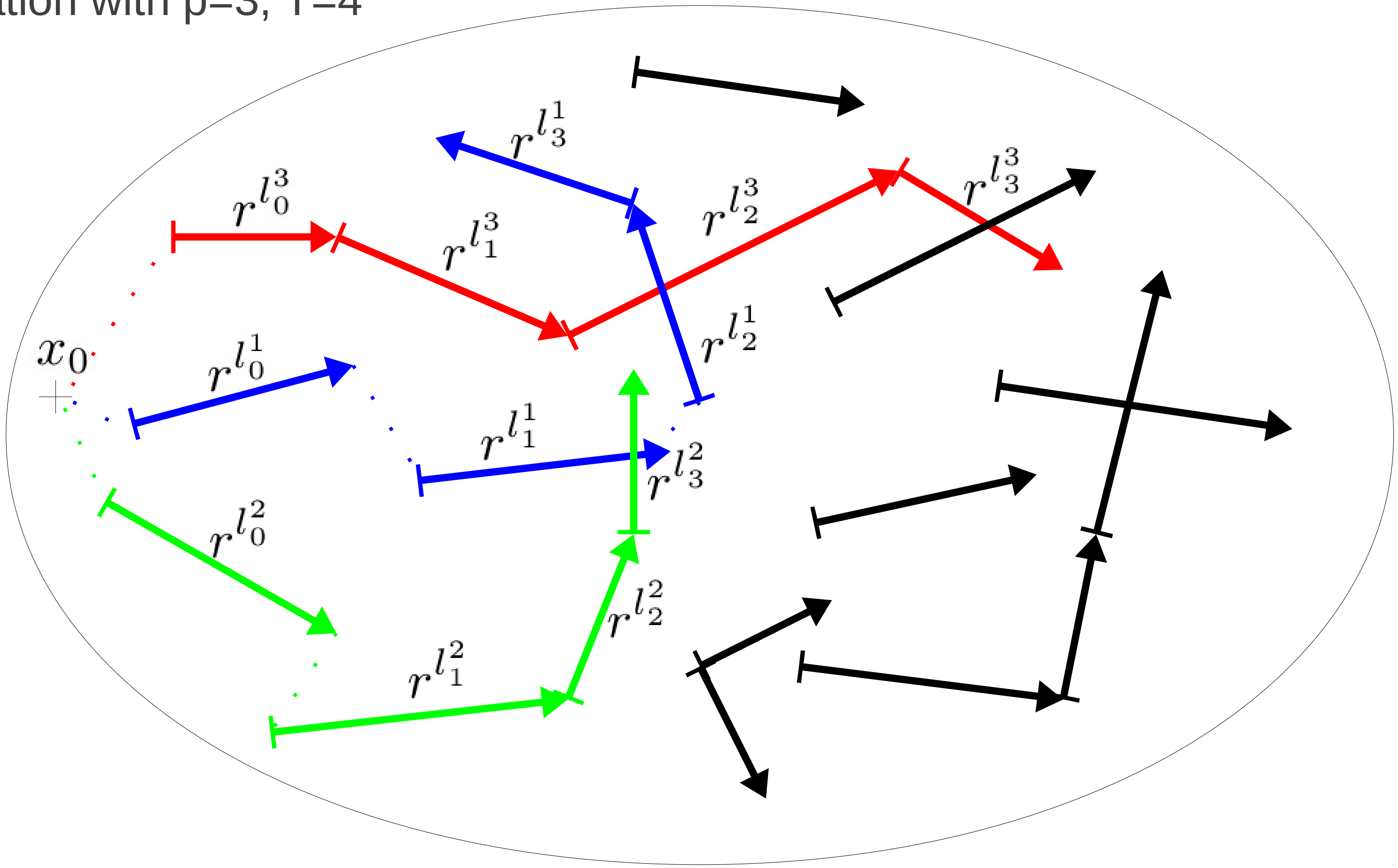


$$\mathbb{M}_3^h(x_0) = \frac{\left(r^{l_0^1} + r^{l_1^1} + r^{l_2^1} + r^{l_3^1}\right) + \left(r^{l_0^2} + r^{l_1^2} + r^{l_2^2} + r^{l_3^2}\right) + \left(r^{l_0^3} + r^{l_1^3} + r^{l_2^3} + r^{l_3^3}\right)}{3}$$

Estimating the Performances of Policies

Model-free Monte Carlo Estimator

- Illustration with p=3, T=4



$$\mathfrak{M}_3^h\left(\mathcal{F}_n, x_0\right) = \frac{\left(r^{l_0^1} + r^{l_1^1} + r^{l_2^1} + r^{l_3^1}\right) + \left(r^{l_0^2} + r^{l_1^2} + r^{l_2^2} + r^{l_3^2}\right) + \left(r^{l_0^3} + r^{l_1^3} + r^{l_2^3} + r^{l_3^3}\right)}{3}$$

Estimating the Performances of Policies

Additionnal Assumptions

Assumption: Lipschitz continuity of the functions f, ρ and h.
$\forall\ (x, x', u, u', w) \in \mathcal{X}^2 \times \mathcal{U}^2 \times \mathcal{W}$,

$$\begin{aligned}
\|f(x,u,w) - f(x',u',w)\|_{\mathcal{X}} &\leq L_f(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}), \\
|\rho(x,u,w) - \rho(x',u',w)| &\leq L_\rho(\|x-x'\|_{\mathcal{X}} + \|u-u'\|_{\mathcal{U}}), \\
\|h(t,x) - h(t,x')\|_{\mathcal{U}} &\leq L_h\|x-x'\|_{\mathcal{X}} \ , \forall t \in \{0, \ldots, T-1\}
\end{aligned}$$

Definition (Distance Metric Δ)

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

Definition ($k-$Dispersion)

$$\alpha_k(\mathcal{P}_n) = \sup_{(x,u)\in\mathcal{X}\times\mathcal{U}} \Delta_k^{\mathcal{P}_n}(x,u) \ ,$$

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

Estimating the Performances of Policies

Theoretical Results

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

$$\left|J^h(x_0) - E^h_{p,\mathcal{P}_n}(x_0)\right| \leq C\alpha_{pT}\left(\mathcal{P}_n\right)$$

$$with\ C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left(L_f(1+L_h)\right)^i$$

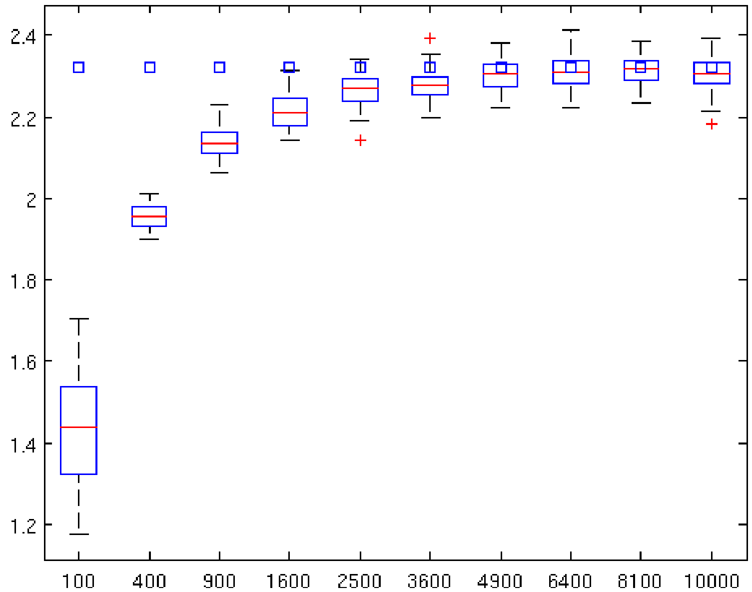
Theorem (**Variance Bound for** $\mathfrak{M}_p^h\left(\tilde{\mathcal{F}}_n, x_0\right)$)

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

$$with\ C = L_\rho \sum_{t=0}^{T-1} \sum_{i=0}^{T-t-1} \left(L_f(1+L_h)\right)^i$$

Estimating the Performances of Policies

Experimental Results

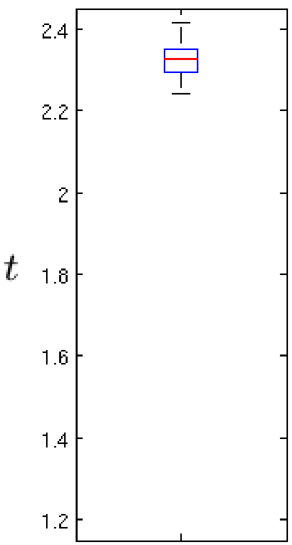


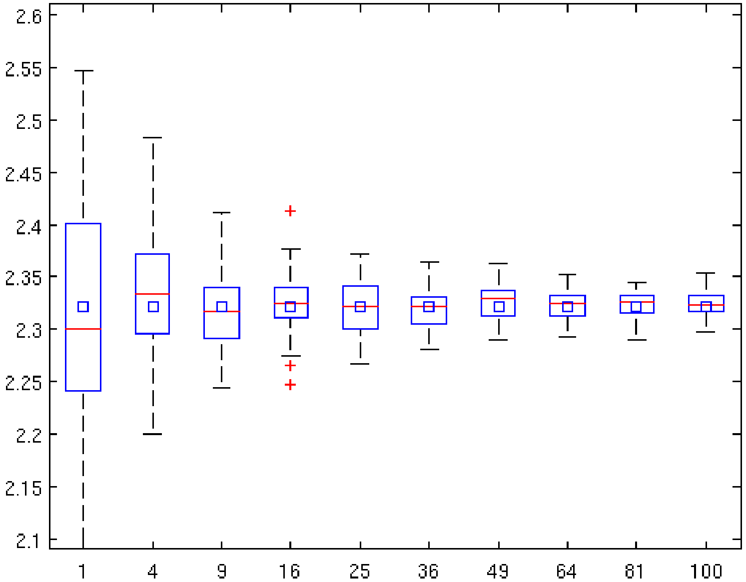
$$x_{t+1} = \sin\left(\frac{\pi}{2}(x_t + u_t + w_t)\right)$$

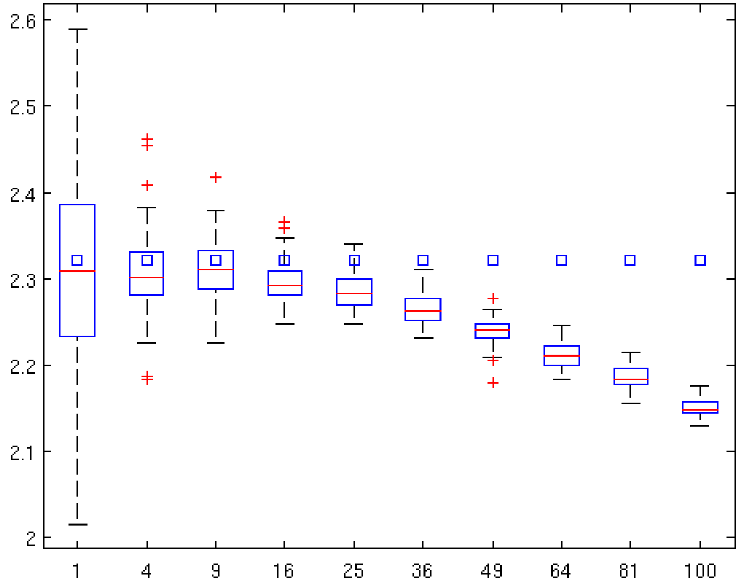
$$\rho(x_t, u_t, w_t) = \frac{1}{2\pi} e^{-\frac{1}{2}(x_t^2 + u_t^2)} + w_t$$

$$h(t, x) = -\frac{x}{2} \qquad x_0 = -0.5$$

$$\mathcal{W} = [-\tfrac{\epsilon}{2}, \tfrac{\epsilon}{2}]$$







Estimating the Performances of Policies

Value-at-Risk

- 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:

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

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

$$b \in \mathbb{R}$$

$$c \in [0, 1[$$

Deterministic Case: Computing Bounds

Lower Bound from a Single Trajectory

Proposition (Lower Bound from any Artificial Trajectory)

Let $\left[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})\right]_{t=0}^{T-1}$ *be any artificial trajectory. Then,*

$$J^h(x_0) \geq \sum_{t=0}^{T-1} r^{l_t} - \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta\left((y^{l_{t-1}}, h(t, y^{l_{t-1}})), (x^{l_t}, u^{l_t})\right)$$

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)**

$$L^h(\mathcal{F}_n, x_0) = \max_{[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})]_{t=0}^{T-1} \in \mathcal{F}_n^T} \sum_{t=0}^{T-1} r^{l_t} - \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta\left((y^{l_{t-1}}, h(t, y^{l_{t-1}})), (x^{l_t}, u^{l_t})\right)$$

Definition **(Minimal Upper Bound)**

$$U^h(\mathcal{F}_n, x_0) = \min_{[(x^{l_t}, u^{l_t}, r^{l_t}, y^{l_t})]_{t=0}^{T-1} \in \mathcal{F}_n^T} \sum_{t=0}^{T-1} r^{l_t} + \sum_{t=0}^{T-1} L_{Q_{T-t}} \Delta\left((y^{l_{t-1}}, h(t, y^{l_{t-1}})), (x^{l_t}, u^{l_t})\right)$$

Deterministic Case: Computing Bounds

Tightness of Maximal Bounds

Proposition (Tightness of the Bounds)

$$\exists C_b > 0 : \quad \begin{aligned} J^h(x_0) - L^h(\mathcal{F}_n, x_0) &\leq C_b \alpha_1(\mathcal{P}_n) \\ U^h(\mathcal{F}_n, x_0) - J^h(x_0) &\leq C_b \alpha_1(\mathcal{P}_n) \end{aligned}$$

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:

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

- 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 $\hat{\pi}^*_{\mathcal{F}_n,x_0}$)
Let $\mathfrak{J}^*(x_0)$ be the set of optimal open-loop policies:

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

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

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

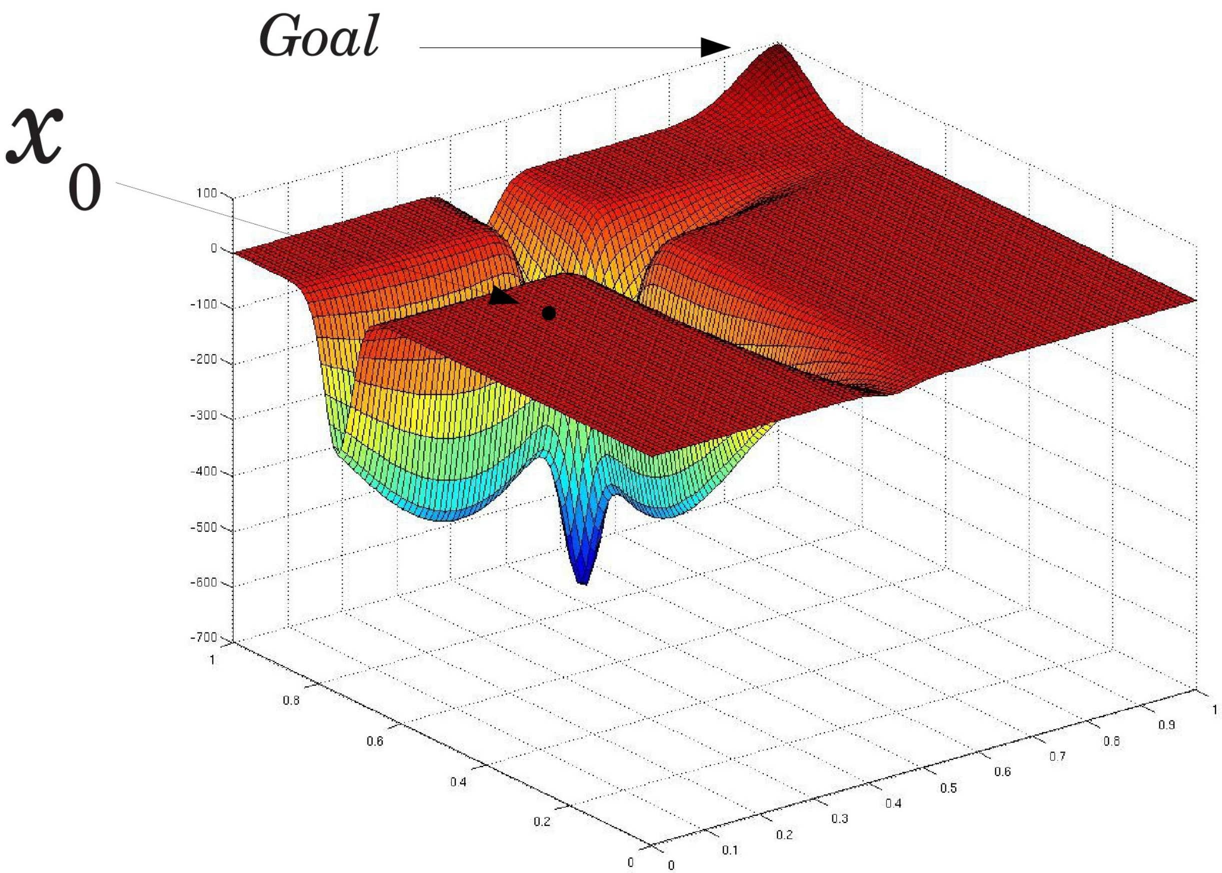
Then,

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

Inferring Safe Policies

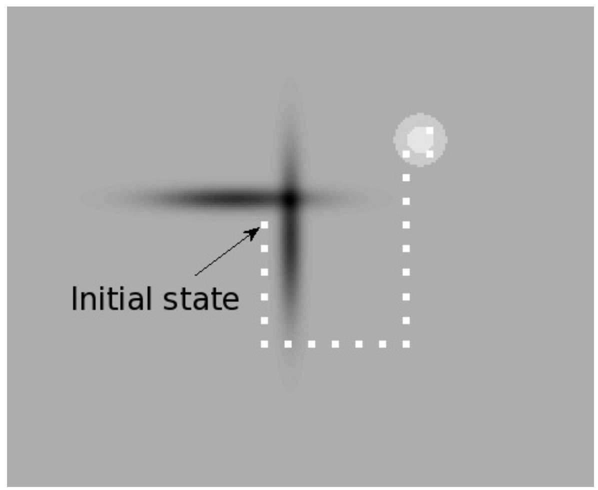
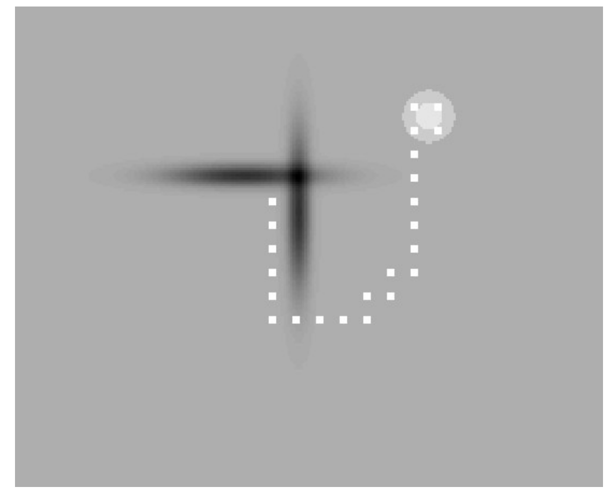
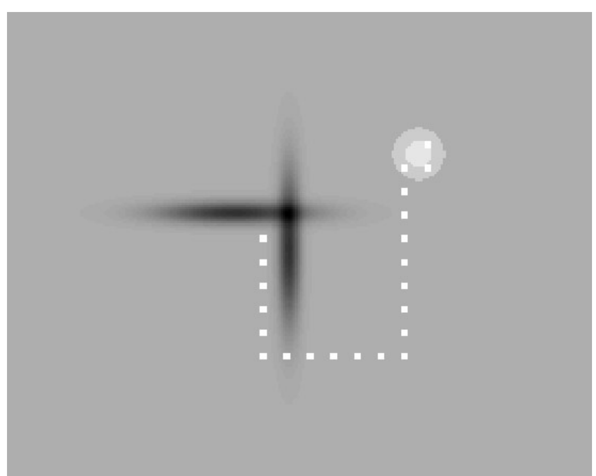
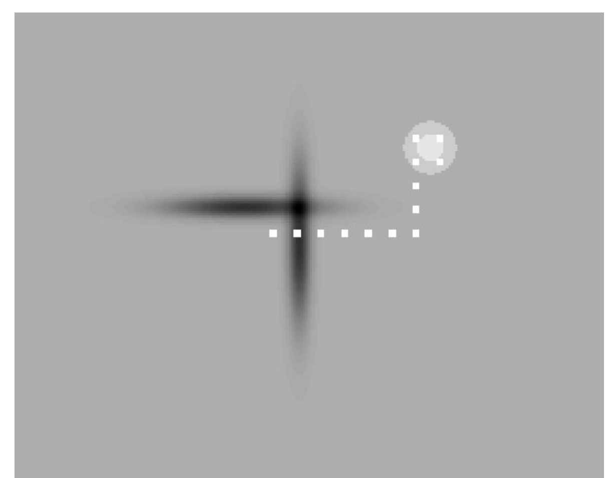
Experimental Results

- The puddle world benchmark



Inferring Safe Policies

Experimental Results

	CGRL	FQI (Fitted Q Iteration)
The state space is uniformly covered by the sample		
Information about the Puddle area is removed		

Inferring Safe Policies

Bonus

Theorem (Optimal Policies computed from Optimal Trajectories)

Let $\pi^*_{x_0} \in \mathfrak{J}^*(x_0)$ *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*

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

such that

$$\begin{aligned} x^{l_0} &= x_0 \ , \\ u^{l_t} &= \pi^*_{x_0}(t) \qquad \forall t \in \{0, \ldots, T-1\} \ . \end{aligned}$$

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

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

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

Sampling Strategies

An Artificial Trajectories Viewpoint

- Given a sample of system transitions

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

How can we determine where to sample additional transitions ?

- We define the set of candidate optimal policies:

$$\Pi(\mathcal{F}, x_0) = \left\{ \pi \in \Pi \quad | \quad \forall \pi' \in \Pi, U^{\pi}(\mathcal{F}, x_0) \geq L^{\pi'}(\mathcal{F}, x_0) \right\}$$

- 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

$$\forall (x^l, u^l, r^l, y^l) \in \mathcal{F}, \quad (u^l = u) \implies \begin{cases} \left| r - r^l \right| & \leq L_\rho \| x - x^l \|_{\mathcal{X}}, \\ \left\| y - y^l \right\|_{\mathcal{X}} & \leq L_f \| x - x^l \|_{\mathcal{X}} \end{cases}$$

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

Sampling Strategies

An Artificial Trajectories Viewpoint

- Iterative scheme:

$$(x^{m+1}, u^{m+1}) \in \underset{(x,u) \in \mathcal{X} \times \mathcal{U}}{\arg\min} \left\{ \max_{\substack{(r,y) \in \mathbb{R} \times \mathcal{X} \;\; s.t. (x,u,r,y) \in \mathcal{C}(\mathcal{F}_m) \\ \pi \in \Pi(\mathcal{F}_m \cup \{(x,u,r,y)\}, x_0)}} \delta^{\pi}(\mathcal{F}_m \cup \{(x,u,r,y)\}, x_0) \right\}$$

with

$$\delta^{\pi}(\mathcal{F}, x_0) = U^{\pi}(\mathcal{F}, x_0) - L^{\pi}(\mathcal{F}, x_0)$$

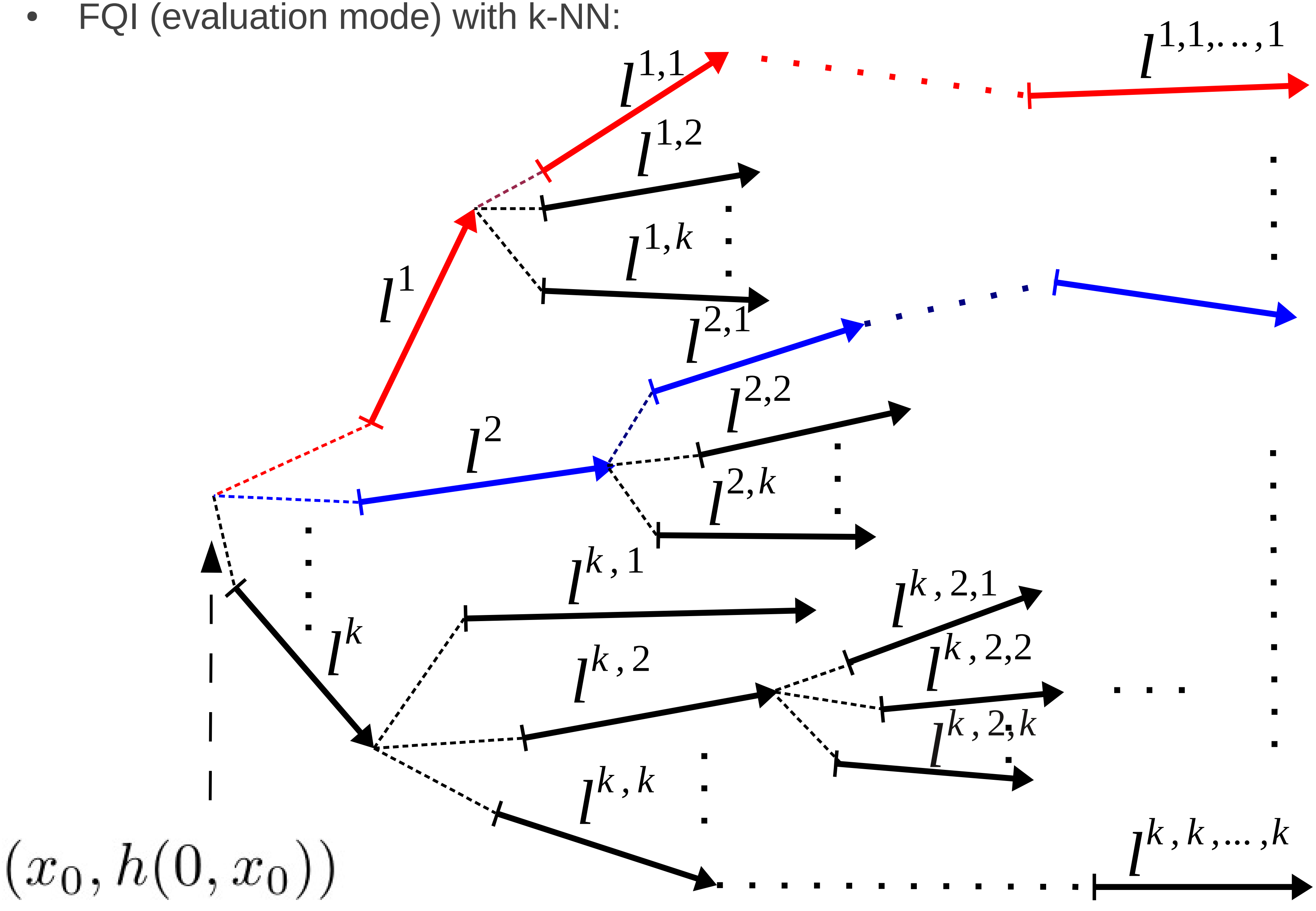
- Conjecture:

$$\exists m_0 \in \mathbb{N} \setminus \{0\} : \forall m \in \mathbb{N}, \left(m \geq m_0 \right) \implies \Pi\left(\mathcal{F}_m, x_0\right) = \mathfrak{J}^*(x_0)$$

Connexion to Classic Batch Mode RL

Towards a New Paradigm for Batch Mode RL

- FQI (evaluation mode) with k-NN:



Connexion to Classic Batch Mode RL

Towards a New Paradigm for Batch Mode RL

Proposition ($k-$**NN FQI-PE using Artificial Trajectories**)

$$\hat{J}^h_{FQI}(\mathcal{F}_n, x_0) = \frac{1}{k^T} \sum_{i_0=1}^{k} \ldots \sum_{i_{T-1}=1}^{k} \left(r^{l^{i_0}} + r^{l^{i_0,i_1}} + \ldots + r^{l^{i_0,i_1,\ldots,i_{T-1}}} \right) .$$

where the set of rebuilt artificial trajectories

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

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

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

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.

"Generating informative trajectories by using bounds on the return of control policies". R. Fonteneau, S.A. Murphy, L. Wehenkel and D. Ernst. Proceedings of the Workshop on Active Learning and Experimental Design 2010 (in conjunction with AISTATS 2010), 2-page highlight paper, Chia Laguna, Sardinia, Italy, May 16, 2010.

"Model-free Monte Carlo-like policy evaluation". R. Fonteneau, S.A. Murphy, L. Wehenkel and D. Ernst. In Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010), JMLR W&CP 9, pp 217-224, Chia Laguna, Sardinia, Italy, May 13-15, 2010.

"A cautious approach to generalization in reinforcement learning". R. Fonteneau, S.A. Murphy, L. Wehenkel and D. Ernst. Proceedings of The International Conference on Agents and Artificial Intelligence (ICAART 2010), 10 pages, Valencia, Spain, January 22-24, 2010.

"Inferring bounds on the performance of a control policy from a sample of trajectories". R. Fonteneau, S.A. Murphy, L. Wehenkel and D. Ernst. In Proceedings of The IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009), 7 pages, Nashville, Tennessee, USA, 30 March-2 April, 2009.