Similarity-based transfer learning of decision policies

Eliška Zugarová
Department of Adaptive Systems
Institute of Information Theory and Automation
The Czech Academy of Sciences
Prague, Czech Republic
eliska.zugarova@gmail.com

Tatiana V. Guy, *Senior Member, IEEE*
Department of Adaptive Systems
Institute of Information Theory and Automation
The Czech Academy of Sciences
Prague, Czech Republic
guy@utia.cas.cz

***Abstract*—We consider a problem of learning decision policy from past experience available. Using the Fully Probabilistic Design (FPD) formalism, we propose a new general approach for finding a stochastic policy from the past data. The proposed approach assigns degree of similarity to all of the past closed-loop behaviors. The degree of similarity expresses how close the current decision making task is to a past task. Then it is used by Bayesian estimation to learn an approximate optimal policy, which comprises the best past experience. The approach learns decision policy directly from the data without interacting with any supervisor/expert or using any reinforcement signal. The past experience may consider a decision objective different than the current one. Moreover the past decision policy need not to be optimal with respect to the past objective. We demonstrate our approach on simulated examples and show that the learned policy achieves better performance than optimal FPD policy whenever a mismodeling is present.**

***Index Terms*—probabilistic model, transfer learning, closed-loop behavior, fully probabilistic design, Bayesian estimation, sequential decision making**

I. Introduction

Learning from the past experience that mimics the expert's/teacher's decisions or behaviour (for instance imitation learning [1], apprenticeship learning, [2]) has became popular in the recent years.

Successful applications, like natural language processing [3], autonomous micro aerial vehicles [4], mimicking human body language in games [5], support the interest in developing these approaches. Often expert's behavior is very complex (for instance car driving) and can hardly be represented via a set of feasible algorithms. On the other hand demonstrating a desired behavior may be easy for the expert but designing algorithms imitating this behaviour is mostly difficult. Even when designed, these algorithms have a limited ability to generalise past experience and to find an optimal decision policy under new circumstances (for instance change of the system or new decision making preferences). Another problem is that to reach a high quality decision policy the existing algorithms may require a large amount of demonstration/expert data collected in long-horizon decision making [6].

Partial support of the project MŠMT LTC18075 is acknowledged.

Many successful approaches rely on querying the expert, thus becoming dependent on an expert's feedback which is often not feasible or restricted by application domain. Generally learning targeted, sequential decision-making behavior is quite difficult as the resulting algorithm must often reason about the long-term consequences of the currently chosen actions.

In the paper we propose a new approach for finding a stochastic policy using past data (either generated by expert or not).

The proposed approach uses the Fully Probabilistic Design (FPD) formalism [7], [8]. We compute value of similarity that reflects how much each past closed-loop behavior matches to the targeted closed-loop behavior. Then, using these similarity values the optimal decision policy is learned from all of the past data. The resulting decision policy thus comprises the best experience obtained in the past. The approach learns decision policy directly from the data without interacting with a supervisor/expert or using any reinforcement signal. The past experience may consider a decision objective different than the current one. Moreover the past decision policy need not to be optimal with respect to the past objective. We demonstrate our approach on simulated examples and show that the learned policy achieves better performance than optimal FPD policy whenever a mismodeling is present.

The paper outline is as follows. The next section introduces notations and notions, recalls necessary information about the Markov Decision Processes and the Fully Probabilistic Design. Section III formulates and solves similarity-based transfer learning. Section IV describes the algorithm and experiments performed; summarises and discusses the obtained results. Section V provides concluding remarks.

II. Preliminaries

A. General Notation

- $\mathbb{N}$ and $\mathbb{R}$ stand for sets of natural and real numbers, respectively.
- Sets of values of discrete random variables are denoted by bold capital letters, i.e. $\mathbf{X}$ is a set of values x.
- $|\mathbf{X}|$ represents the cardinality of set $\mathbf{X}$.
- Value of variable x at discrete time $t \in \mathbb{N}$ is denoted as x_t.

- $p(x)$ denotes probability mass function of discrete random variable x onwards referred to as probability function.
- $p(x|y)$ is the conditional probability of a discrete random variable x conditioned on random variable y.
- $\mathrm{E}[x]$ is expectation of random variable x and $\mathrm{E}[x|y]$ denotes conditional expectation of random variable x conditioned on a random variable y.

B. Markov Decision Process

Markov Decision Process (MDP) [9] is a framework widely used for sequential decision-making problems. It serves to model an agent that interacts with a system by deliberately choosing actions to achieve its objectives expressed via a reward function.

Definition 1 (MDP): A finite-horizon discrete-time fully observable Markov Decision Process is given by $\{\mathbf{T}, \mathbf{S}, \mathbf{A}, p, r\}$, where

$\mathbf{T} = \{1, 2, ..., N\}, N \in \mathbb{N}$, is a set of decision epochs,
$\mathbf{S}$ is a discrete finite set of all achievable system states,
$\mathbf{A}$ is a discrete finite set of all possible actions of the agent,
$p : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to [0, 1]$ is a transition probability function that models the evolution of the system, $p(s_t|a_t, s_{t-1})$ is the probability that the system moves from state $s_{t-1} \in \mathbf{S}$ to state $s_t \in \mathbf{S}$ after action $a_t \in \mathbf{A}$ is taken, $t \in \mathbf{T}$,
$r : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to \mathbb{R}$ is a reward function; $r(s_t, a_t, s_{t-1})$ is the immediate reward the agent receives after taking action $a_t \in \mathbf{A}$ in state $s_{t-1} \in \mathbf{S}$ and prompting the system to move to state $s_t \in \mathbf{S}$, $t \in \mathbf{T}$.

The system transition is ruled by the Markov property [9], which means that it depends only on the last system state and the chosen action.

The agent selects actions to maximise overall possible reward. The optimal behavior is determined by choosing an action maximizing the total expected reward at each decision epoch. The action selection is ruled by a *decision policy*, which is a sequence of *decision rules* and can be expressed as

$$\left\{p(a_t|s_{t-1})|a_t \in \mathbf{A}, s_{t-1} \in \mathbf{S}\right\}_{t=1}^{N}.$$

Each decision rule is a conditional probability function over the action set.

C. Fully Probabilistic Design

The Fully Probabilistic Design (FPD) framework [7], [10], [11] models sequential decision-making problems and allows for a more general definition of the agent's reward, which enables to express the agent's preferences more effectively.

Using the MDP notation, the behavior of agent-system pair can be modelled as follows.

Definition 2 (Closed-loop model): The behavior of the closed-loop formed of the agent-system pair up to time $t \in \mathbb{N}$ is described by a joint probability function $p(s_t, a_t, s_{t-1}, \ldots, s_1, a_1, s_0)$, where $s_\tau \in \mathbf{S}$, $0 \le \tau \le t$, are states of the system and $a_\tau \in \mathbf{A}$, $1 \le \tau \le t$ denote actions of the agent.

By applying the Markov property [9] and the chain rule for probabilities, the closed-loop model can be written in the form

$$\begin{aligned} p(s_t, &a_t, s_{t-1}, \ldots, s_1, a_1, s_0) \\ &= \prod_{\tau=1}^{t} p(s_\tau|a_\tau, s_{\tau-1}) p(a_\tau|s_{\tau-1}) p(s_0), \end{aligned} \tag{1}$$

where $p(s_\tau|a_\tau, s_{\tau-1})$ is *the transition model*, $p(a_\tau|s_{\tau-1})$ is *the decision rule* at decision epoch τ, and $p(s_0)$ represents the prior distribution of the initial state.

Instead of defining a reward function like in the MDP problem formulation, the agent's preferences over possible states and actions are expressed via an ideal closed-loop model.

Definition 3 (Ideal closed-loop model[1]): A targeted behavior of the agent-system loop up to time $t \in \mathbb{N}$ is described by a joint probability function ${}^{I}p(s_t, a_t, s_{t-1}, \ldots, s_1, a_1, s_0)$, $s_\tau \in \mathbf{S}$ for $0 \le \tau \le t$ and $a_\tau \in \mathbf{A}$ for $1 \le \tau \le t$.

When factorising the ideal closed-loop model in a way similar to (1), the first factor describes targeted dynamics of the system and the second factor reflects possible preferences among possible actions. Definition 2 and Definition 3 allow formulating the underling DM problem via minimization of the Kullback-Leibler divergence between the actual closed-loop model (Definition 1) and the desired closed-loop model (Definition 3). In other words, the optimal decision policy should make the closed-loop behavior as close as possible to the targeted one. This is an essence of FPD [10] and the FPD optimal policy can be defined as follows.

Definition 4 (Optimal FPD decision policy): An optimal FPD decision policy is defined as

$$\begin{aligned} \pi_{FPD}^{opt} = \underset{\left\{p(a_t|s_{t-1})\right\}_{t=1}^{H}}{\arg\min} \boldsymbol{D}\Big(&p(s_H, a_H, \ldots, s_1, a_1, s_0) \\ &\big|\big|{}^{I}p(s_H, a_H, \ldots, s_1, a_1, s_0)\Big), \end{aligned}$$

where $p(a_t|s_{t-1})$ is a DM rule at time t, $H \in \mathbb{N}$ is optimization horizon, $s_\tau \in \mathbf{S}$, for $0 \le \tau \le H$, $a_\tau \in \mathbf{A}$, for $1 \le \tau \le H$, and $\boldsymbol{D}(\cdot||\cdot)$ is the Kullback-Leibler divergence.
The following theorem solves the FPD problem.

Theorem 1 (Solution to FPD): The explicit optimal FPD policy minimizing the KL divergence (see Definition 4) is constructed using the following equations

$$\begin{aligned} {}^{opt}p(a_t|s_{t-1}) &= {}^{I}p(a_t|s_{t-1}) \\ &\quad\times \frac{\exp\big(-\alpha(a_t, s_{t-1}) - \beta(a_t, s_{t-1})\big)}{\gamma(s_{t-1})} \\ \alpha(a_t, s_{t-1}) &= \sum_{s_t \in \mathbf{S}} p(s_t|a_t, s_{t-1}) \ln \frac{p(s_t|a_t, s_{t-1})}{{}^{I}p(s_t|a_t, s_{t-1})} \\ \beta(a_t, s_{t-1}) &= -\sum_{s_t \in \mathbf{S}} \ln(\gamma(s_t)) p(s_t|a_t, s_{t-1}) \end{aligned}$$

[1]for brevity the term *ideal model* is sometimes used.

$$\gamma(s_{t-1}) = \sum_{a_t \in \mathbf{A}} {}^I p(a_t|s_{t-1}) \times \exp\big(-\alpha(a_t, s_{t-1}) - \beta(a_t, s_{t-1})\big)$$
$$\gamma(s_H) = 1$$

for all $t \in \{1, \ldots, H\}$, where $H \in \mathbb{N}$ is a horizon of optimization.

Proof: See [10]. ■

Relation between FPD and MDP: FPD formulation is more general and an MDP problem can be formulated and solved as FPD problem with

$$r(s_t, a_t, s_{t-1}) = -\ln \frac{p(s_t, a_t|s_{t-1})}{{}^I p(s_t, a_t|s_{t-1})}. \quad (2)$$

The relation (2) results from a direct application of the Kullback-Leibler divergence and Definition 4.

III. Similarity-Based Transfer Learning

In this section we present an approach to learning a DM policy from past data. The approach is based on the probabilistic modeling used within FPD methodology and the newly introduced similarity.

The agent interacts with the system and aims to find an optimal DM policy that ensures reaching the targeted DM preferences. Let us also suppose that there are data describing the *past* closed-loop behaviour formed of the same system and an agent[2]. Such data may be obtained from the experts (for instance demonstration or training data) or describe solution of other DM tasks solved on the same system. The past DM policies are assumed to be consistent with some unknown ideal model, though not necessary optimal with respect to this ideal. The past ideal closed-loop models can significantly differ from the current one thus the past data should not match the current DM objective. We are interested in learning the optimal DM policy from these data, i.e. in transferring the best experience gained on the system to the present DM task.

A. Solution Concept and Similarity Notion

Let the agent sequentially interact with the system. The agent's DM preferences are expressed via ideal model ${}^I p$ (see Definition 3). We need to find an optimal sequence of DM rules ensuring reaching this ideal. Consider past data $\{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$ describing the previous, already completed, DM task. Actions $a_\tau, \tau = 1, ..., t-1$ were selected while reaching past (and unknown) ideal model ${}^I \tilde{p}$ which can generally be different from the current ideal, ${}^I p$. Note that actions $a_\tau, \tau = 1, ..., t-1$ do not even need to be optimal with respect to the past ideal.

The proposed approach learns a sequence of DM rules for the current DM task from the past data available. The key idea uses the fact that behavior of any system is substantially determined by fixed dependencies (for instance given by the first principles) that are independent of states and actions. Besides, data communicate indirect information about decision patterns[3] applied in the past. Once the experience was collected on the same system, we can learn DM rules that suit the current DM objectives. To distinguish relevant experience we should be able to measure "degree of matching" of the past data to the current DM objectives. To evaluate that, we introduce the term of similarity quantifying the extent to which the past behavior fits the current DM aim.

Definition 5 (Similarity): Let $\{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$ be a set of observations of a completed decision-making task. We define the *similarity* between the current decision problem with the ideal model ${}^I p$ and a past problem from decision epoch τ as

$$\sigma_\tau = {}^I p(s_\tau, a_\tau|s_{\tau-1}) \in [0, 1], \quad (3)$$

where $(s_\tau, a_\tau, s_{\tau-1})$ is an observation of one decision and state transition, and $0 < \tau < t$.

The definition of similarity, has a clear and intuitive meaning. Whenever past data $(s_\tau, a_\tau, s_{\tau-1})$ bring high values of the current ideal model ${}^I p$[4], see Definition 3, past transition $(s_{\tau-1}, a_\tau) \rightarrow s_\tau$, is close to the targeted behavior in the current DM problem. The value of the similarity is small whenever a past decision pattern: i) simulates state transition that does not fully match the current DM preferences (expressed by ideal model ${}^I p$), ii) is considered disadvantageous for the current DM preferences. If the past system transition is desirable with regard to the current DM problem, the similarity is high.

The following definition of similarity is almost identical to Definition 5, except the values are normalized.

Definition 6 (Normalized similarity): We will consider the set $\{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$ be a set of observations of a completed decision-making task. The *normalized similarity* between the current decision problem with the ideal model ${}^I p$ and a past problem from decision epoch τ, $0 < \tau < t$, is defined as

$$\sigma_\tau = \frac{{}^I p(s_\tau, a_\tau|s_{\tau-1})}{\sigma_{max}} \in [0, 1], \text{ where}$$
$$\sigma_{max} = \max_{s_t, s_{t-1} \in \mathbf{S}, a_t \in \mathbf{A}} {}^I p(s_t, a_t|s_{t-1}). \quad (4)$$

Introducing a normalized version of the similarity is important because similarity equals to an ideal likelihood of past data, so it is necessary to adjust obtained values to a common scale. Therefore to judge whether the obtained value of similarity is high enough we need to normalise them.

Note that Definition 5 and Definition 6 can be used in case the past data is the only information available, i.e. no information about the past DM preferences is available. It is clear that if past ideal models are known, the similarity can be measured via any divergence measure on the space of probability distributions.

B. Bayesian Estimation of the Decision Policy

This section introduces Bayesian approach [12] that guides the optimal DM rule selection based on the past data. Through-

[2]generally past data can be generated by a different agent(s) than the current one

[3]i.e. dependence "system state→corresponding action→next state"

[4]or by other words the likelihood is high

out this section, $(s', a) \to s$ denotes an arbitrary system state transition $(s_{\tau-1}, a_\tau) \to s_\tau$, where $\tau \in \mathbf{T}$.

Consider a DM task characterized by ideal model ${}^I p$ and past data collected on the same system though for a different DM task. The data consists of a sequence of state transitions $d_{t-1} = \{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$ and our goal is to infer the targeted DM rule using d_{t-1}.

Generally the unknown closed-loop model $p(s_t, a_t|s_{t-1})$, which implicitly contains DM rule $p(a_t|s_{t-1})$ at time t, can be parameterized as $p(s_t, a_t|s_{t-1}, \theta)$, where $\theta \in \mathbf{\Theta}$ in an unknown finite-dimensional parameter and $\mathbf{\Theta}$ is a continuous parameter space. We define the parameter space as

$$\mathbf{\Theta} = \Bigg\{ \theta_{s,a|s'} \Big| s, s' \in \mathbf{S}, a \in \mathbf{S}, \theta_{s,a|s'} \in [0,1], \sum_{s\in\mathbf{S}, a\in\mathbf{A}} \theta_{s,a|s'} = 1, \forall s' \in \mathbf{S} \Bigg\},$$

and the parametrization as $\theta_{s_t,a_t|s_{t-1}} = p(s_t, a_t|s_{t-1}, \theta)$.

The closed-loop behavior based on the observed data at time t is then described using marginalization and the chain rule as

$$\hat{p}(s_t, a_t|d_{t-1}) = \int_\Theta p(s_t, a_t|d_{t-1}, \theta) p(\theta|d_{t-1}) d\theta. \quad (5)$$

The second factor $p(\theta|d_{t-1})$ in (5) is a distribution expressing our beliefs about the unknown parameter based on d_{t-1}.

Using the available data d_{t-1} we can write the posterior distribution of the parameter at decision epoch $t \in \mathbb{N}$ via the weighted Bayes' rule [13]:

$$p(\theta|d_{t-1}) = \frac{\prod_{\tau=1}^{t-1} p(s_\tau, a_\tau|s_{\tau-1}, \theta)^{\omega_\tau} p(\theta|s_0)}{\int_\Theta \prod_{\tau=1}^{t-1} p(s_\tau, a_\tau|s_{\tau-1}, \theta)^{\omega_\tau} p(\theta|s_0) d\theta}. \quad (6)$$

In (6), $\omega_\tau = \sigma_\tau$ are values of the similarity (4) that numerically express how data d_{t-1} fit the current ideal model ${}^I p$. New system transition, $(s_{t-1}, a_t) \to s_t$, enriches data with the tuple (s_t, a_t, s_{t-1}), and the posterior distribution can be updated to $p(\theta|d_t)$ via (6). We simplified the formula (6) using the Markov property (1) stating that the system state transition depends on the last state and action only.

It is assumed that initial state s_0 does not change the prior beliefs about parameters of the closed-loop model, i.e. $p(\theta|s_0) = p(\theta)$. This assumption is justified by considering the initial state as an initial condition not dependent on the parameter [12].

Additionally, we assume that $\theta_{s_t,a_t|s_{t-1}} = p(s_t, a_t|s_{t-1}, \theta)$ follows multinomial distribution and prior $p(\theta)$ is a product of Dirichlet distributions[5]:

$$p(\theta) = \prod_{s'\in\mathbf{S}} \mathrm{Dir}\Big(\theta_{\cdot,\cdot|s'}, \nu_0^{\cdot,\cdot|s'}\Big), \quad (7)$$

where $\theta_{\cdot,\cdot|s'}$ is a vector of parameters, and $\nu_0^{\cdot,\cdot|s'}$ is a vector of values $\nu_0^{s,a|s'} > 0$, $s \in \mathbf{S}$, $a \in \mathbf{A}$. Then the posterior obtained using the weighted Bayes rule (6) has the form

$$p(\theta|d_{t-1}) \propto \prod_{s'\in\mathbf{S}} \mathrm{Dir}\Big(\theta_{\cdot,\cdot|s'}, V_{t-1}^{\cdot,\cdot|s'}\Big) \quad (8)$$

with concentration parameters defined recursively $\forall 1 \le \tau \le t-1$

$$V_\tau^{s,a|s'} = \omega_\tau \delta(s, s_\tau)\delta(a, a_\tau)\delta(s', s_{\tau-1}) + V_{\tau-1}^{s,a|s'},$$
$$V_0^{s,a|s'} = \nu_0^{s,a|s'}.$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta function and $(s_\tau, a_\tau, s_{\tau-1}) \in d_{t-1}$, $1 \le \tau \le t-1$, are *observed* realizations of states and actions. These realizations describe the possible closed-loop transitions. These learned parameters essentially represent the number of transitions $(s', a) \to s$, that were observed in the past, weighted by "usefulness" of a particular transition for the current DM problem.

The deduced form of the posterior distribution (8) is then used to derive the learned optimal DM rule

$$\hat{p}(a_t|s_{t-1}) = \sum_{s_t\in\mathbf{S}} \int_\Theta p(s_t, a_t|s_{t-1}, \theta) p(\theta|d_{t-1}) d\theta. \quad (9)$$

After the computing using the definition of the Beta function that appears as a normalizing constant in the Dirichlet distribution, and utilizing properties of the Gamma function, (9) can be rewritten as

$$ {}^{opt}\hat{p}(a_t|s_{t-1}) = \frac{\sum_{\tau=1}^{t-1} \omega_\tau \delta(a_t, a_\tau)\delta(s_{t-1}, s_{\tau-1}) + \sum_{s\in\mathbf{S}} \nu_0^{s,a_t|s_{t-1}}}{\sum_{\tau=1}^{t-1} \omega_\tau \delta(s_{t-1}, s_{\tau-1}) + \sum_{s\in\mathbf{S}} \sum_{a\in\mathbf{A}} \nu_0^{s,a|s_{t-1}}}. \quad (10)$$

The formula (10) gives an optimal DM rule that was learned from the past history. The learned rule comprises the best past experience which can be useful for the current DM objective. Note that this rule, though called optimal, is an approximation of the unknown optimal rule.

C. Exploration

The approach proposed above exploits all available information about the closed-loop behavior and the best experience available. However, past data can be i) incomplete; ii) obtained for DM preferences significantly differing from current objectives (defined by the ideal model ${}^I p$). Then an exploration ability should be added as it helps to gather more information about the system.

A computationally inexpensive exploration strategy is the ϵ-greedy explorative strategy. It was introduced in [15] as a strategy solving the multi-armed bandit problem. It chooses the currently optimal action (i.e. uses the optimal decision rule (10)) with probability 1-ϵ and a random action with probability ϵ, $\epsilon \in [0,1]$.

To prevent unnecessary over-exploration, an ϵ-greedy exploration technique is applied whenever there is a lack of data, which is visible from the mean value of $m \in \mathbf{N}$ last computed similarities. If the mean is lower than a given threshold

[5]Dirichlet distribution as prior is a common choice in Bayesian theory. It simplifies the computation of the posterior distribution because the prior and the posterior distributions are conjugate for multinomial distribution sampling [14]

$q \in [0,1]$, ϵ-greedy exploration is activated. Otherwise, the learned optimal decision rule (10) is applied directly.

The resulting algorithm of finding the optimal decision rule with incorporated exploration and using the normalized version of the similarity (4) is shown in Algorithm 1.

Algorithm 1 Transfer learning of an optimal decision rule with exploration

Input: data $d_{t-1} = \{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$, ideal model ${}^I p$
for $\tau = 1, \dots, t-1$ **do**
  Compute weight $\omega_\tau \equiv \sigma_\tau = \frac{{}^I p(s_\tau, a_\tau | s_{\tau-1})}{\sigma_{max}}$;
end for
while $t \leq N$ **do**
  Learn the optimal decision rule ${}^{opt}\hat{p}(a_t|s_{t-1})$ (10);
  if $\frac{1}{m}\sum_{\tau=t-m}^{t-1} \omega_\tau < q$ **then**
    Generate ξ_t from a uniform distribution on the interval [0,1];
    if $\xi_t < \epsilon$ **then**
      Use the uniform decision rule $p(a_t|s_{t-1}) = \frac{1}{|\mathbf{A}|}$;
    else
      Use the learned decision rule (10);
    end if
  else
    Use the learned decision rule (10);
  end if
  Observe new state transition $(s_{t-1}, a_t) \to s_t$;
  Calculate new weight $\omega_t \equiv \sigma_t = \frac{{}^I p(s_t, a_t | s_{t-1})}{\sigma_{max}}$;
  $t = t + 1$;
end while

IV. Illustrative experiments

The proposed approach was demonstrated and verified through a series of simulated experiments. Each experiment was repeated 100 times.

General setting. We consider a discrete system with state space $\mathbf{S} = \{s^1, s^2, s^3\}$. The action space contains four actions, $\mathbf{A} = \{a^1, a^2, a^3, a^4\}$. The particular coefficients of the transition model, $p(s_t|a_t, s_{t-1})$, were generated randomly, so the system dynamics, was different each time. Initial state s_0 was also chosen randomly with respect to the uniform distribution. The overall experiment tasks were as follows:

- generate data for different DM tasks (determined by different DM preferences), i.e. imitate past experience
- set a new DM task characterised by the new DM preferences
- learn the optimal DM policy for the new DM task by using the proposed transfer learning
- apply the DM policy learned and compare the obtained close-loop performance.

How were the past data generated? To simplify further comparison and verification of the proposed approach, the FPD settings were used for past data generating. The following experiment was performed. The past DM objectives were set and expressed via ideal model (see Definition 3). Then optimal FPD decision policy (Theorem 1) was computed and applied to the system. The optimal policy was computed for the completely known transition model (no mismodeling). The horizon of the policy optimization was set to $H = 10$ decision epochs. The resulting closed-loop behavior was observed over $k = 60$ decision epochs, so the past data available were $d_{60} = \{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{60}$. These data were further used for transfer learning.

What were DM preferences of the past DM tasks? Three different ideal transition models were used during the generation of the demonstration data d_{60}. For all of them, the ideal decision rule was uniform, i.e. no preference over actions existed. The first ideal model, labeled as ${}^I\tilde{p}_1$, favored state s^1 and was defined as

$$\begin{aligned} {}^I\tilde{p}_1(s_t = s^1|a_t, s_{t-1}) &= 0.99998, \\ {}^I\tilde{p}_1(s_t \neq s^1|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{11}$$

for all $a_t \in \mathbf{A}$, $s_{t-1} \in \mathbf{S}$. The second ideal model, ${}^I\tilde{p}_{1,2}$, reflected equal preference for s^1 and s^2. For all $a_t \in \mathbf{A}$ and all $s_{t-1} \in \mathbf{S}$ it was defined as follows

$$\begin{aligned} {}^I\tilde{p}_{1,2}(s_t = s^1|a_t, s_{t-1}) &= 0.499995, \\ {}^I\tilde{p}_{1,2}(s_t = s^2|a_t, s_{t-1}) &= 0.499995, \\ {}^I\tilde{p}_{1,2}(s_t = s^3|a_t, s_{t-1}) &= 0.00001. \end{aligned} \tag{12}$$

The third ideal model ${}^I\tilde{p}_3$ favored state s^3 only:

$$\begin{aligned} {}^I\tilde{p}_3(s_t = s^3|a_t, s_{t-1}) &= 0.99998, \\ {}^I\tilde{p}_3(s_t \neq s^3|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{13}$$

for all $a_t \in \mathbf{A}$ and for all $s_{t-1} \in \mathbf{S}$.

There were no special preferences over actions. Thus the ideal decision rule was a uniform probability function: ${}^I p(a_t|s_{t-1}) = \frac{1}{|\mathbf{A}|} = 0.25$, for all $a_t \in \mathbf{A}$, $s_{t-1} \in \mathbf{S}$. The agent's ideal transition model ${}^I p$ was the same as ${}^I\tilde{p}_1$ (11), focused on reaching state s^1. It was

$$\begin{aligned} {}^I p(s_t = s^1|a_t, s_{t-1}) &= 0.99998, \\ {}^I p(s_t \neq s^1|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{14}$$

for all $a_t \in \mathbf{A}$, $s_{t-1} \in \mathbf{S}$.

How the results were compared? With the past data d_{60} collected, an optimal decision policy with respect to ideal ${}^I p$ for $h = 100$ was searched. Normalized version of the similarity (4) was used to weight the past observations.

To verify the proposed approach, the optimal DM policy for the current DM task was searched via different algorithms (names correspond to the notations used in Fig. 1- Fig. 4):

- Rand - random policy;
- TL - the proposed similarity-based transfer learning an optimal policy without exploration, (Section III)
- $\text{TL}_{\text{explore}}$ - the proposed similarity-based transfer learning an optimal policy with exploration strategy, Section III-C.
- $\text{FPD}_{\text{learn}}$ - FPD method (Section II-C) when the transition model is unknown and learned on-line
- FPD - FPD method (Section II-C) that uses the complete

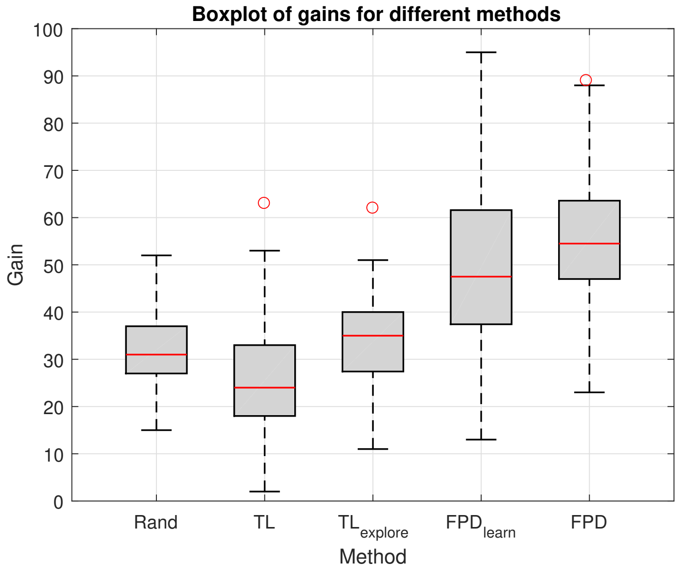

Fig. 1. Boxplot of gains, data gathered with ideal model ${}^{I}\tilde{p}_3$. Rand - random policy, TL - TL method, $TL_{explore}$ - TL method with exploration, FPD_{learn} - learning FPD method, FPD - FPD method with complete knowledge.

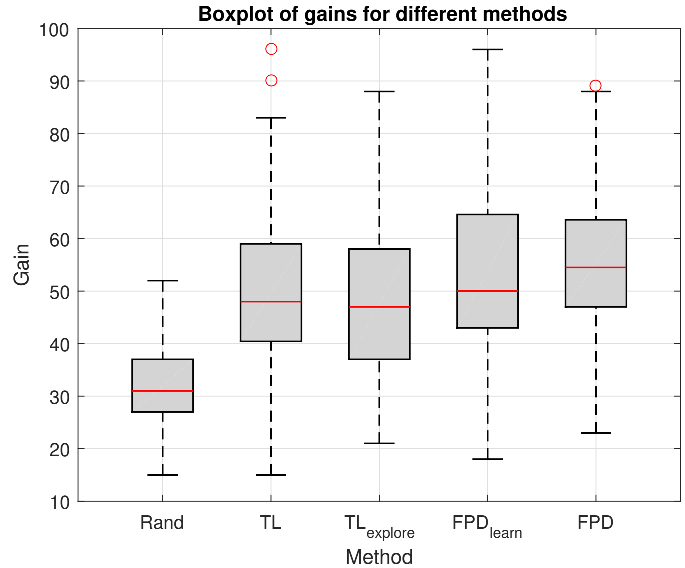

Fig. 2. Boxplot of gains, data gathered with ideal model ${}^{I}\tilde{p}_{1,2}$. Rand - random policy, TL - TL method, $TL_{explore}$ - TL method with exploration, FPD_{learn} - learning FPD method, FPD - FPD method with complete knowledge.

knowledge of the transition model.

The closed-loop behaviors corresponding to the different methods of policy generation were then compared based on the closed-loop performance.

The performance of the TL method was measured by *gain*, which was defined as the overall number of occurrences of state s^1. Prior distribution parameters (7) were chosen so that they were all equal to

$$\nu_0 = \frac{1}{|\mathbf{S}|} \min_{\substack{s_t, s_{t-1} \in \mathbf{S} \\ a_t \in \mathbf{A}}} {}^{I}p(s_t, a_t | s_{t-1}),$$

which suggests no prior information about the parameter of the closed-loop model.

The seed for reproducibility of results was set to 10. The methods and experiments were implemented in Matlab R2016b®. Boxplot figures were generated using Alternative box plot function for Matlab from the IoSR Matlab Toolbox [16].

A. Comparison of the TL and the FPD methods

The TL method was used either without any exploration (10), or with adjusted exploration strategy (see Algorithm 1). Then the exploration rate was set to $\epsilon = 0.3$, the threshold of low average similarity was $q = 0.4$, and the number of previous similarities to be averaged was $m = 10$.

The FPD method (Theorem 1) was employed either with complete knowledge of the transition model $p(s_t|a_t, s_{t-1})$, or without any prior knowledge of the model. In the latter case, Bayesian estimation was applied to learn the transition model using the same set of observations d_{60} as those available for the TL method. The case with complete knowledge of the transition model represents a boundary situation because it is not common in real-life applications and served to comparison only. FPD policy was optimized over a horizon of $H = 10$ decision epochs in both cases.

The two methods were also compared to a *random policy*, that is a policy that chooses actions randomly at each decision epoch and is defined for all $a_t \in \mathbf{A}$ and all $s_{t-1} \in \mathbf{S}$ as

$$p(a_t|s_{t-1}) = \frac{1}{|\mathbf{A}|}. \quad (15)$$

Fig. 1 shows a boxplot representing results of a method comparison where past data d_{60} were collected using ideal transition model ${}^{I}\tilde{p}_3$ (13), so with completely different DM preferences than the current ideal model, ${}^{I}p$ (14). Fig. 2 illustrates results of the experiment for the past ideal model, ${}^{I}\tilde{p}_{1,2}$, (12) that expresses DM preferences that partly overlap with the current ones, expressed via ${}^{I}p$. Fig. 3 represents gains of the compared methods when past data d_{60} were generated with ${}^{I}\tilde{p}_1$ (11), i.e. DM preferences of the past and current tasks coincide.

Fig. 1 illustrates that when there is no overlap of past and present objectives, the TL performs worse even than the random policy. When TL with exploration was used, the gains rose slightly above the random policy gains. However, they were still considerably worse than the FPD ones. As shown in Fig. 2, the results improved greatly when the decison policy had been learned from past data more relevant to the current DM preferences. The performance of the TL is nearly equal to that of the FPD with completely known transition model. Finally, as can be seen in Fig. 3, the TL method outperforms the FPD method in the conditions of data matching current objectives. Note that the exploration strategy worsened the results only slightly when past data were appropriate.

Results of the same experiments as in Fig.1, Fig.2 and Fig.3 are shown in Fig. 4, where gains of the random policy were

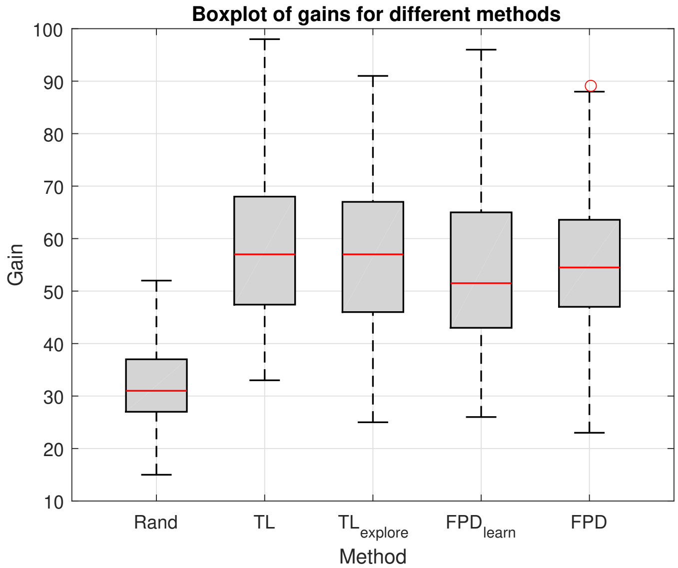

Fig. 3. Boxplot of gains of different methods, data gathered using ideal transition model ${}^I\tilde{p}_1$. Rand - random policy, TL - TL method, $\text{TL}_{\text{explore}}$ - TL method with exploration, $\text{FPD}_{\text{learn}}$ - learning FPD method, FPD - FPD method with complete knowledge.

subtracted from gains of other methods. The transition model parameters were different for each simulation so the difficulty of obtaining the desired states varied. Fig. 4 depicts success of each DM policy compared to the random policy depending on the quality of the past data used. Naturally the results of the FPD method with complete knowledge of the transition model were the same for all three types of data because the method did not need to use the data to estimate the transition model.

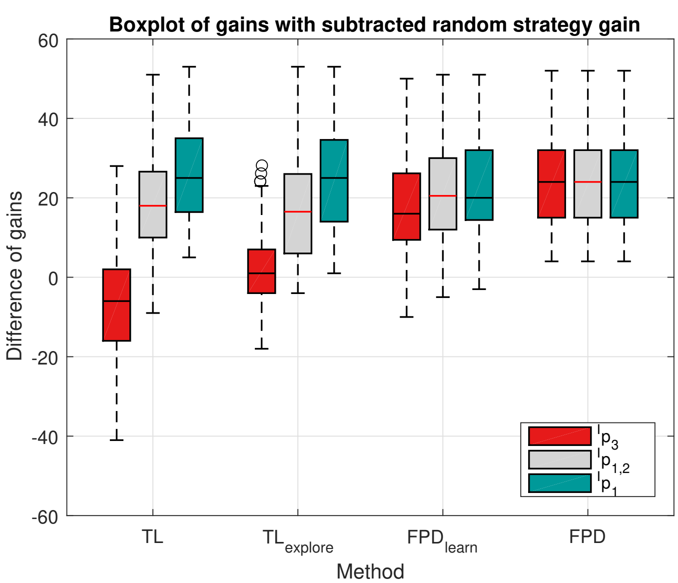

Fig. 4. Boxplot of gains of different methods with gain of random policy subtracted, data gathered with three different ideal models ${}^I\tilde{p}$. TL - TL method, $\text{TL}_{\text{explore}}$ - TL method with exploration, $\text{FPD}_{\text{learn}}$ - learning FPD method, FPD - FPD method with complete knowledge.

B. Computational complexity

An important aspect of an algorithm is its computational complexity. The complexity of determining one decision rule was estimated using the ”big O” notation [17], which indicates asymptotic number of operations. It can be considered as an upper bound of the complexity. Estimating the optimal decision rule using the TL method with exploration and with normalized similarity, see Algorithm 1, takes asymptotically $O(\max(k, |\mathbf{S}|^2 \cdot |\mathbf{A}|))$ operations, where k is the number of past observations available (length od the data), $|\mathbf{S}|$ is the number of states and $|\mathbf{A}|$ is the number of actions. When determining the decision rule, the first step is computing the similarities using the data of length k. The similarities are then normalized, so a normalizing constant has to be found as a maximum value of the ideal model ${}^I p(s_t, a_t|s_{t-1})$, which has the dimensions of $|\mathbf{S}| \cdot |\mathbf{A}| \cdot |\mathbf{S}|$. Lastly, the decision rule is learnt using the computed k similarities. Multiplicative and additive constants are omitted because the ”big O” symbol describes the asymptotic long-term growth of the number of operations.

Computing the optimal decision policy with FPD learning method takes $O(\max(k, H \cdot |\mathbf{S}|^2 \cdot |\mathbf{A}|))$ operations, where H is the horizon of policy optimization. First, the unknown transition model has to be estimated using the k observations, then the optimal decision rule is computed (Theorem 1) over the horizon H. In our experiment H was set to 10, so it can be considered as a constant and omitted. Then both methods have the same theoretical asymptotic complexity $O(\max(k, |\mathbf{S}|^2 \cdot |\mathbf{A}|))$.

In practice, the omitted coefficients and constants as well as other factors are important for the true computational time. That is why it is necessary to carry out experiments measuring the real time complexity of both algorithms. An experiment was conducted comparing the CPU time required for computing the decision rule using the TL method with exploration and the learning FPD method. The CPU time was determined using the Matlab® in-built *timeit* function. It runs a specified function several times and returns the median of the elapsed times. The CPU time depends on the computer used, thus all results should be perceived as an illustration of the expected behavior. The computer used to provide the results presented here was SAMSUNG 900X3C, 2.00 GHz Intel Core i7 with 4GB RAM.

In Fig. 5, the median time complexity of computing the first decision rule with changing number of states $|\mathbf{S}|$ is shown. The number of observations was fixed at $k = 30$, the number of actions was fixed at $|\mathbf{A}| = 4$. It can be noted that the elapsed time using the FPD method increases much faster for growing $|\mathbf{S}|$ than the elapsed time using the TL method. Even though the theoretical asymptotic complexity is the same for both, the real time complexity is significantly smaller for high number of states using the TL method. The true order of complexity of the TL is possibly lower than the true order of complexity of the FPD.

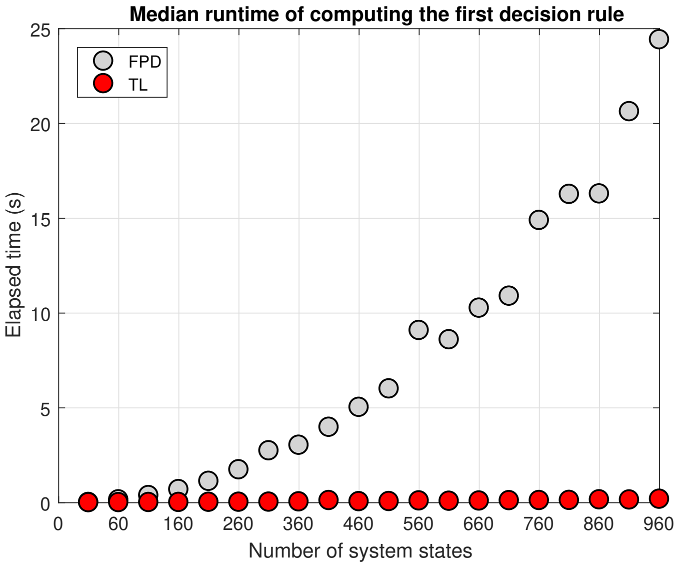

Fig. 5. Median CPU time required to determine the first decision rule after obtaining the data d_k for growing number of states using the FPD learning and the TL method with exploration.

V. Concluding remarks

The sequential decision making was considered. The paper proposes learning an optimal decision policy using the experience gained during solving other DM tasks on the same system. The approach belongs to a class of approaches like imitation learning, apprenticeship learning while uses the whole past experience available irrespectively of i) the past DM objectives; ii) quality of applied policies, and iii) the resulting overall success. The key features of the proposed solution are:

- Useful experience from the past will be amplified and transferred to a new decision policy.
- Useless (and even harmful) experience will not be neglected but transferred with much smaller weights. This allows making learned decision policy "aware" of possible bad consequences without experiencing them in reality.
- Possible non-optimality of the past decision policies serve as a natural source of exploration.
- The proposed solution is robust to the errors that can be transferred from the past as the resulting policy comprises all kinds of past behaviours: successful and not.
- There is no need to use expert's demonstration data, the good experience can come from anywhere (data coming from dozens of non-experienced drivers may give rise the decision policy that overcomes an expert's policy).

Further research will consider: i) real-application experiments, ii) the possibility to construct and work with complex multi-dimensional DM preferences.

References

[1] Y.-H. Wu, N. Charoenphakdee, H. Bao, V. Tangkaratt, and M. Sugiyama, "Imitation learning from imperfect demonstration," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 6818–6827.

[2] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: Association for Computing Machinery, 2004.

[3] K.-W. Chang, A. Krishnamurthy, A. Agarwal, H. Daumé, and J. Langford, "Learning to search better than your teacher," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15, 2015, p. 2058–2066.

[4] E. Kaufmann, M. Gehrig, P. Foehn, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Beauty and the beast: Optimal methods meet learning for drone racing," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 690–696.

[5] S. Levine, C. Theobalt, and V. Koltun, "Real-time prosody-driven synthesis of body language," *ACM Trans. Graph.*, vol. 28, no. 5, 2009.

[6] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9, 2010, pp. 661–668.

[7] M. Kárný and T. Guy, "Fully probabilistic control design," *Systems & Control Letters*, vol. 55, no. 4, pp. 259–265, 2006.

[8] M. Kárný, "Fully probabilistic design unifies and supports dynamic decision making under uncertainty," *Information Sciences*, vol. 509, pp. 104 – 118, 2020.

[9] M. Puterman, *Markov Decission Processes*. John Wiley & Sons, Inc., 1994.

[10] M. Kárný, "Towards fully probabilistic control design," *Automatica*, vol. 32, no. 12, pp. 1719–1722, 1996.

[11] M. Kárný, J. Böhm, T. V. Guy, L. Jirsa, I. Nagy, P. Nedoma, and L. Tesař, *Optimized Bayesian dynamic advising*. Springer London, 2006.

[12] V. Peterka, "Bayesian approach to system identification," in *Trends and Progress in System Identification*, P. Eykhoff, Ed. Oxford: Pergamon Press, 1981, pp. 239–304.

[13] M. Kárný, K. Macek, and T. Guy, "Lazy fully probabilistic design of decision strategies," in *Advances in Neural Networks – ISNN 2014*, Z. Zeng, Y. Li, and I. King, Eds., International Symposium on Neural Networks. Springer, 2014, pp. 140–149.

[14] T. Ferguson, "Prior distributions on spaces of probability measures," *The Annals of Statistics*, vol. 2, no. 4, pp. 615–629, 1974.

[15] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, May 1989.

[16] C. Hummersone, "Alternative box plot," 2016. [Online]. Available: https://www.github.com/IoSR-Surrey/MatlabToolbox

[17] A. Aho and J. Ullman, *Foundations of computer science*. W.H. Freeman & Co., 1994, ch. The Running Time of Programs.