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:

\[ \pi : S \rightarrow \Delta(A) \]

Performance measure:

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

Policy functions

Parametric policies [Sutton and Barto, 2011]:

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

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

where \( \Theta \) is the space of the parameters.
Solution of an MDP:

- Find a policy that maximizes the performance

\[ \theta^* \in \arg \max_{\theta \in \Theta} J_\theta \]

- Search inside the policy space \( \Pi_\Theta \)
- Gradient based approach [Deisenroth et al., 2013]
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 $\theta \in \Theta$
  - 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 $I \subseteq \{1, \ldots, d\}$
- Let $\Theta_I = \{\theta \in \Theta : \theta_i = 0, \forall i \in \{1, \ldots, d\} \setminus I\}$, i.e., $I$ is the set of indexes that can be changed by the agent if the parameter space were $\Theta_I$.
- Let $\pi^* \in \Pi_\Theta$

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

\[
\pi^* \in \Pi_{\Theta_{I^*}} \quad (1)
\]

\[
\forall i \in I^*: \pi^* \notin \Pi_{\Theta_{I^* \setminus \{i\}}} \quad (2)
\]
Combinatorial Identification Rule

Test all the possible subsets of parameters:

\[ I \subseteq \{1, ..., d\} \]

For each \( I \) we consider the pair of hypotheses:

\[ H_{0,I} : \pi^* \in \Pi_{\Theta_I} \]
\[ H_{1,I} : \pi^* \in \Pi_{\Theta \setminus \Theta_I} \]

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

\[ \lambda_I = -2 \log \frac{\sup_{\theta \in \Theta_I} \hat{L}(\theta)}{\sup_{\theta \in \Theta} \hat{L}(\theta)} \]
Combinatorial Identification Rule

A correct subset must satisfy:

\[ \lambda_I \leq c(|I|) \]  \hspace{1cm} (1)

\[ \forall i \in I : \lambda_{I \setminus \{i\}} > c(|I \setminus \{i\}|) \]  \hspace{1cm} (2)

where \( c(l) \) are the critical values.
Combinatorial Identification Rule

A correct subset must satisfy:

\[ \lambda_I \leq c(|I|) \]  \hspace{1cm} (1)
\[ \forall i \in I : \lambda_{I \setminus \{i\}} > c(|I \setminus \{i\}|) \]  \hspace{1cm} (2)

where \( c(l) \) are the critical values.

Drawback: \textit{exponential complexity} \( \mathcal{O}(2^d) \)
The policy space is identifiable if, for all $\theta, \theta' \in \Theta$, we have:

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

Under this assumption, there exists a unique set of parameters that is correct w.r.t. $\pi^*$. 
Simplified Identification Rule

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

For all $i \in \{1, \ldots, d\}$ 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:

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

where $\Theta_i = \{\theta \in \Theta : \theta_i = 0\}$. 
Simplified Identification Rule

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

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

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
Main limitation of previous approach:

- distinguish when a parameter is **not controllable** or just useless for the current task
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 $\omega$ specifying:
  - transition model $P_\omega$
  - initial state distribution $\mu_\omega$
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 $\omega$ specifying:
  - transition model $P_\omega$
  - initial state distribution $\mu_\omega$
- 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 $\hat{\mathcal{I}}_0$
- After each identification update the estimated policy space:
  $$\hat{\mathcal{I}} \leftarrow \hat{\mathcal{I}} \cup \hat{\mathcal{I}}_k$$
- For each parameter $i \in \{1, \ldots, d\} : i \notin \hat{\mathcal{I}}$
  - Find a new model $\omega_k$
  - Collect data $D_k$ observing $\pi^*(\omega_k)$
  - Perform an identification obtaining $\hat{\mathcal{I}}_k$ and update $\hat{\mathcal{I}}$
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: $\hat{\alpha}$ and $\hat{\beta}$ errors for \textit{conf} and \textit{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: \( \hat{\alpha} \) and \( \hat{\beta} \) errors for \textit{conf} and \textit{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:

- $A_1$ perceives distance and friction
- $A_2$ perceives only distance

![Figure: Performance of the optimal policy varying the putter length $\omega$ for agents $A_1$ and $A_2$.](image)

**Figure:** Performance of the optimal policy varying the putter length $\omega$ for agents $A_1$ and $A_2$. 
Minigolf
Performance comparison

Performance of $A_2$ with different strategies to select $\omega$:

- (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
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


Markov Decision Process
Definition

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

Policy Gradient methods use the following update rule:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J_\theta$$

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

$$\nabla_\theta J_\theta = \int_\tau \nabla_\theta p_\theta(\tau) R(\tau) d\tau$$
Generalized Likelihood Ratio test

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

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

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

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

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

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

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

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

$$\omega^* \in \arg \max_{\omega \in \Omega} \| \nabla_\theta J_M(\theta^*(\omega_0)) |_I \|^2,$$

where $\omega_0$ is the initial model.