ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

POLITECNICO
MILANO 1863

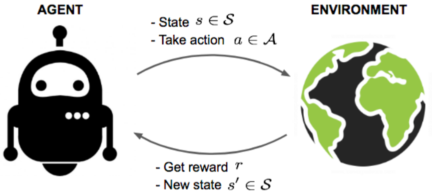
Handling Non-Stationary Experts in Inverse Reinforcement Learning: A Water System Control Case Study

A. Likmeta A. M. Metelli G. Ramponi A. Tirinzoni
M. Giuliani M. Restelli

January 15, 2021
M2L Summer School

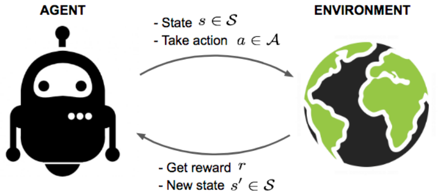
Motivations and Goal

- **RL**: Maximize sum of rewards



Motivations and Goal

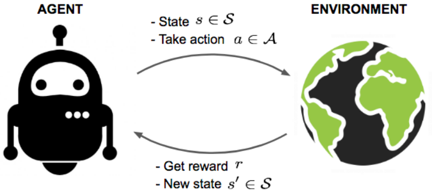
- **RL**: Maximize sum of rewards



! Reward function hard to define.

Motivations and Goal

- **RL**: Maximize sum of rewards

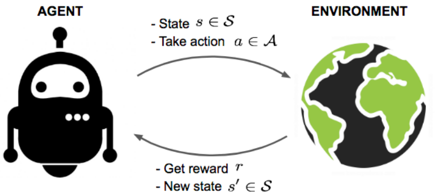


! Reward function hard to define.

! Demonstrations often available.

Motivations and Goal

- **RL**: Maximize sum of rewards



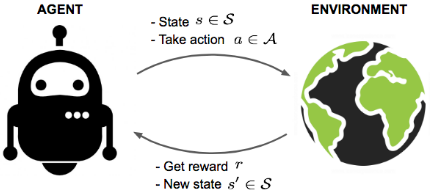
! Reward function hard to define.

! Demonstrations often available.

- **IRL**: Recovers **unknown reward** from expert demonstrations

Motivations and Goal

- **RL**: Maximize sum of rewards



! Reward function hard to define.

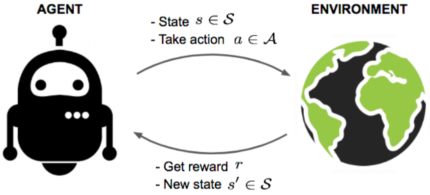
! Demonstrations often available.

- **IRL**: Recovers **unknown reward** from expert demonstrations

! Using only demonstrations.

Motivations and Goal

- **RL**: Maximize sum of rewards



! Reward function hard to define.

! Demonstrations often available.

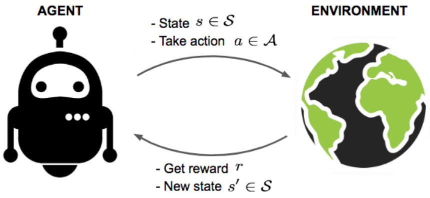
- **IRL**: Recovers **unknown reward** from expert demonstrations

! Using only demonstrations.

! No interactions with environment.

Motivations and Goal

- **RL**: Maximize sum of rewards



! Reward function hard to define.

! Demonstrations often available.

- **IRL**: Recovers **unknown reward** from expert demonstrations

! Using only expert demonstrations.

! No interactions with environment.

! Non-stationarity in the objectives.

Problem setting: IRL

- MDP without Reward (MDP$\backslash R$) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma, \mu)$ (Puterman, 1994)

Problem setting: IRL

- MDP without Reward (MDP$\backslash R$) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma, \mu)$ (Puterman, 1994)
- Linearly parameterized rewards: $R_{\boldsymbol{\omega}}(s, a) = \boldsymbol{\phi}(s, a)^T \boldsymbol{\omega}$

Problem setting: IRL

- MDP without Reward (MDP$\backslash R$) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma, \mu)$ (Puterman, 1994)
- Linearly parameterized rewards: $R_{\boldsymbol{\omega}}(s, a) = \boldsymbol{\phi}(s, a)^T \boldsymbol{\omega}$
- Parametrized policy $\pi_{\boldsymbol{\theta}}$:

$$J_\pi(\boldsymbol{\theta}, \boldsymbol{\omega}) = \mathop{\mathbb{E}}_{\substack{s_0 \sim \mu \\ a_t \sim \pi_{\boldsymbol{\theta}}}} \left[\sum_{t=0}^{+\infty} \gamma^i R_{\boldsymbol{\omega}}(s_t, a_t) \right] = \boldsymbol{\omega}^T \boldsymbol{\psi}(\boldsymbol{\theta})$$

General overview

1. Infer reward functions only from demonstrations
2. Handle Non-Stationarity in demonstrations
3. Use Case:Como Lake Water Control System

General overview

1. Infer reward functions only from demonstrations
2. Handle Non-Stationarity in demonstrations
3. Use Case:Como Lake Water Control System

Infer reward functions from demonstrations

- No interaction with the environment and no forward learning

Infer reward functions from demonstrations

- No interaction with the environment and no forward learning

Idea: if the expert's policy is optimal then $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) = 0$

Infer reward functions from demonstrations

- No interaction with the environment and no forward learning

Idea: if the expert's policy is optimal then $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) = 0$

Σ-Gradient Inverse Reinforcement Learning (Ramponi et al., 2020)

$$\min_{\|\boldsymbol{\omega}\|_1 = 1} \left\| \widehat{\nabla}_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) \boldsymbol{\omega} \right\|^2_{\left[(\boldsymbol{\omega} \otimes \mathbf{I}_d)^T \boldsymbol{\Sigma} (\boldsymbol{\omega} \otimes \mathbf{I}_d)\right]^{-1}}$$

Infer reward functions from demonstrations

- No interaction with the environment and no forward learning

Idea: if the expert's policy is optimal then $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) = 0$

Σ-Gradient Inverse Reinforcement Learning (Ramponi et al., 2020)

$$\min_{\|\boldsymbol{\omega}\|_1=1} \left\| \widehat{\nabla}_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) \boldsymbol{\omega} \right\|^2_{\left[(\boldsymbol{\omega} \otimes \mathbf{I}_d)^T \boldsymbol{\Sigma} (\boldsymbol{\omega} \otimes \mathbf{I}_d)\right]^{-1}}$$

! Need the parametrized expert policy (perform Behavioral Cloning).

Infer reward functions from demonstrations

- No interaction with the environment and no forward learning

Idea: if the expert's policy is optimal then $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) = 0$

Σ-Gradient Inverse Reinforcement Learning (Ramponi et al., 2020)

$$\min_{\|\boldsymbol{\omega}\|_1 = 1} \left\| \widehat{\nabla}_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) \boldsymbol{\omega} \right\|^2_{\left[(\boldsymbol{\omega} \otimes \mathbf{I}_d)^T \boldsymbol{\Sigma} (\boldsymbol{\omega} \otimes \mathbf{I}_d)\right]^{-1}}$$

! Need the parametrized expert policy (perform Behavioral Cloning).

! Gradients are estimated from demonstrations
$\rightarrow$ could not exist a $\boldsymbol{\omega}$ s.t. $\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) = 0$

Infer reward functions from demonstrations

- No interaction with the environment and no forward learning

Idea: if the expert's policy is optimal then $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) = 0$

Σ-Gradient Inverse Reinforcement Learning (Ramponi et al., 2020)

$$\min_{\|\boldsymbol{\omega}\|_1 = 1} \left\| \widehat{\nabla}_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}) \boldsymbol{\omega} \right\|^2_{\left[(\boldsymbol{\omega} \otimes \mathbf{I}_d)^T \boldsymbol{\Sigma} (\boldsymbol{\omega} \otimes \mathbf{I}_d)\right]^{-1}}$$

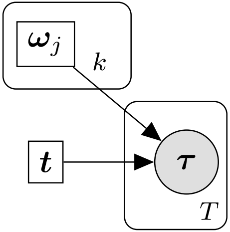
! Need the parametrized expert policy (perform Behavioral Cloning).

! Gradients are estimated from demonstrations
  - $\rightarrow$ could not exist a $\boldsymbol{\omega}$ s.t. $\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{\omega}) = 0$
  - $\rightarrow$ account for the uncertainty in the Jacobian estimation
  - $\rightarrow$ allows computing likelihood of D given $\boldsymbol{\omega}$, $p(D|\boldsymbol{\omega})$

General overview

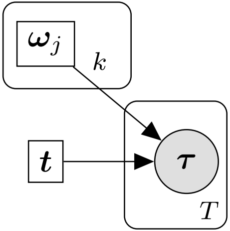
1. Infer reward functions only from demonstrations
2. Handle Non-Stationarity in demonstrations
3. Use Case:Como Lake Water Control System

Problem setting: Non-Stationary IRL



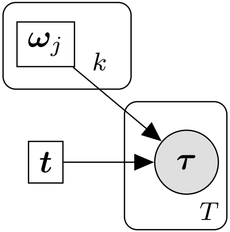
- Consider dataset $D = \{\tau_i\}_{i=1}^{T}$ as a lifelong trajectory $\boldsymbol{\tau} = (\tau_1 | \dots | \tau_T)$

Problem setting: Non-Stationary IRL



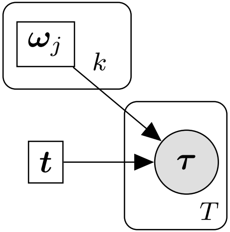
- Consider dataset $D = \{\tau_i\}_{i=1}^T$ as a lifelong trajectory $\boldsymbol{\tau} = (\tau_1 | \dots | \tau_T)$
- Non-stationary behavior characterized by $k \leqslant T$ *regimes*

Problem setting: Non-Stationary IRL



- Consider dataset $D = \{\tau_i\}_{i=1}^{T}$ as a lifelong trajectory $\boldsymbol{\tau} = (\tau_1 | \dots | \tau_T)$
- Non-stationary behavior characterized by $k \leqslant T$ *regimes*
- Find change points $1 < t_1 < \cdots < t_{k-1} < t_k = T$

Problem setting: Non-Stationary IRL



- Consider dataset $D = \{\tau_i\}_{i=1}^T$ as a lifelong trajectory $\boldsymbol{\tau} = (\tau_1|\dots|\tau_T)$
- Non-stationary behavior characterized by $k \leqslant T$ *regimes*
- Find change points $1 < t_1 < \cdots < t_{k-1} < t_k = T$
- Find rewards $\mathbf{R} = (R_{\boldsymbol{\omega}_1}, \dots, R_{\boldsymbol{\omega}_k})$

Non Stationary-Σ-GIRL

Idea: Take inspiration from *change-point detection* (Aminikhanghahi and Cook, 2017)

Idea: Take inspiration from *change-point detection* (Aminikhanghahi and Cook, 2017)

NS-Σ-GIRL (adaptation of *Opt* (Truong et al., 2020))

1. Define Likelihood of $\boldsymbol{\tau}$ under solution $\Omega = (\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_k, t_1, \dots, t_{k-1})$

$$\mathcal{L}(\Omega|\boldsymbol{\tau}) = p(\boldsymbol{\tau}|\Omega) = \prod_{i=1}^{T} \sum_{j=1}^{k} p(\tau_i|\boldsymbol{\omega}_j) \mathbb{1}_{\{i \in I_j\}}$$

2. For each $1 \leqslant u < v \leqslant T$ solve:

$$\min_{\substack{\boldsymbol{\omega}_{uv} \in \mathbb{R}^q_{\geqslant 0} \\ \|\boldsymbol{\omega}_{uv}\|_1 = 1}} (v-u) \sum_{i=u}^{v-1} \left\| \widehat{\nabla}_{\boldsymbol{\theta}} \psi_i(\boldsymbol{\theta}) \boldsymbol{\omega}_{uv} \right\|^2_{[(\boldsymbol{\omega}_{uv} \otimes \mathbf{I}_d) \boldsymbol{\Sigma}_i (\boldsymbol{\omega}_{uv} \otimes \mathbf{I}_d)^T]^{-1}} .$$

General overview

1. Infer reward functions only from demonstrations
2. Handle Non-Stationarity in demonstrations
3. Use Case:Como Lake Water Control System

System Modelling

- **Problem:** Infer operator intentions from hystorical dam operation.

System Modelling

- **Problem:** Infer operator intentions from hystorical dam operation.
- **Problem:** The intentions of the operators change during 60 years

System Modelling

- **Problem:** Infer operator intentions from hystorical dam operation.
- **Problem:** The intentions of the operators change during 60 years
- Continuous state: water stored in the lake S_t, a continuous action: water released a_t, a state-transition function of: lake inflow q_{t+1}

$$S_{t+1} = S_t + q_{t+1} - r_{t+1}(S_t, a_t, q_{t+1})$$

- **Problem:** Infer operator intentions from hystorical dam operation.
- **Problem:** The intentions of the operators change during 60 years
- Continuous state: water stored in the lake S_t, a continuous action: water released a_t, a state-transition function of: lake inflow q_{t+1}

$$S_{t+1} = S_t + q_{t+1} - r_{t+1}(S_t, a_t, q_{t+1})$$

- Data are daily and were provided by Consorzio dell'Adda (`www.addaconsorzio.it`)

Reward Function

- Three reward features representing conflicting objectives:

Reward Function

- Three reward features representing conflicting objectives:
- *Supply deficit* - (ϕ^D): deficit between the release and the demand

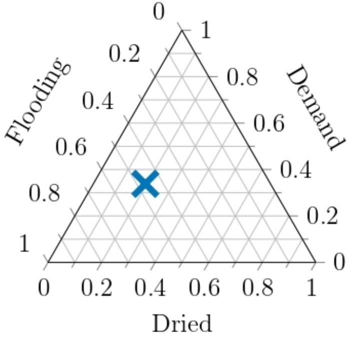
Reward Function

- Three reward features representing conflicting objectives:
- *Supply deficit* - (ϕ^D): deficit between the release and the demand
- *Flood risk* (ϕ^F): penalize small releases associated to high lake levels

Reward Function

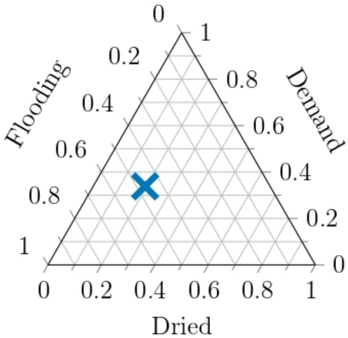
- Three reward features representing conflicting objectives:
- *Supply deficit* - (ϕ^D): deficit between the release and the demand
- *Flood risk* (ϕ^F): penalize small releases associated to high lake levels
- *Drought risk* (ϕ^L): penalize large releases with low lake levels

Single Reward Setting



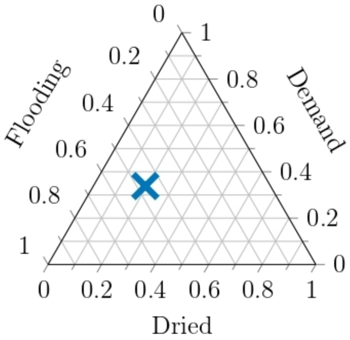
- Slight predominance for controlling the floods ($\omega^F = 0.47$)

Single Reward Setting



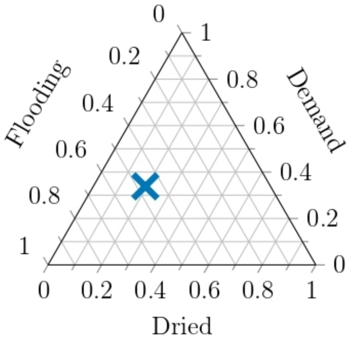
- Slight predominance for controlling the floods ($\omega^F = 0.47$)
- Remaining weight is divided between demand ($\omega^D = 0.34$) and drought control ($\omega^L = 0.19$)

Single Reward Setting



- Slight predominance for controlling the floods ($\omega^F = 0.47$)
- Remaining weight is divided between demand ($\omega^D = 0.34$) and drought control ($\omega^L = 0.19$)
- Results in line with literature results (Giuliani et al., 2019)

Single Reward Setting



- Slight predominance for controlling the floods ($\omega^F = 0.47$)
- Remaining weight is divided between demand ($\omega^D = 0.34$) and drought control ($\omega^L = 0.19$)
- Results in line with literature results (Giuliani et al., 2019)
- Expert almost Pareto optimal

Single Reward Setting

Recent Flooding (Como, 2020)

- Slight predominance for controlling the floods ($\omega^F = 0.47$)
- Remaining weight is divided between demand ($\omega^D = 0.34$) and drought control ($\omega^L = 0.19$)
- Results in line with literature results (Giuliani et al., 2019)
- Expert almost Pareto optimal

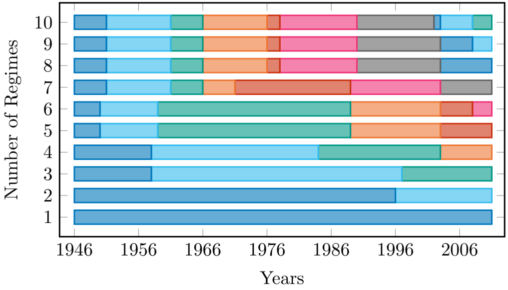
Non-Stationary Reward Setting

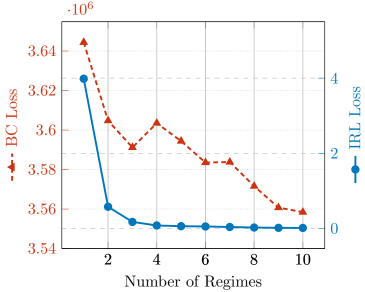
- **Problem:** Lake Como is a **non-stationary** system that has undergone several alterations

Non-Stationary Reward Setting

- **Problem:** Lake Como is a **non-stationary** system that has undergone several alterations
- **Idea:** Consider the dataset as a lifelong trajectory. Make subdivisions yearly. Find K **intention change points**.

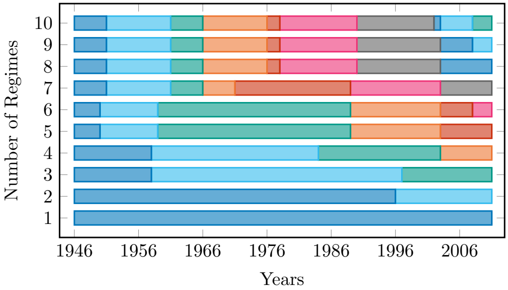
- **Problem:** Lake Como is a **non-stationary** system that has undergone several alterations
- **Idea:** Consider the dataset as a lifelong trajectory. Make subdivisions yearly. Find K **intention change points**.

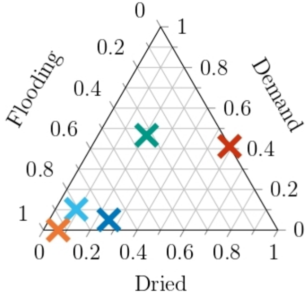




Non-Stationary Reward Setting

- **Problem:** Lake Como is a **non-stationary** system that has undergone several alterations
- **Idea:** Consider the dataset as a lifelong trajectory. Make subdivisions yearly. Find K **intention change points**.





Conclusions

- Extended Σ-GIRL to the non-stationary setting
- Carefully designed reward features that explain the expert behaviour
- Indentified several hystorical regimes of interest

Thank You for Your Attention!

References

Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

Qui Como. Il lago è fuori: le nuove barriere anti-esondazione, October 2020. URL `https://www.quicomo.it/attualita/lago-di-como-fuori-3-ottobre-2020.html`.

Matteo Giuliani, Marta Zaniolo, Andrea Castelletti, Guido Davoli, and Paul Block. Detecting the state of the climate system via artificial intelligence to improve seasonal forecasts and inform reservoir operations. *Water Resources Research*, 55:9133–9147, 2019.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994. ISBN 0471619779.

Giorgia Ramponi, Amarildo Likmeta, Alberto Maria Metelli, Andrea Tirinzoni, and Marcello Restelli. Truly batch model-free inverse reinforcement learning about multiple intentions. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2359–2369, Online, 26–28 Aug 2020. PMLR. URL `http://proceedings.mlr.press/v108/ramponi20a.html`.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.