Exploiting Environment Configuration for Policy Space Identification

Guglielmo Manneschi

Supervisor: Marcello Restelli
Co-supervisor: Alberto Maria Metelli

Politecnico di Milano
M.Sc. in Computer Science and Engineering

30th September 2019

Outline

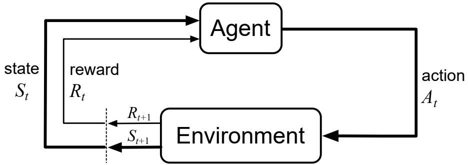
- Introduction
- Policy Space Identification
- Exploiting environment configuration
- Experimental evaluation
- Applications

Introduction

Markov Decision Process

Introduction

Framework to model **sequential decision-making** problems [Puterman, 2014].



Markov Decision Process
Policy

Policy:

$$\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$$

Performance measure:

$$J_\pi = \mathop{\mathbb{E}}_{\tau \sim p_\pi} \left[\sum_{t=1}^{T(\tau)} \gamma^t r_{\tau,t} \right]$$

Markov Decision Process

Policy functions

Parametric policies [Sutton and Barto, 2011]:

- The policy is defined by a vector of parameters $\boldsymbol{\theta}$: $\pi_{\boldsymbol{\theta}}(a|\phi(s))$
- Useful for large (or infinite) state spaces
- Each state is represented by a **feature vector** $\phi(s)$, where $\phi : \mathcal{S} \to \mathbb{R}^q$
  - Perceptions of the agent

Policy search
Policy space

The *policy space* is the class of all the representable policies:

$$\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d\},$$

where Θ is the space of the parameters.

Policy search
Learning

Solution of an MDP:

- Find a policy that maximizes the performance

$$\boldsymbol{\theta}^* \in \arg\max_{\boldsymbol{\theta} \in \Theta} J_{\boldsymbol{\theta}}$$

- Search inside the policy space Π_Θ
- Gradient based approach [Deisenroth et al., 2013]

Policy Space Identification

Goal

We want to **identify the policy space** of an agent:

- by observing demonstrations coming from the optimal policy
- assuming that the policy space of the agent is a subset of a known super-space:
    - a policy is determined by a d-dimensional vector $\boldsymbol{\theta} \in \Theta$
    - the agent can control only $d^* < d$ parameters
- identifying which parameters it can control

Outline

- Introduction
- Policy Space Identification
- Exploiting environment configuration
- Experimental evaluation
- Applications

Correctness

- Let $I \subseteq \{1, ..., d\}$
- Let $\Theta_I = \{\boldsymbol{\theta} \in \Theta : \theta_i = 0, \forall i \in \{1, ..., d\} \setminus I\}$, i.e., I is the set of **indexes that can be changed** by the agent if the parameter space were Θ_I.
- Let $\pi^* \in \Pi_\Theta$

A set of parameter indexes $I^* \subseteq \{1, ..., d\}$ is *correct* w.r.t. π^* if:

$$\pi^* \in \Pi_{\Theta_{I^*}} \tag{1}$$

$$\forall i \in I^* : \ \pi^* \notin \Pi_{\Theta_{I^* \setminus \{i\}}} \tag{2}$$

Combinatorial Identification Rule

Test **all the possible subsets** of parameters:

$$I \subseteq \{1, ..., d\}$$

For each I we consider the pair of hypotheses:

$$\mathcal{H}_{0,I} : \pi^* \in \Pi_{\Theta_I}$$
$$\mathcal{H}_{1,I} : \pi^* \in \Pi_{\Theta \setminus \Theta_I}$$

The GLR statistic [Lehmann and Romano, 2006] is:

$$\lambda_I = -2 \log \frac{\sup_{\boldsymbol{\theta} \in \Theta_I} \widehat{\mathcal{L}}(\boldsymbol{\theta})}{\sup_{\boldsymbol{\theta} \in \Theta} \widehat{\mathcal{L}}(\boldsymbol{\theta})}$$

Combinatorial Identification Rule

A correct subset must satisfy:

$$\lambda_I \leq c(|I|) \quad (1)$$

$$\forall i \in I : \ \lambda_{I \setminus \{i\}} > c(|I \setminus \{i\}|) \quad (2)$$

where $c(I)$ are the critical values.

Combinatorial Identification Rule

A correct subset must satisfy:

$$\lambda_I \leq c(|I|) \tag{1}$$

$$\forall i \in I: \ \lambda_{I \setminus \{i\}} > c(|I \setminus \{i\}|) \tag{2}$$

where $c(I)$ are the critical values.

Drawback: **exponential complexity** $\mathcal{O}(2^d)$

Identifiability Assumption

The policy space is *identifiable* if, for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, we have:

$$\pi_{\boldsymbol{\theta}} = \pi_{\boldsymbol{\theta}'} \text{ almost surely } \implies \boldsymbol{\theta} = \boldsymbol{\theta}'.$$

Under this assumption, there exists a **unique set of parameters** that is correct w.r.t. π^*.

Simplified Identification Rule

Under the identifiability assumption, we can test **one parameter at a time**.

For all $i \in \{1, ..., d\}$ we consider the pair of hypotheses:

$$\mathcal{H}_{0,i} : \theta_i^* = 0$$
$$\mathcal{H}_{1,i} : \theta_i^* \neq 0$$

and the GLR statistic:

$$\lambda_i = -2 \log \frac{\sup_{\boldsymbol{\theta} \in \Theta_i} \widehat{\mathcal{L}}(\boldsymbol{\theta})}{\sup_{\boldsymbol{\theta} \in \Theta} \widehat{\mathcal{L}}(\boldsymbol{\theta})},$$

where $\Theta_i = \{\boldsymbol{\theta} \in \Theta : \theta_i = 0\}$.

Simplified Identification Rule

The set of parameter indexes that defines the policy space is:

$$\widehat{I}_c = \{i \in \{1, ..., d\} : \lambda_i > c(1)\},$$

where $c(1)$ is the critical value.

This method has **linear complexity** $\mathcal{O}(d)$.

Theoretical analysis of the simplified identification rule:

- Bounds on first and second type error probabilities

Outline

- Introduction
- Policy Space Identification
- Exploiting environment configuration
- Experimental evaluation
- Applications

Policy Space Identification in Configurable Environment
Motivation

Main limitation of previous approach:

- distinguish when a parameter is **not controllable** or just **useless for the current task**

Policy Space Identification in Configurable Environment

Motivation

Main limitation of previous approach:

- distinguish when a parameter is **not controllable** or just **useless for the current task**

Solution:

- change the task

Policy Space Identification in Configurable Environment

Conf-MDP

Configurable MDP [Metelli et al., 2018]

- Extension of the classical MDP framework
- Allows the **configuration of the environment** with a vector or parameters $\boldsymbol{\omega}$ specifying:
  - transition model $\mathcal{P}_{\boldsymbol{\omega}}$
  - initial state distribution $\mu_{\boldsymbol{\omega}}$

Policy Space Identification in Configurable Environment

Conf-MDP

Configurable MDP [Metelli et al., 2018]

- Extension of the classical MDP framework
- Allows the **configuration of the environment** with a vector or parameters $\boldsymbol{\omega}$ specifying:
  - transition model $\mathcal{P}_{\boldsymbol{\omega}}$
  - initial state distribution $\mu_{\boldsymbol{\omega}}$
- Select a configuration in which the parameters to examine have an **optimal value different from zero**

Policy Space Identification in Configurable Environment

Algorithm

- Perform a first identification of the policy space, and obtain $\widehat{I}_0$
- After each identification update the estimated policy space: $\widehat{I} \leftarrow \widehat{I} \cup \widehat{I}_k$
- For each parameter $i \in \{1, ..., d\} : i \notin \widehat{I}$:
  - Find a new model ω_k
  - Collect data D_k observing $\pi^*(\omega_k)$
  - Perform an identification obtaining $\widehat{I}_k$ and update $\widehat{I}$

Outline

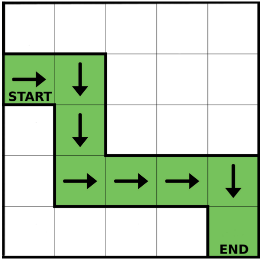
- Introduction
- Policy Space Identification
- Exploiting environment configuration
- Experimental evaluation
- Applications

Experiments

- Grid World
  - Error probability in configurable and fixed environment
- Continuous Gridworld
  - Error probability in configurable and fixed environment
  - Graphical configuration example
- Minigolf
  - Performance with different policy spaces
  - Benefits of knowing the policy space
- Car Driving
  - Without identifiability assumption

Grid World
Description

- Two-dimensional world (5x5 cells)
- Discrete actions in the four directions
- Binary features
- Softmax initial state distribution
    - initial agent position
    - goal position
    - configurable



Grid World

Error probability

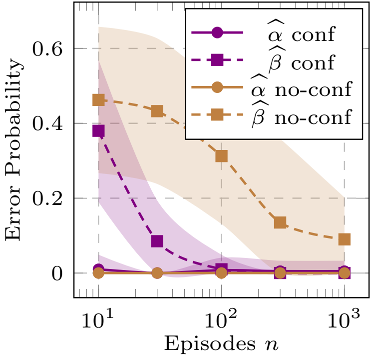
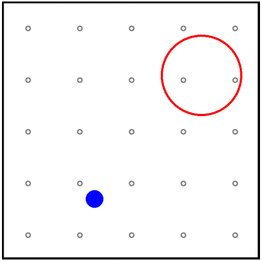


Figure: $\widehat{\alpha}$ and $\widehat{\beta}$ errors for *conf* and *no-conf* cases varying the number of episodes. 25 runs 95% c.i.

Continuous Grid World

Description

- Two-dimensional continuous world
- Two-dimensional continuous actions
- Features are Radial Basis Functions representing the distances of the agent and the goal from a set of fixed points
- Gaussian initial state distribution
    - initial agent position
    - goal position
    - configurable



Continuous Grid World

Error probability

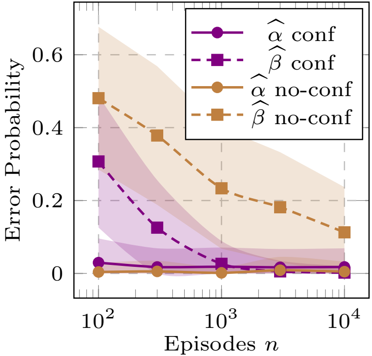


Figure: $\widehat{\alpha}$ and $\widehat{\beta}$ errors for *conf* and *no-conf* cases varying the number of episodes. 25 runs 95% c.i.

Continuous Grid World

Environment configuration

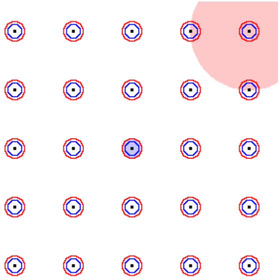


Figure: Initial model

Continuous Grid World

Environment configuration

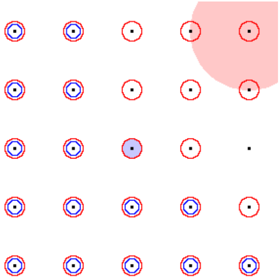


Figure: Identification

Continuous Grid World

Environment configuration

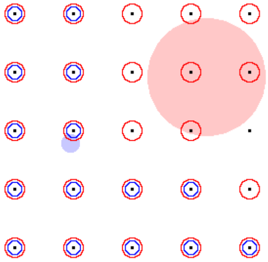


Figure: Configuration

Continuous Grid World

Environment configuration

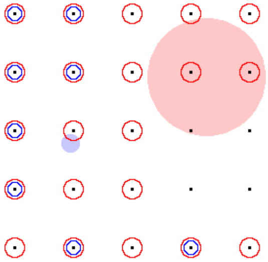


Figure: Identification

Continuous Grid World

Environment configuration

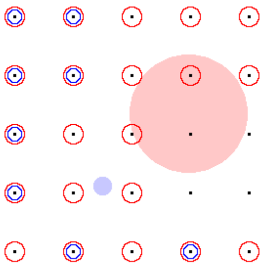


Figure: Configuration

Continuous Grid World

Environment configuration

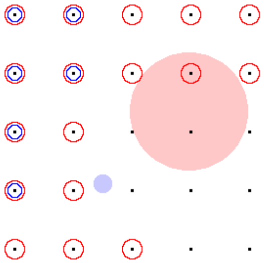


Figure: Identification

Continuous Grid World

Environment configuration

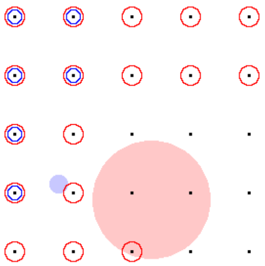


Figure: Configuration

Continuous Grid World

Environment configuration

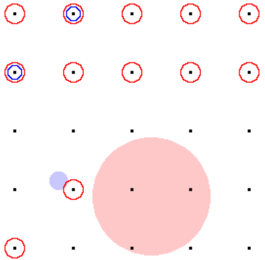


Figure: Identification

Continuous Grid World

Environment configuration

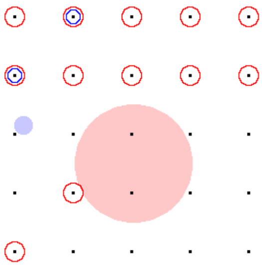


Figure: Configuration

Continuous Grid World

Environment configuration

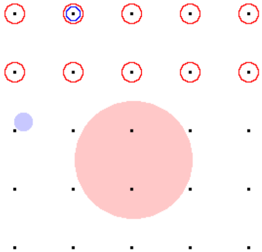


Figure: Identification

Continuous Grid World

Environment configuration

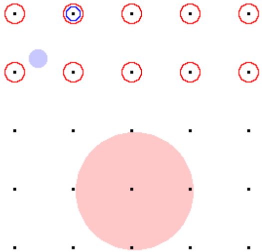


Figure: Configuration

Continuous Grid World

Environment configuration

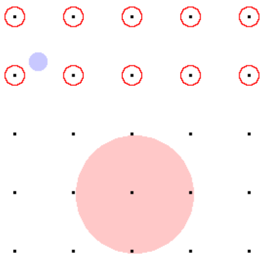


Figure: Identification

Continuous Grid World

Environment configuration

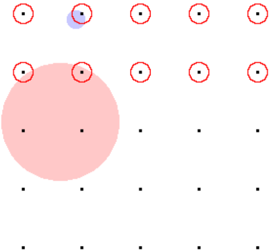


Figure: Configuration

Continuous Grid World

Environment configuration

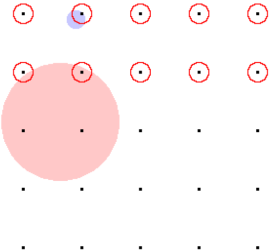


Figure: Identification

Continuous Grid World

Environment configuration

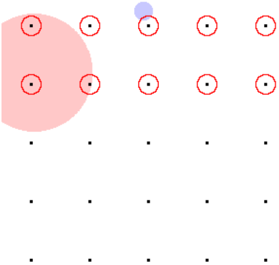


Figure: Configuration

Continuous Grid World

Environment configuration

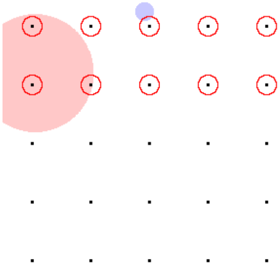


Figure: Identification

Minigolf
Description

- Reaching the hole in the minimum number of steps
- Surpassing the goal gives a penalty
- Distance and friction features
- Action is the force of the stroke
- Length of the "putter"
    - configurable

Minigolf
Choice of the environment

Two agents:

- $\mathcal{A}_1$ perceives distance and friction
- $\mathcal{A}_2$ perceives only distance

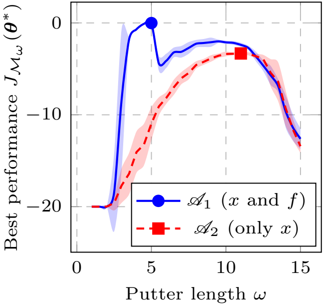
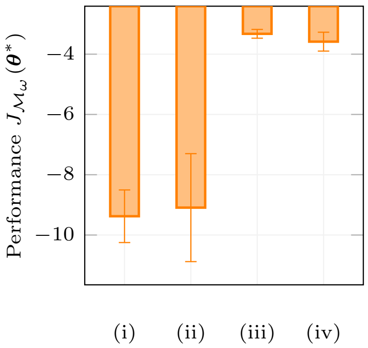


Figure: Performance of the optimal policy varying the putter length ω for agents $\mathcal{A}_1$ and $\mathcal{A}_2$.

Minigolf
Performance comparison

Performance of $\mathcal{A}_2$ with different strategies to select ω:



(i) random
(ii) wrong policy space
(iii) oracle
(iv) identified policy space

Simulated Car Driving
Description

- Reach the end of the road
- State: speed, sensors
- Two-dimensional action: acceleration, steering angle
- Neural network policy (no identifiability assumption)

Simulated Car Driving
Identification Rules with no Identifiability

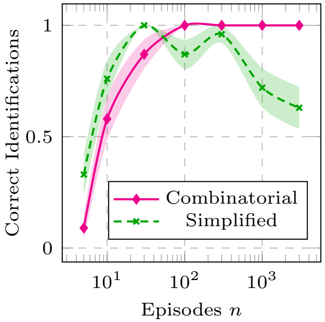


Figure: Fraction of correct identifications varying the number of episodes. 100 runs 95% c.i.

Outline

- Introduction
- Policy Space Identification
- Exploiting environment configuration
- Experimental evaluation
- Applications

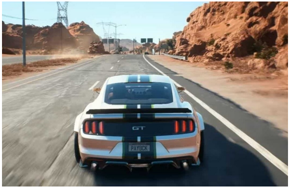
Applications
Behavioral Cloning

Imitate the behavior of an expert by recovering its policy [Argall et al., 2009]

- can be cast to a supervised learning problem
- the policy space gives a suitable hypothesis space to use
- avoid underfitting/overfitting

E.g.,

- learning to drive by observing a pilot
- learning to walk by imitating humans



Applications

Choosing a suitable task for the agent

Each agent may have a **different learning capacity**

- choose a suitable task to solve [Metelli et al., 2018]
- choose an appropriate difficulty

E.g.,

- select road type or vehicle properties in a car driving scenario
- **change the difficulty** of a game according to the player's abilities

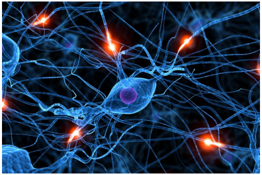
Applications
Neuroscience

The controllable parameters are associated to the **observable state features**

- understanding the perceived state features of an agent

E.g.,

- studying the **perceptions** of living organisms



Contributions

- Two procedures for the identification of the policy space
  - Combinatorial: exponential complexity
  - Simplified: identifiability assumption
- Extension based on Conf-MDP
- Theoretical analysis of the simplified identification rule
  - Bounds on first and second type error probabilities
- Paper submitted to AAAI 2020

Future works

- Theoretical analysis of the combinatorial identification rule
- Improving the complexity of the combinatorial rule using mathematical insights
- Applications to Imitation Learning

References I

Brenna D. Argall, Sonia Chernova, Manuela M. Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2): 1–142, 2013.

Erich L Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

Alberto Maria Metelli, Mirco Mutti, and Marcello Restelli. Configurable markov decision processes. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3488–3497. PMLR, 2018. URL `http://proceedings.mlr.press/v80/metelli18a.html`.

Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

References II

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 2011.

Markov Decision Process

Definition

$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mu \rangle$

- $\mathcal{S}$: set of states
- $\mathcal{A}$: set of actions
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$: Markovian transition model
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$: reward function
- $\gamma \in [0, 1]$: discount factor
- $\mu \in \Delta(\mathcal{S})$: initial state distribution

Policy search
Policy Gradient methods

Policy Gradient methods use the following update rule:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$$

The quantity $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$ can be estimated by trajectories using $\pi_{\boldsymbol{\theta}}$:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \int_{\tau} \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\tau) R(\tau) \mathrm{d}\tau$$

Generalized Likelihood Ratio test

- We consider a parametric model having density function $p_{\boldsymbol{\theta}}$ with $\boldsymbol{\theta} \in \Theta$.
- Let $\Theta_0 \subset \Theta$ a subset of parameters (e.g., Θ_0 may have some parameters set to zero).
- $\boldsymbol{\theta}^*$ is the true parameter

Generalized Likelihood Ratio test

We want to understand whether $\boldsymbol{\theta}^* \in \Theta_0$ or not, i.e.,

$$\mathcal{H}_0 : \boldsymbol{\theta}^* \in \Theta_0$$
$$\mathcal{H}_1 : \boldsymbol{\theta}^* \in \Theta \setminus \Theta_0$$

The GLR statistic [Lehmann and Romano, 2006] is defined as:

$$\lambda(\mathcal{D}) = -2 \log \frac{\sup_{\boldsymbol{\theta} \in \Theta_0} \{\widehat{\mathcal{L}}(\mathcal{D}; \boldsymbol{\theta})\}}{\sup_{\boldsymbol{\theta} \in \Theta} \{\widehat{\mathcal{L}}(\mathcal{D}; \boldsymbol{\theta})\}},$$

where $\widehat{\mathcal{L}}(\mathcal{D}; \boldsymbol{\theta})$ is the likelihood function. Wilk's theorem states that $\lambda(\mathcal{D})$ under $\mathcal{H}_0$ is asymptotically distributed like a χ^2 distribution, which can be used to perform hypothesis testing.

Policy Space Identification in Configurable Environment

Objective

Use Conf-MDP to select a configuration in which the parameters to examine have an **optimal value different from zero**.

Let $I \subseteq \{1, ..., d\}$ be a set of parameter indices we want to test. Intuitively: find the model that maximizes the corresponding components of the gradient, i.e.,

$$\boldsymbol{\omega}^* \in \arg\max_{\boldsymbol{\omega} \in \Omega} \|\nabla_{\boldsymbol{\theta}} J_{\mathcal{M}_{\boldsymbol{\omega}}}(\boldsymbol{\theta}^*(\boldsymbol{\omega}_0))|_I\|^2,$$

where $\boldsymbol{\omega}_0$ is the initial model.