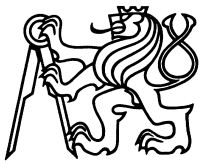
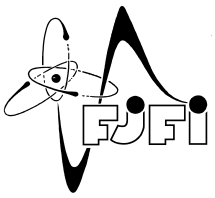


Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering



Transfer learning in sequential decision making tasks

Přenosové učení v úlohách sekvenčního rozhodování

Master's Thesis

Author: **Bc. Eliška Zugarová**

Supervisor: **Ing. Tatiana Valentine Guy, Ph.D.**

Academic year: 2019/2020

Katedra: matematiky Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	Bc. Eliška Zugarová
Studijní program:	Aplikace přírodních věd
Studijní obor:	Matematické inženýrství
Název práce (česky):	Přenosové učení v úlohách sekvenčního rozhodování
Název práce (anglicky):	Transfer learning in sequential decision making tasks

Pokyny pro vypracování:

1) Prostudujte plně pravděpodobnostní návrh rozhodovacích strategií.
2) Prozkoumejte možnosti použití již získané znalosti pro návrh budoucích rozhodovacích strategií.
3) Navrhněte vícekrokovou verzi plně pravděpodobnostního návrhu rozhodovacích strategií využívající přenosové učení pro Markovský rozhodovací proces.
4) Navrhněte a implementujte algoritmus dle návrhu z bodu 3.
5) Demonstrujte chování implementovaného postupu na simulovaných datech.
6) Vyhodnoťte dosažené výsledky a shrňte získané zkušenosti.

Doporučená literatura:

1) S. J. Pan, Q. Yang, A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22 (10), 2009, 1345--1359.
2) M. Kárný, T.V.Guy, Fully probabilistic control design. Systems and Control Letters 55 (4), 2006, 259--265.
3) M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, 2005.
4) R. Ramakrishnan, J. Shah, Towards Interpretable Explanations for Transfer Learning in Sequential Tasks. In '2016 AAAI Spring Symposium', 2016.

Jméno a pracoviště vedoucího diplomové práce:

Ing. Tatiana Valentine Guy, Ph.D.
ÚTIA AV ČR, Pod Vodárenskou veží 4, 180 00, Praha 8, Česká republika

Jméno a pracoviště konzultanta:

Datum zadání diplomové práce: 31.10.2019

Datum odevzdání diplomové práce: 4.5.2020

Doba platnosti zadání je dva roky od data zadání.

Acknowledgment:
I would like to express my gratitude and appreciation to Ing. Tatiana Valentine Guy, Ph.D. for her expert guidance, support and language assistance that have been invaluable throughout the creation of this thesis. I would also like to thank Ing. Miroslav Kárný, DrSc., Dr. Siavash Fakhimi Derakhshan, Ph.D. and Ing. Marko Ruman for their helpful advice, and my friends and family for support. This thesis has been partially supported by the project MŠMT LTC18075.

Author's declaration:
I declare that this Diploma Thesis is entirely my own work and I have listed all the used sources in the bibliography.

Prague, June 18, 2020 Bc. Eliška Zugarová

Název práce:

Přenosové učení v úlohách sekvenčního rozhodování

Autor: Bc. Eliška Zugarová

Obor: Matematické inženýrství

Druh práce: Diplomová práce

Vedoucí práce: Ing. Tatiana Valentine Guy, Ph.D., Oddělení adaptivních systémů, Ústav teorie informace a automatizace Akademie věd České republiky

Abstrakt: Diplomová práce se zabývá dynamickým rozhodovacím problémem řešeným pomocí přenosového učení. Problém je modelován s použitím plně pravděpodobnostního návrhu, který pomocí pravděpodobnostních funkcí umožňuje vyjádřit rozmanité a komplexní cíle rozhodovače. Navržená metoda přenosového učení využívá plně pravděpodobnostního návrhu a optimální strategii se učí na základě pozorování a bez znalosti modelu interagujícího systému. Předávají-li daná pozorování dostatečné množství informací, dosahuje navržená metoda výsledků srovnatelných s výsledky plně pravděpodobnostního návrhu. Metoda je navíc méně výpočetně náročná. Pro případ, kdy v pozorováních chybí důležité informace, byla navržena jednoduchá technika prozkoumávání. Tato technika přináší přijatelné vylepšení výsledků.

Klíčová slova: Markovský rozhodovací proces, plně pravděpodobnostní návrh, přenosové učení, rozhodování, teorie řízení, řízení uzavřené smyčky

Title:

Transfer learning in sequential decision making tasks

Author: Bc. Eliška Zugarová

Abstract: This thesis focuses on solving a dynamic decision problem via transfer learning. It is based on the theory of the fully probabilistic design (FPD), which is a framework that models sequential decision-making as a closed-loop. It enables to express diverse preferences and goals of the decision-maker in a probabilistic way. The proposed method of transfer learning uses FPD formulation of the problem and learns an optimal decision policy based on observed behavior. Other knowledge of the interacting system or of the preferences that guided the observed decision-making is not avaiable. When the observations contain enough information about the closed-loop, the approach provides comparable results to the FPD while being less computationally complex. In case there in a significant lack of information in the data, a simple explorative strategy is introduced. It allows to overcome the problem of missing knowledge to an acceptable degree.

Key words: closed-loop control, control theory, decision-making, fully probabilistic design, Markov decision process, transfer learning

Contents

Introduction

Decision-making (DM) is present in the lives of people since the dawn of civilization. Decisions are an ordinary part of the day for everyone; some of them are small, some are more important and complex with significant consequences. In modern days, DM has become a subject widely studied in the fields of artificial intelligence [32], machine-learning [26] or intelligent systems [31]. Findings and innovations brought by these domains enable the development of computer algorithms that help people with making informed and appropriate decisions, or replace the human element in the DM process completely. Examples of many applications include stock market prediction [6], energy management of buildings [28] and automated driving [9].

This thesis deals with sequential DM, in which the decision process is repetitive and the choice of action affects future DM conditions. The decision-maker is represented by an intelligent agent. The agent selects actions that are optimal with respect to certain DM preferences. These preferences are established before the process of choosing an action starts. The actions are taken in the context of a system the agent interacts with. The system reacts to the agent's actions by moving from one state to another. The transition dynamics of the system are, in general, unknown to the agent. To perform optimally, the agent has to consider not only the DM preferences but also has to correctly predict the system's behavior by taking advantage of knowledge already acquired by observing the system. The cycle of the agent's actions and the system's responses is called the closed-loop and it either continues indefinitely or it ends after a certain condition is met (for example after a given number of steps).

A tool commonly used to model the closed-loop of an agent and a system is the Markov decision process (MDP) [34], see [5], [17] for examples of its application. It formulates the problem in a probabilistic way. The system's behavior is described by a probability density function. The agent's preferences are defined by a reward function and the agent selects actions that maximize the expected reward.

The fully probabilistic design (FPD) [24] is another framework that solves dynamic sequential decision problems. Similarly to MDP, it uses a probabilistic formulation of the system's state transition but in addition, the agent's preferences are also defined in probabilistic terms. The resulting optimal decision policy is non-deterministic and is represented by probability density function. This allows for a more universal and precise definition of the agent's various possible preferences. FPD is a tool enabling efficient formulation of a DM problem and provides its explicit solution. However it can suffer from a high computational complexity whenever the dimensions of the problem are high. A common problem in tasks solved via MDP or FPD is the absence of knowledge of a complete model of the underlying system. The agent integrates data about previously completed similar problems to learn the model online.

The goal of this thesis is to, using FPD methodology for MDP, propose a method of designing an optimal decision policy based on available knowledge about previous observed interaction with the system. This knowledge compensates missing information about the system. The observed behavior does not generally correspond to the desirable behavior defined by the DM preferences of the agent. In other words, it is assumed that the agent has data from some past experiment based on possibly entirely different and unknown DM preferences. Even if this past experiment was solved optimally

with respect to these unknown preferences, the actions might not be optimal with respect to the current preferences. The aim is to infer an optimal decision policy using the data. The emphasis is on taking maximum advantage of the available information in order to offer a simpler solution with possibly lower computational demands than FPD.

The task stated above, of reusing past experiences to solve a new problem, can be described as a transfer learning (TL) task. TL techniques [44], [43] are techniques that compare a new problem with a similar related problem from the past and apply past findings to determine a solution. In the survey [44], it is mentioned that transfer of experience in performing a source task helps improve and speed up the learning performance of a target task. The authors of [44] and [43] note that transfer learning is often applied in the context of other learning methods such as reinforcement learning. Reinforcement learning methods [42] solve sequential DM problems and are typically used in combination with MDP problem formulations.

An example of a transfer in reinforcement learning is presented in [27]. The authors reduce the complexity of learning in MDP by reusing (transferring) samples from a source task in a target task based on the likelihood of target samples being generated by the source task models. They define a similarity measure between the tasks. Ammar et al. [7] also rely on a similarity measure between MDP samples to identify similar tasks.

Another area of research that applies source-task experience to learn an optimal policy of a target task is imitation learning [23], [29], [8]. An agent is allowed to learn a policy by observing another agent perform a similar task. In [33] the authors describe an approach of implicit imitation. An intelligent agent learns by imitation but does not necessarily repeat the observed actions. The actions are not automatically considered as appropriate. Instead, the information contained in the observed behavior is adapted to the agent's own context. Wu et al. [47] developed a method of learning an optimal policy using demonstrations with confidence scores. These scores indicate the probability that a given observed trajectory is optimal. The values of the scores are given by the expert that produced the demonstration.

Case-based reasoning [1] is a concept that uses the same logic as transfer learning, although often it is applied more in data-analysis problems, see for example [38], [10]. Other related approaches that also deal with solving problems using information about different or resembling problems are lazy learning (see [41] and [18]), apprenticeship (see for example [2], [22]), or cloning (see [13]).

The majority of the mentioned approaches rely on some expert, who provides confidence scores, sets a similarity value, or performs a special kind of demonstration. This limitation may significantly prevent broad use of these methods.

To summarize, the aim of this thesis is to develop a method that uses the frameworks of MDP and FPD and the idea of transfer learning to determine the optimal decision policy of an agent using available data.

The text is organized as follows. Chapter 1 presents necessary notation, the theory of Markov decision processes and fully probabilistic design. Chapter 2 is devoted to transfer learning and provides the algorithm of finding the optimal decision policy using observations. In Chapter 3, the exploration-exploitation tradeoff is discussed and an explorative strategy is deduced. Chapter 4 demonstrates results of simulated experiments that verify the performance of the proposed method. Finally, the thesis concludes with a summary and open questions.

Chapter 1

Mathematical preliminaries

This chapter contains an overview of the notation used throughout the text and introduces necessary mathematical concepts and results.

1.1 Notation and basic formulas

Notation is established in this section followed by the necessary mathematical formulas.

The sets of natural and real numbers are denotes as $\mathbb{N}$ and $\mathbb{R}$, respectively. Sets of values are denoted by bold capital letters, i.e. $\mathbf{X}$ is a set of values $x \in \mathbf{X}$. Lower index indicates the value of a variable at a discrete time, i.e. x_t is the value of x at time $t \in \mathbb{N}$.

The lowercase letter p is used to denote probability mass function. So $p(x)$ symbolizes the probability of the random variable x and $p(x|y)$ is the conditional probability of random variable x conditioned on random variable y. $\mathrm{E}[x]$ symbolizes expectation of random variable x, $\mathrm{E}[x|y]$ symbolizes conditional expectation of random variable x given random variable y.

Let p and $\tilde{p}$ be two arbitrary probability mass functions of a random discrete variable x with values in $\mathbf{X}$. Then the *Kullback-Leibler (KL) divergence* between p and $\tilde{p}$ is defined as

$$\boldsymbol{D}(p\|\tilde{p}) = \sum_{x\in\mathbf{X}} p(x) \ln \frac{p(x)}{\tilde{p}(x)}. \tag{1.1}$$

An important property of the KL divergence is that it is always non-negative, i.e. $\boldsymbol{D}(p\|\tilde{p}) \geq 0$, and it is zero iff $p = \tilde{p}$ almost everywhere.

The *Bayes' formula* is an important tool for describing conditional probability

$$p(x|y) = \frac{p(y|x)p(x)}{\sum_{x\in\mathbf{X}} p(y|x)p(x)}, \tag{1.2}$$

where x and y are discrete random quantities. It is used to update predictions based on a new evidence.

The *Kronecker delta* is a function of two variables of the form

$$\delta(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases} \tag{1.3}$$

The *Gamma function* is an integral function defined for complex numbers z with positive real part, i.e. $\mathrm{Re}(z) > 0$, as

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt. \tag{1.4}$$

It satisfies the recursive property: $\Gamma(z+1) = z\Gamma(z)$.

The *Multivariate beta function* is a function on an n-dimensional vector space, $n \in \mathbb{N}$, where the vectors are of the form $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\alpha_i > 0$ for all $i \leq n$. The definition of the Multivariate beta function is

$$\mathrm{B}(\alpha) = \frac{\prod_{i=1}^{n} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{n} \alpha_i)}. \tag{1.5}$$

The *Dirichlet distribution* with concentration parameter $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\alpha_i > 0$ for all $i \leq n$, is a probability distribution with probability density function

$$p(x|\alpha) \equiv \mathrm{Dir}(x, \alpha) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=1}^{n} x_i^{\alpha_i - 1}, \tag{1.6}$$

where $x = (x_1, x_2, \ldots, x_n)$ is a vector of n random quantities, for which it holds that $x_i \in [0, 1]$ for all $i \leq n$ and $\sum_{i=1}^{n} x_i = 1$.

1.2 Markov decision process

A brief summary of the theory of Markov decision processes (MDPs) [34] is provided in this section. MDP is a framework often used for solving sequential DM tasks. In this thesis, we deal with discrete sequential DM problems that are defined over a restricted horizon. The decision-maker is called an *agent* and by making decisions (choosing *actions*) it influences its environment, i.e. a *system*.

Definition 1 (Markov decision process)**.** A finite-horizon discrete-time *Markov decision process* is a tuple $\{\mathbf{T}, \mathbf{S}, \mathbf{A}, p, r\}$, where

- $\mathbf{T} = \{1, 2, \ldots, N\}$, $N \in \mathbb{N}$, is a discrete finite set of decision epochs,
- $\mathbf{S}$ is a discrete finite set of system states,
- $\mathbf{A}$ is a discrete finite set of actions available to the agent in any of the system states,
- $p : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to [0, 1]$ is a *transition model*, $p(s_t|a_t, s_{t-1})$ represents the conditional 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 chosen,
- $r : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to \mathbb{R}$ is a *reward function*, $r(s_t, a_t, s_{t-1})$ represents 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 $s_t \in \mathbf{S}$.

In MDP, DM objectives of the agent are expressed by the reward function. Desired results of the agent's DM are awarded by a large reward, while for unwanted results the agent is punished by receiving only a very small, possibly negative reward. The system moves between states stochastically, so there is uncertainty concerning future states and rewards. To ensure maximum reward, at each decision epoch the agent has to execute an action maximizing the total expected reward [16].

At decision epoch $t \in \mathbf{T}$, the transition of the system is described as $(s_{t-1}, a_t) \to s_t$. That is choosing a_t at state s_{t-1} leads to a new state s_t with probability $p(s_t|a_t, s_{t-1})$ The agent then obtains reward $r(s_t, a_t, s_{t-1})$. For this text, it is assumed that the transition model and the reward function are stationary, i.e. do not change over the course of the horizon.

The system transition depends only on the current system state thanks to the *Markov property* [34], i.e.

$$p(s_t|a_t, s_{t-1}, \ldots, s_1, a_1, s_0) = p(s_t|a_t, s_{t-1}), \tag{1.7}$$

where $s_\tau \in \mathbf{S}$ for $\tau = 0, \ldots, t$ and $a_\tau \in \mathbf{A}$ for $\tau = 1, \ldots, t$.

Selection of action a_t at decision epoch $t \in \mathbf{T}$ is controlled by a *randomized decision rule*, $p(a_t|s_{t-1})$, where $a_t \in \mathbf{A}$ and $s_{t-1} \in \mathbf{S}$. For all system states $s_{t-1} \in \mathbf{S}$, the decision rule $p(a_t|s_{t-1})$ forms a probability mass function over action set $\mathbf{A}$, so $\sum_{a_t \in \mathbf{A}} p(a_t|s_t) = 1$ and $p(a_t|s_t) \in [0, 1]$, $\forall a_t \in \mathbf{A}$. The decision rule depends on time implicitly through the indices of state s_{t-1} and action a_t, so it can be different at each decision epoch.

A sequence of decision rules over a given horizon H, $\{p(a_t|, s_{t-1})|a_t \in \mathbf{A}, s_{t-1} \in \mathbf{S}\}_{t=1}^{H}$, forms a *decision policy*. As mentioned above, the agent's goal is to maximize the expected reward. To do this, it needs to find an optimal decision policy over horizon H, which satisfies

$$\pi_{MDP}^{opt} = \underset{\{p(a_t|s_t)\}_{t=1}^{H}}{\arg\max} \sum_{t=1}^{H} E[r_t(s_t, a_t, s_{t-1})|s_{t-1}]. \tag{1.8}$$

Generally, the decision rule can be different at each decision epoch t as a result of growing knowledge in case of incomplete information about the system model[1], or due to the optimization being performed over a given horizon.

1.3 Fully probabilistic design

This section presents the fully probabilistic design (FPD), first introduced in [24] and further developed in [20], [21]. It is a dynamic DM framework that enables modeling of a closed-loop (control) system. FPD defines the desired behavior of the closed-loop (DM objective) using ideal probability mass function and determines DM policies that provide the expressed objective.

Similarly to the previous section, we deal with a discrete, finite-horizon sequential DM problem where an agent selects actions based on the targeted objectives, and an interacting system, influenced by the agent's actions, moves from one state to another. An example of such a problem could be treating a sick patient.

Example 1. A doctor (*agent*) observes the patient's state and has to choose a type of medication, with the goal of curing the patient (*system*). Some medication might work better but have possible side effects that would require another treatment. So the doctor has to optimize their decision and respond to the state of the patient's health in an adaptive way. In other words, the doctor selects optimal sequence of treatments (*DM policy*) to cure the patient with none or minimal side-effects of the treatment (*DM objective*).

With the use of notation and terminology from the previous section, the agent and the system form a closed-loop and its behavior over a given horizon is described by a joint probability function defined as follows.

Definition 2 (Closed-loop description)**.** The behavior of the closed-loop of the pair 'agent-system' until a discrete time $t \in \mathbb{N}$ is modeled by a *closed-loop description* which is a joint probability mass function $p(s_t, a_t, s_{t-1}, \ldots, s_1, a_1, s_0)$, where $s_\tau \in \mathbf{S}$, $\tau = 0, \ldots, t$, are system states, $a_\tau \in \mathbf{A}$, $\tau = 1, \ldots, t$, are the actions of the agent.

Using the Markov property (1.7) and the chain rule, the closed-loop behavior can be written in the following form

$$p(s_t, a_t, s_{t-1}, \ldots, s_1, a_1, s_0) = \prod_{\tau=1}^{t} p(s_\tau|a_\tau, s_{\tau-1}) p(a_\tau|s_{\tau-1}) p(s_0). \tag{1.9}$$

[1] If the knowledge is non-informative, the DM rule does not change significantly.

The factor $p(s_\tau|a_\tau, s_{\tau-1})$ is the *transition model* of the system, $p(a_\tau|s_{\tau-1})$ is the *decision rule* and $p(s_0)$ represents the prior distribution of the initial state, which incorporates any subjective prior knowledge.

To define FPD, the following definition of the ideal closed-loop model is introduced.

Definition 3 (Ideal closed-loop model)**.** An *ideal closed-loop model* until time $t \in \mathbb{N}$ is a joint probability mass function ${}^I p(s_t, a_t, s_{t-1}, \ldots, s_1, a_1, s_0)$, $s_\tau \in \mathbf{S}$ for $\tau = 0, \ldots, t$ and $a_\tau \in \mathbf{A}$ for $\tau = 1, \ldots, t$, that describes the desired behavior of the closed-loop composed of the agent and the system.

The ideal closed-loop model can be factorized in the same way as in (1.9)

$$ {}^I p(s_t, a_t, s_{t-1}, \ldots, s_1, a_1, s_0) = \prod_{\tau=1}^{t} {}^I p(s_\tau|a_\tau, s_{\tau-1}) {}^I p(a_\tau|s_{\tau-1}) {}^I p(s_0). \tag{1.10} $$

The factor ${}^I p(s_\tau|a_\tau, s_{\tau-1})$ is the *ideal transition model*, which describes the agent's preferences over the behavior of the system, and the factor ${}^I p(a_\tau|s_{\tau-1})$ is the *ideal decision rule* that defines the agent's preferences over actions. The last factor ${}^I p(s_0)$ represents ideal model of the initial state. We assume that the agent does not change its ideal model (1.10) during the DM process, i.e. the agent's preferences remain the same.

To follow up with the real-life example, Example 1, presented at the beginning of this section, the evolution of the patient's health follows a transition model (unknown to the doctor). It describes the probability of getting cured or developing side-effects after being administered individual medicaments. The aim of the doctor, i.e. curing the patient, can be described in probabilistic terms. The desired result is obtaining a state of health, so the ideal model assigns a high probability to each system transition that leads to this healthy state. In addition, some type of medication is known to have very serious side effects, so the ideal decision rule regulates choosing the drug by assigning it (its selection) with a smaller probability.

The aim is to solve the DM problem by choosing an optimal decision policy. Unlike solving MDP, where the DM objectives are quantified via the reward function (1.8), FPD specifies the agent's preferences via the ideal model. The optimal FPD decision policy then minimizes the Kullback-Leibler divergence (1.1) between the real closed-loop behavior (Definition 2) and the ideal closed-loop model (Definition 3) over the DM horizon. In other words, the optimal decision policy makes the closed-loop description as close as possible to the desired ideal one.

Definition 4 (Optimal FPD decision policy)**.** An *optimal decision policy* for an FPD problem is

$$ \pi^{opt}_{FPD} = \underset{\{p(a_t|s_{t-1})\}_{t=1}^{H}}{\arg\min} \boldsymbol{D}\Big(p(s_H, a_H, \ldots, s_1, a_1, s_0) \Big\| {}^I p(s_H, a_H, \ldots, s_1, a_1, s_0)\Big), \tag{1.11} $$

where $H \in \mathbb{N}$ is an optimization horizon, $s_\tau \in \mathbf{S}$, for $\tau = 0, \ldots, H$, $a_\tau \in \mathbf{A}$, for $\tau = 1, \ldots, H$, and $\boldsymbol{D}(\cdot\|\cdot)$ is the Kullback-Leibler divergence.

The optimal decision policy is a sequence of optimal decision rules, which are conditioned probability mass functions, so the problem of finding the solution is a problem of minimization (1.11) of a functional under constraints $\sum_{a_t \in \mathbf{A}} p(a_t|s_{t-1}) = 1$, $p(a_t|s_{t-1}) \in [0, 1]$, $\forall s_{t-1} \in \mathbf{S}$, $\forall t = 1, \ldots, H$.

In terms of MDP (Definition 1), the FPD can be described as follows.

Definition 5 (MDP in FPD terms)**.** An FPD description of a DM problem is composed of the following elements

- a set of decision epochs $\mathbf{T} = \{1, 2, \ldots, N\}$, where $N \in \mathbb{N}$ is a horizon,

- a set of system states $\mathbf{S}$,
- a set of actions $\mathbf{A}$,
- a transition model $p(s_t|a_t, s_{t-1})$ defined for $a_t \in \mathbf{A}$ and $s_t, s_{t-1} \in \mathbf{S}$, which comes from (1.9),
- a reward function defined for $a_t \in \mathbf{A}$ and $s_t, s_{t-1} \in \mathbf{S}$ as

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

where ${}^{I}p(s_t|a_t, s_{t-1})$ is the ideal description of the target behavior of the 'agent-system' pair.

1.3.1 Solution to FPD

An explicit solution to the FPD problem exists and is presented below. It was first introduced in [24].

Proposition 1. The optimal solution to FPD corresponding to the optimal decision policy minimizing the KL divergence (1.11) is constructed using the following equations

$$\begin{aligned}
{}^{opt}p(a_t|s_{t-1}) &= {}^{I}p(a_t|s_{t-1})\frac{\exp\big(-\alpha(a_t, s_{t-1}) - \beta(a_t, s_{t-1})\big)}{\gamma(s_{t-1})} \\
\alpha(a_t, s_{t-1}) &= \sum_{s_t \in \mathbf{S}} p(s_t|a_t, s_{t-1}) \ln \frac{p(s_t|a_t, s_{t-1})}{{}^{I}p(s_t|a_t, s_{t-1})} \\
\beta(a_t, s_{t-1}) &= -\sum_{s_t \in \mathbf{S}} \ln(\gamma(s_t)) p(s_t|a_t, s_{t-1}) \\
\gamma(s_{t-1}) &= \sum_{a_t \in \mathbf{A}} {}^{I}p(a_t|s_{t-1}) \exp\big(-\alpha(a_t, s_{t-1}) - \beta(a_t, s_{t-1})\big) \\
\gamma(s_H) &= 1
\end{aligned} \tag{1.12}$$

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

Proof. This proof is inspired by the proof from [20]. Throughout the proof, $p_{1:t}$ and ${}^{I}p_{1:t}$ stand for the closed-loop description and the ideal closed-loop description until horizon $t \in \mathbf{T}$, respectively, and $\boldsymbol{D}(p_{1:t}\|{}^{I}p_{1:t})$ denotes the KL divergence between the real and the ideal closed-loop model over a specified horizon t.

The main idea of the proof is to show iteratively that the proposed decision rules (1.12) form the solution to FPD.

The optimal decision policy is defined as the minimizer of the KL divergence (1.11)

$$\min_{\{p(a_t|s_{t-1})\}_{t=1}^{H}} \boldsymbol{D}(p_{1:H}\|{}^{I}p_{1:H}) = \min_{\{p(a_t|s_{t-1})\}_{t=1}^{H}} \sum_{\substack{a_t \in \mathbf{A}, s_t \in \mathbf{S} \\ t=1,\ldots,H}} p(s_H, a_H, \ldots, s_1, a_1, s_0) \ln\left(\frac{p(s_H, a_H, \ldots, s_1, a_1, s_0)}{{}^{I}p(s_H, a_H, \ldots, s_1, a_1, s_0)}\right) \tag{1.13}$$

Applying (1.9) and using logarithm properties on the minimized term yields

$$\sum_{\substack{a_t\in\mathbf{A},s_t\in\mathbf{S}\\ t=1,\dots,H}}\prod_{t=1}^{H}p(s_t|a_t,s_{t-1})p(a_t|s_{t-1})p(s_0)\sum_{t=1}^{H}\ln\left(\frac{p(s_t|a_t,s_{t-1})p(a_t|s_{t-1})}{{}^{I}p(s_t|a_t,s_{t-1})^{I}p(a_t|s_{t-1})}\right)$$

$$=\sum_{\substack{a_t\in\mathbf{A},s_t\in\mathbf{S}\\ t=1,\dots,H-1}}\prod_{t=1}^{H-1}p(s_t|a_t,s_{t-1})p(a_t|s_{t-1})p(s_0)\left[\sum_{t=1}^{H-1}\ln\left(\frac{p(s_t|a_t,s_{t-1})p(a_t|s_{t-1})}{{}^{I}p(s_t|a_t,s_{t-1})^{I}p(a_t|s_{t-1})}\right)\right.$$

$$\left.+\sum_{a_H\in\mathbf{A},s_H\in\mathbf{S}}p(s_H|a_H,s_{H-1})p(a_H|s_{H-1})\ln\left(\frac{p(s_H|a_H,s_{H-1})p(a_H|s_{H-1})}{{}^{I}p(s_H|a_H,s_{H-1})^{I}p(a_H|s_{H-1})}\right)\right].$$

To get the last form we used the fact that $\sum_{a_H\in\mathbf{A},s_H\in\mathbf{S}}p(s_H,a_H|s_{H-1})=1$ for fixed $s_{H-1}\in\mathbf{S}$. The prior distribution is not influenced by the decision policy, so $p(s_0)={}^{I}p(s_0)$. Therefore, the prior is not included in the logarithm part of the minimized expression.

The minimum (1.13) is then equal to

$$\min_{\{p(a_t|s_{t-1})\}_{t=1}^{H-1}}\left\{\boldsymbol{D}(p_{1:(H-1)}\|{}^{I}p_{1:(H-1)})\right.$$
$$+\min_{p(a_H|s_{H-1})}\sum_{\substack{a_t\in\mathbf{A},s_t\in\mathbf{S}\\ t=1,\dots,H-1}}\prod_{t=1}^{H-1}p(s_t|a_t,s_{t-1})p(a_t|s_{t-1})p(s_0)$$
$$\left.\cdot\underbrace{\left[\sum_{a_H\in\mathbf{A},s_H\in\mathbf{S}}p(s_H|a_H,s_{H-1})p(a_H|s_{H-1})\ln\left(\frac{p(s_H|a_H,s_{H-1})p(a_H|s_{H-1})}{{}^{I}p(s_H|a_H,s_{H-1})^{I}p(a_H|s_{H-1})}\right)\right]}_{B_H(s_{H-1})}\right\}. \tag{1.14}$$

We will now focus on the term labeled as $B_H(s_{H-1})$, where only the last decision rule for decision epoch H appears. The following equality exploits logarithmic properties and the fact that for any $s_{H-1}\in\mathbf{S}$ and any $a_H\in\mathbf{A}$, $\sum_{s_H\in\mathbf{S}}p(s_H|a_H,s_{H-1})=1$.

$$B_H(s_{H-1})=\sum_{a_H\in\mathbf{A}}p(a_H|s_{H-1})\left[\ln\left(\frac{p(a_H|s_{H-1})}{{}^{I}p(a_H|s_{H-1})}\right)+\underbrace{\sum_{s_H\in\mathbf{S}}p(s_H|a_H,s_{H-1})\ln\left(\frac{p(s_H|a_H,s_{H-1})}{{}^{I}p(s_H|a_H,s_{H-1})}\right)}_{\alpha(a_H,s_{H-1})}\right]$$

Subtracting and adding $\ln\gamma(s_{H-1})$ to $B_H(s_{H-1})$ generates

$$B_H(s_{H-1})=\sum_{a_H\in\mathbf{A}}p(a_H|s_{H-1})\ln\left(\frac{p(a_H|s_{H-1})}{\frac{{}^{I}p(a_H|s_{H-1})}{\gamma(s_{H-1})}}\right)-\ln\gamma(s_{H-1})+\sum_{a_H\in\mathbf{A}}p(a_H|s_{H-1})\alpha(a_H,s_{H-1})$$

It follows immediately that

$$B_H(s_{H-1})=\sum_{a_H\in\mathbf{A}}p(a_H|s_{H-1})\ln\left(\frac{p(a_H|s_{H-1})}{{}^{I}p(a_H|s_{H-1})\frac{\exp(-\alpha(a_H,s_{H-1}))}{\gamma(s_{H-1})}}\right)-\ln\gamma(s_{H-1}). \tag{1.15}$$

The first term in (1.15) is the KL divergence between conditional probability densities $p(\cdot|s_{H-1})$ and ${}^{I}p(\cdot|s_{H-1})\frac{\exp(-\alpha(\cdot,s_{H-1}))}{\gamma(s_{H-1})}$, and the second term is independent of $p(\cdot|s_{H-1})$. The minimum of the KL

divergence is zero and it is achieved when

$$^{opt}p(a_H|s_{H-1}) = {}^{I}p(a_H|s_{H-1})\frac{\exp(-\alpha(a_H, s_{H-1}))}{\gamma(s_{H-1})}. \tag{1.16}$$

Taking $\gamma(s_H) = 1$, we get $\beta(a_H, s_{H-1}) = 0$, then (1.16) is the optimal decision rule in the last decision epoch H.

We can now substitute the optimal decision rule into (1.14) and continue in the same fashion to obtain the remaining optimal decision rules for decision epochs $H-1, H-2, \ldots, 1$. □

Chapter 2

Transfer learning of decision policies

This chapter presents the main contribution of the thesis. In this chapter, the method of policy learning based on transfer of knowledge about one closed-loop to another closed-loop is introduced. The main idea of the approach is to, instead of directly optimizing the decision policy, estimate the optimal decision rule using the information contained in the results of a previously solved problem. This is possible under the assumption that the DM problem concerns the same system and that the system moves between states based on some fixed underlying principles[1]. In other words, the transition model is stationary in all the considered decision problems or at most slowly-time varying. Then we can use decisions made in the previous DM task even if they were obtained for different objectives.

Even if the agent that solved the past decision problem selected its decisions optimally, these actions were chosen optimally with respect to some past ideal model, which is generally different from the current one. The knowledge transformation must be adapted to take the possible difference in objectives into consideration. The optimality of past decisions is not necessary in our approach.

Let us demonstrate transfer learning on the illustrative example of the doctor and their patient mentioned in Section 1.3, Example 1. The doctor has information about the patient's past treatment (performed by another doctor) and how it affected the patient's body and health. However, the doctor *i)* does not know complete reasoning of the doctor who prescribed this treatment, and *ii)* has own experience and personal preferences for curing techniques. But the doctor can take into account the other doctor's past experience while maintaining their own preferences and selecting the best treatment for the patient. The proposed approach can form a core of an expert system helping doctors to diagnose or treat patients based on past experience [36].

2.1 Similarity of two decision-making problems

Suppose that we need to solve a DM task on some system, i.e. find an optimal DM policy that ensures reaching our DM objective with respect to the system. Let us have a record of state-action transitions describing the solution of some past DM task (possibly with different objective than the current one). We intend to use the past experience gained on the same system to learn the (approximate) optimal policy for the current DM task. The key idea is to transfer the past knowledge to the new task.

One of the main problems when transferring knowledge from one DM task to another is to recognize whether the knowledge is appropriate for the current task. To quantify a degree of suitability of the transferred knowledge we use the notion of similarity. The similarity weighs past observations with

[1]This assumption is not restrictive as any system (except for a completely random one) has some dependencies mostly given by first principles.

regard to the current ideal model. It quantifies the extent to which past behavior matches the present DM preferences. The current ideal model ${}^{I}p$ is assumed to be fixed.

Definition 6 (Similarity)**.** Let $\{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$ be a set of observations of a completed DM 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

$$\sigma_\tau = {}^{I}p(s_\tau, a_\tau | s_{\tau-1}) \in [0, 1], \tag{2.1}$$

where $(s_\tau, a_\tau, s_{\tau-1})$ is an observation of decision a_t and the corresponding state transition, and $\tau = 1, \ldots, t-1$.

The introduced definition of similarity has a clear and intuitive meaning. Whenever past observations $(s_\tau, a_\tau, s_{\tau-1})$ bring high values of the current ideal model ${}^{I}p$, the system transition $(s_{\tau-1}, a_\tau) \to s_\tau$, is similar to the targeted behavior in the current DM problem. The value of similarity is small whenever past action: *i)* simulates state transition that does not fully match the current DM preferences (expressed by the ideal model ${}^{I}p$), *ii)* is considered disadvantageous with regard to the current DM preferences. If the past system transition is desirable regarding the current DM problem, the similarity is high.

Second definition of similarity is almost identical to Definition 6, except the values are normalized.

Definition 7 (Normalized similarity)**.** Let $\{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$ be a set of observations of a completed DM task. The *normalized similarity* between the current decision problem with the ideal model ${}^{I}p$ and a past problem from decision epoch τ, $\tau = 1, \ldots, t-1$, is defined as

$$\begin{aligned} \sigma_\tau &= \frac{{}^{I}p(s_\tau, a_\tau | s_{\tau-1})}{\sigma_{max}} \in [0, 1], \text{ where} \\ \sigma_{max} &= \max_{s_t, s_{t-1} \in \mathbf{S}, a_t \in \mathbf{A}} {}^{I}p(s_t, a_t | s_{t-1}). \end{aligned} \tag{2.2}$$

Introducing a normalized version of the similarity is important because the range of possible values of the similarity is generally not the interval $[0, 1]$. Each similarity equals to an ideal likelihood of past data, so the maximum possible value of the similarity is the same as the maximum value of the ideal model. For a value of the non-normalized similarity, Definition 6, we need to know other values to conclude whether the similarity is high or low. Whereas a value of the normalized similarity, Definition 7, is informative even when it stands alone.

It is suitable to use above defined similarities in case the past data is the only information available, i.e. the agent does not have knowledge about the past ideal models. Once past ideal models (i.e. past DM preferences) are known, the similarity can be measured via any divergence on the space of probability distributions.

2.2 Bayes similarity-based transfer learning

The goal of any decision-making is to find an optimal decision policy that helps to reach DM objectives. The problem of real-life applications is a lack of knowledge, mainly precise knowledge of the system model. In this thesis we look for a model-free learning of the optimal decision policy. Bayesian estimation is used to find an estimate of the optimal DM policy from available past data [11], [30].

Consider a DM task characterized by ideal model ${}^{I}p$ and past data d_{t-1} collected up to decision epoch $t-1$ on the same system, though for a different DM problem. The data consists of a sequence of system transition triples $d_{t-1} = \{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$. Our goal is to infer the targeted decision rule at decision epoch t from data d_{t-1}.

Following Bayesian approach, let the unknown closed-loop model $p(s_t, a_t|s_{t-1})$ be parameterized as $p(s_t, a_t|s_{t-1}, \theta)$, where $\theta \in \Theta$ is an unknown finite-dimensional parameter and Θ is a continuous parameter space. The closed-loop behavior based on the observed data at decision epoch t is then described using marginalization and the chain rule as

$$\hat{p}(s_t, a_t|d_{t-1}) = \int_\Theta p(s_t, a_t, \theta|d_{t-1})d\theta = \int_\Theta p(s_t, a_t|d_{t-1}, \theta)p(\theta|d_{t-1})d\theta, \tag{2.3}$$

where $p(\theta|d_{t-1})$ represents probability distribution of the unknown parameter based on the available data d_{t-1}. Note that the closed-loop model (2.3) implicitly contains the decision rule

$$\hat{p}(a_t|d_{t-1}) = \sum_{s_t \in \mathbf{S}} \hat{p}(s_t, a_t|d_{t-1}) = \sum_{s_t \in \mathbf{S}} \int_\Theta p(s_t, a_t|d_{t-1}, \theta)p(\theta|d_{t-1})d\theta. \tag{2.4}$$

The second factor $p(\theta|d_{t-1})$ in integrals (2.3) and (2.4) is a posterior distribution of the parameter. With each new piece of data, the parameter model is updated. The update is determined using the Bayes' formula (1.2)

$$p(\theta|d_{t-1}) = \frac{p(s_{t-1}, a_{t-1}|d_{t-2}, \theta)p(\theta|d_{t-2})}{\int_\Theta p(s_{t-1}, a_{t-1}|d_{t-2}, \theta)p(\theta|d_{t-2})d\theta}.$$

But the data available do not necessarily come from the same closed-loop, the past ideal model can be different from our current ideal model. An action established as optimal using one ideal model may not be considered optimal with respect to a different ideal model. That is why we consider a weighted Bayes' formula [3], [25] and the update is expressed as

$$p(\theta|d_{t-1}) = \frac{p(s_{t-1}, a_{t-1}|d_{t-2}, \theta)^{\omega_{t-1}} p(\theta|d_{t-2})}{\int_\Theta p(s_{t-1}, a_{t-1}|d_{t-2}, \theta)^{\omega_{t-1}} p(\theta|d_{t-2})d\theta}.$$

The weights ω_{t-1} aim to correct a possible bias resulting from the difference of ideal closed-loop descriptions of the past and the current DM task. Similarity values, see Definition 6 and Definition 7, are chosen as the weights, so $\omega_{t-1} = \sigma_{t-1}$. Similarity numerically expresses how the past data fit the current ideal model ${}^I p$.

Using the weighted Bayes rule repeatedly yields

$$p(\theta|d_{t-1}) = \frac{\prod_{\tau=1}^{t-1} p(s_\tau, a_\tau|d_{\tau-1}, \theta)^{\omega_\tau} p(\theta|s_0)}{\int_\Theta \prod_{\tau=1}^{t-1} p(s_\tau, a_\tau|d_{\tau-1}, \theta)^{\omega_\tau} p(\theta|s_0)d\theta}. \tag{2.5}$$

We can now simplify the formula using the Markov property (1.7), which states that the system state transition depends on the last state only. The posterior parameter distribution (2.5) becomes

$$p(\theta|d_{t-1}) = \frac{\prod_{\tau=1}^{t-1} p(s_\tau, a_\tau|s_{\tau-1}, \theta)^{\omega_\tau} p(\theta|s_0)}{\int_\Theta \prod_{\tau=1}^{t-1} p(s_\tau, a_\tau|s_{\tau-1}, \theta)^{\omega_\tau} p(\theta|s_0)d\theta} \tag{2.6}$$

and the decision rule estimated using data d_{t-1} (2.4) is

$$\hat{p}(a_t|s_{t-1}) = \sum_{s_t \in \mathbf{S}} \int_\Theta p(s_t, a_t|s_{t-1}, \theta)p(\theta|d_{t-1})d\theta. \tag{2.7}$$

The remainder of this section describes how to find an explicit form of the estimated optimal decision rule using past data. The solution is suggested by the ensuing proposition.

Proposition 2. The estimate of the optimal decision rule based on observations $d_{t-1} = \{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$ available at decision epoch $t \in \mathbf{T}$ has the form

$$ {}^{opt}\hat{p}(a_t|s_{t-1}) = \frac{\sum_{\tau=1}^{t-1} \omega_\tau \delta(a_t, a_\tau)\delta(s_{t-1}, s_{\tau-1}) + \sum_{s\in\mathbf{S}} \nu_0^{s,a_t|s_{t-1}}}{\sum_{\tau=1}^{t-1} \omega_\tau \delta(s_{t-1}, s_{\tau-1}) + \sum_{s\in\mathbf{S}} \sum_{a\in\mathbf{A}} \nu_0^{s,a|s_{t-1}}}, \tag{2.8} $$

for all $a_t \in \mathbf{A}$ and $s_{t-1} \in \mathbf{S}$, where ω_τ, $\tau = 1, \ldots, t-1$, are weights representing similarities (2.1) or normalized similarities (2.2). The function $\delta(\cdot,\cdot)$ is the Kronecker delta (1.3) and $\nu_0^{s,a|s_{t-1}} > 0$, $s \in \mathbf{S}$, $a \in \mathbf{A}$, represent prior knowledge about the closed-loop model.

Proof. Throughout the proof, $(s', a) \to s$ denotes a system state transition $(s_{\tau-1}, a_\tau) \to s_\tau$, where $\tau \in \mathbf{T}$ is some past decision epoch, $\tau < t$.

We define the parametrization (2.3) of the unknown closed-loop description so that the parameter space is

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

and

$$ \begin{aligned} \theta_{s_t,a_t|s_{t-1}} &\equiv p(s_t, a_t|s_{t-1}, \theta) \\ &= \prod_{s'\in\mathbf{S}} \prod_{a\in\mathbf{A}} \prod_{s\in\mathbf{S}} \theta_{s,a|s'}^{\delta(s,s_t)\delta(a,a_t)\delta(s',s_{t-1})}. \end{aligned} \tag{2.9} $$

We assume the observation of the initial system state s_0 does not change the prior beliefs about the parameter of the closed-loop model, i.e. $p(\theta|s_0) = p(\theta)$. This assumption can also be justified by the Bayes' formula (1.2)

$$ p(\theta|s_0) = \frac{p(s_0|\theta)p(\theta)}{\int_\Theta p(s_0|\theta)p(\theta)d\theta}, $$

because the above expression is equal to $p(\theta)$ when the initial state is considered as an initial condition not dependent of the parameter, i.e. $p(s_0|\theta) = p(s_0)$ [30].

Additionally, we assume that $\theta_{\cdot,\cdot|s_{t-1}} = p(\cdot,\cdot|s_{t-1})$ follows multinomial distribution and the prior distribution of the model parameter is a product of Dirichlet distributions (1.6). 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 (from the same family of distributions) for multinomial distribution sampling [15]. The prior is expressed as

$$ p(\theta) = \prod_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod_{a\in\mathbf{A}} \prod_{s\in\mathbf{S}} \theta_{s,a|s'}^{\nu_0^{s,a|s'}-1} = \prod_{s'\in\mathbf{S}} \mathrm{Dir}\left(\theta_{\cdot,\cdot|s'}, \nu_0^{\cdot,\cdot|s'}\right). \tag{2.10} $$

For all $s' \in \mathbf{S}$, $\nu_0^{\cdot,\cdot|s'}$ is a vector of values $\nu_0^{s,a|s'} > 0$, $s \in \mathbf{S}$, $a \in \mathbf{A}$, and $\theta_{\cdot,\cdot|s'}$ is a vector of parameters from a subspace $\Theta_{s'} = \{\theta_{s,a|s'} | s \in \mathbf{S}, a \in \mathbf{A}, \theta_{s,a|s'} \in \Theta\} \subset \Theta$.

We will now focus on rewriting the posterior distribution. It can be expressed using the form (2.6) and the proposed parametrization (2.9) as

$$ p(\theta|d_{t-1}) = \frac{\prod_{\tau=1}^{t-1} \prod_{s'\in\mathbf{S}} \prod_{a\in\mathbf{A}} \prod_{s\in\mathbf{S}} \theta_{s,a|s'}^{\omega_\tau\delta(s,s_\tau)\delta(a,a_\tau)\delta(s',s_{\tau-1})} p(\theta)}{\int_\Theta \prod_{\tau=1}^{t-1} \prod_{s'\in\mathbf{S}} \prod_{a\in\mathbf{A}} \prod_{s\in\mathbf{S}} \theta_{s,a|s'}^{\omega_\tau\delta(s,s_\tau)\delta(a,a_\tau)\delta(s',s_{\tau-1})} p(\theta)d\theta}. $$

Next, we substitute the chosen prior (2.10) and we apply the exponent properties.

$$p(\theta|d_{t-1}) = \frac{\prod\limits_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod\limits_{a\in\mathbf{A}}\prod\limits_{s\in\mathbf{S}} \theta_{s,a|s'}^{\sum_{\tau=1}^{t-1}\omega_\tau\delta(s,s_\tau)\delta(a,a_\tau)\delta(s',s_{\tau-1})+\nu_0^{s,a|s'}-1}}{\int\limits_{\Theta}\prod\limits_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod\limits_{a\in\mathbf{A}}\prod\limits_{s\in\mathbf{S}} \theta_{s,a|s'}^{\sum_{\tau=1}^{t-1}\omega_\tau\delta(s,s_\tau)\delta(a,a_\tau)\delta(s',s_{\tau-1})+\nu_0^{s,a|s'}-1} d\theta} \tag{2.11}$$

Let us now introduce a notation that will be used throughout the rest of the proof. We will denote the sum of exponents from the above expression as

$$V_{t-1}^{s,a|s'} = \sum_{\tau=1}^{t-1} \omega_\tau\delta(s,s_\tau)\delta(a,a_\tau)\delta(s',s_{\tau-1}) + \nu_0^{s,a|s'}. \tag{2.12}$$

It is easily seen that the definition (2.12) is recursive

$$\begin{aligned} V_\tau^{s,a|s'} &= \omega_\tau\delta(s,s_\tau)\delta(a,a_\tau)\delta(s',s_{\tau-1}) + V_{\tau-1}^{s,a|s'},\ \tau = 1,\dots,t-1, \\ V_0^{s,a|s'} &= \nu_0^{s,a|s'}. \end{aligned} \tag{2.13}$$

Next, by using the definition (2.13) in (2.11) we get the following form of the posterior parameter distribution

$$p(\theta|d_{t-1}) = \frac{\prod\limits_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod\limits_{a\in\mathbf{A}}\prod\limits_{s\in\mathbf{S}} \theta_{s,a|s'}^{V_{t-1}^{s,a|s'}-1}}{\int\limits_{\Theta}\prod\limits_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod\limits_{a\in\mathbf{A}}\prod\limits_{s\in\mathbf{S}} \theta_{s,a|s'}^{V_{t-1}^{s,a|s'}-1} d\theta} \tag{2.14}$$

It is clear from (2.14) that the posterior distribution is, as mentioned above, a product of Dirichlet distributions (1.6) with concentration parameters equal to vectors $V_{t-1}^{\cdot,\cdot|s'}$, $s' \in \mathbf{S}$.

Substituting the posterior distribution (2.14) into the estimate of the decision rule (2.7) and using the parametrization (2.9) gives

$$\hat{p}(a_t|s_{t-1}) = \frac{\sum\limits_{s_t\in\mathbf{S}}\int\limits_{\Theta}\prod\limits_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod\limits_{a\in\mathbf{A}}\prod\limits_{s\in\mathbf{S}} \theta_{s,a|s'}^{\delta(s,s_t)\delta(a,a_t)\delta(s',s_{t-1})}\theta_{s,a|s'}^{V_{t-1}^{s,a|s'}-1} d\theta}{\int\limits_{\Theta}\prod\limits_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod\limits_{a\in\mathbf{A}}\prod\limits_{s\in\mathbf{S}} \theta_{s,a|s'}^{V_{t-1}^{s,a|s'}-1} d\theta}.$$

If we set the similarity of the current decision epoch $\omega_t = 1$, we can use the recursive definition (2.13) to obtain $V_t^{s,a|s'}$ and get

$$\hat{p}(a_t|s_{t-1}) = \frac{\sum\limits_{s_t\in\mathbf{S}}\int\limits_{\Theta}\prod\limits_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod\limits_{a\in\mathbf{A}}\prod\limits_{s\in\mathbf{S}} \theta_{s,a|s'}^{V_t^{s,a|s'}-1} d\theta}{\int\limits_{\Theta}\prod\limits_{s'\in\mathbf{S}} \frac{1}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \prod\limits_{a\in\mathbf{A}}\prod\limits_{s\in\mathbf{S}} \theta_{s,a|s'}^{V_{t-1}^{s,a|s'}-1} d\theta} = \frac{\sum\limits_{s_t\in\mathbf{S}}\int\limits_{\Theta}\prod\limits_{s'\in\mathbf{S}} \frac{\mathrm{B}\left(V_t^{\cdot,\cdot|s'}\right)}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \mathrm{Dir}\left(\theta_{\cdot,\cdot|s'}, V_t^{\cdot,\cdot|s'}\right)}{\int\limits_{\Theta}\prod\limits_{s'\in\mathbf{S}} \frac{\mathrm{B}\left(V_{t-1}^{\cdot,\cdot|s'}\right)}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)} \mathrm{Dir}\left(\theta_{\cdot,\cdot|s'}, V_{t-1}^{\cdot,\cdot|s'}\right)} \tag{2.15}$$

In the last expression in (2.15), the Beta function coefficients can be put in front of the integrals in the numerator and the denominator, and the integrals are equal to one because they are $\int_\Theta p(\theta|d_\tau)d\theta$,

$\tau = t, t-1$. We then obtain

$$\hat{p}(a_t|s_{t-1}) = \frac{\sum\limits_{s_t\in\mathbf{S}}\prod\limits_{s'\in\mathbf{S}}\frac{\mathrm{B}\left(V_t^{\cdot,\cdot|s'}\right)}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)}}{\prod\limits_{s'\in\mathbf{S}}\frac{\mathrm{B}\left(V_{t-1}^{\cdot,\cdot|s'}\right)}{\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)}} = \frac{\sum\limits_{s_t\in\mathbf{S}}\prod\limits_{s'\in\mathbf{S}}\mathrm{B}\left(V_t^{\cdot,\cdot|s'}\right)}{\prod\limits_{s'\in\mathbf{S}}\mathrm{B}\left(V_{t-1}^{\cdot,\cdot|s'}\right)} = \frac{\sum\limits_{s_t\in\mathbf{S}}\prod\limits_{s'\in\mathbf{S}}\frac{\prod_{s\in\mathbf{S}}\prod_{a\in\mathbf{A}}\Gamma\left(V_t^{s,a|s'}\right)}{\Gamma\left(\sum_{a\in\mathbf{A}}\sum_{a\in\mathbf{A}}V_t^{s,a|s'}\right)}}{\prod\limits_{s'\in\mathbf{S}}\frac{\prod_{s\in\mathbf{S}}\prod_{a\in\mathbf{A}}\Gamma\left(V_{t-1}^{s,a|s'}\right)}{\Gamma\left(\sum_{s\in\mathbf{S}}\sum_{a\in\mathbf{A}}V_{t-1}^{s,a|s'}\right)}}$$

$$= \frac{\sum\limits_{s_t\in\mathbf{S}}\prod\limits_{s'\in\mathbf{S}}\frac{\prod_{s\in\mathbf{S}}\prod_{a\in\mathbf{A}}\Gamma\left(V_{t-1}^{s,a|s'}+\delta(s,s_t)\delta(a,a_t)\delta(s',s_{t-1})\right)}{\Gamma\left(\sum_{s\in\mathbf{S}}\sum_{a\in\mathbf{A}}V_{t-1}^{s,a|s'}+\delta(s,s_t)\right)}}{\prod\limits_{s'\in\mathbf{S}}\frac{\prod_{s\in\mathbf{S}}\prod_{a\in\mathbf{A}}\Gamma\left(V_{t-1}^{s,a|s'}\right)}{\Gamma\left(\sum_{s\in\mathbf{S}}\sum_{a\in\mathbf{A}}V_{t-1}^{s,a|s'}\right)}}.$$

In the above, the second equality was obtained by factoring out the component $\mathrm{B}\left(\nu_0^{\cdot,\cdot|s'}\right)$, which is a normalizing constant of the prior distribution and does not depend on s_t. In the third equality, the definition of the Beta function (1.5) was used. Now we apply the recursive property of the Gamma function (1.4) and get

$$\begin{aligned}\hat{p}(a_t|s_{t-1}) &= \sum_{s_t\in\mathbf{S}}\frac{V_{t-1}^{s_t,a_t|s_{t-1}}}{\sum_{s\in\mathbf{S}}\sum_{a\in\mathbf{A}}V_{t-1}^{s,a|s_{t-1}}}\\ &= \sum_{s_t\in\mathbf{S}}\frac{\sum_{\tau=1}^{t-1}\omega_\tau\delta(s_t,s_\tau)\delta(a_t,a_\tau)\delta(s_{t-1},s_{\tau-1})+\nu_0^{s_t,a_t|s_{t-1}}}{\sum_{\tau=1}^{t-1}\omega_\tau\delta(s_{t-1},s_{\tau-1})+\sum_{s\in\mathbf{S}}\sum_{a\in\mathbf{A}}\nu_0^{s,a|s_{t-1}}}.\end{aligned} \tag{2.16}$$

After the summation over $s_t \in \mathbf{S}$, the last expression in (2.16) is the desired formula, so in conclusion we obtain

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

□

An algorithm describing approximation of the optimal decision policy is shown in Figure 2.1. The algorithm uses the normalized similarity (2.2).

Data: past data $d_{t-1} = \{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$, new ideal ${}^{I}p$, horizon $N > t-1$
for $\tau = 1, \ldots, t-1$ **do**
  Compute weight $\omega_\tau \equiv \sigma_\tau = \frac{{}^{I}p(s_\tau,a_\tau|s_{\tau-1})}{\sigma_{max}}$ (2.2);
end
while $t \le N$ **do**
  Learn and apply the optimal decision rule ${}^{opt}\hat{p}(a_t|s_{t-1})$ (2.8);
  Observe new state transition $(s_{t-1}, a_t) \to s_t$;
  Add new observation into the data sequence: $d_t = d_{t-1} \cup \{(s_t, a_t, s_{t-1})\}$;
  Calculate new weight $\omega_t \equiv \sigma_t = \frac{{}^{I}p(s_t,a_t|s_{t-1})}{\sigma_{max}}$ (2.2);
  $t = t+1$;
end

Figure 2.1: The DM algorithm with similarity-based transfer learning.

Summarizing remarks: The proposed approach relies on the rich though non-optimal past experience. Whenever the experience is insufficient, i.e. few (or none) observed state-action transitions, some additional experiments should be performed first. The algorithm needs a well-specified ideal model (1.10) that describes the agent's DM preferences. If the DM preferences are expressed vaguely or are not feasible, the resulting policy will rely on the transitions that occurred in the past and it can be expected that the past successful behavior will be transferred. The approach has a potential for learning the past DM objectives (cf. inverse reinforcement learning).

Chapter 3

Exploration

The observation-based learning techniques presented in the previous chapter exploit available data by determining decision rules based on observed closed-loop behavior. Exploitation of these rules can yield similar results as those detected in the data. However, potentially, there exist policies that bring the agent closer to its preferences but that are not employed simply because we only imitate past behavior. If past data are incomplete, were obtained with ideal model significantly differing from the current one, or if past decisions were not optimal, the need to explore overlooked and possibly superior actions and states is important. The issue of how to add exploration into the process is dealt with in this chapter.

3.1 Exploration-exploitation dilemma

Exploration-exploitation tradeoff is a compromise between utilizing the currently available knowledge and obtaining new knowledge to improve performance. Exploiting the knowledge at hand represents a safe approach that leads to results optimal according to current assumptions and experience but does not enable us to advance. Exploration can provide important new findings about the environment but can be risky as it can lead to unstable behavior. If the information we already have leads to the best possible decision, we lose time and profit by exploring fruitlessly.

This tradeoff is a well-studied subject in sequential DM, see for example [37],[14] or [40]. Multi-armed bandit problem, first formulated in [39], is a problem designed to model exploration and exploitation balance. As it has been studied for many decades, there exist plenty of algorithms solving it. Among the most used simple solving strategies are: the ϵ-greedy strategy, introduced in [46], the ϵ-first strategy, and the ϵ-decreasing strategy [35]. Throughout the text, the term *strategy* is reserved for exploration strategies and should not be confused with the term *policy*, which refers to the DM policy.

The *ϵ-greedy strategy* chooses the currently optimal action with probability 1-ϵ, and a random action with probability ϵ, $\epsilon \in [0, 1]$. This exploration can be either adopted at every stage of the decision process, or only during a fixed period and then turned off when the performance improves. In the context of this thesis, the ϵ-greedy exploration is used in the following way. The decision rule at decision epoch $t \in \mathbf{T}$ in state $s_t \in \mathbf{S}$ can be written as

$$^{\epsilon}p(a_t|s_t) = \begin{cases} ^{opt}\hat{p}(a_t|s_t) & \text{if } \xi_t < \epsilon, \\ \frac{1}{|\mathbf{A}|} & \text{otherwise,} \end{cases} \tag{3.1}$$

for all actions $a_t \in \mathbf{A}$, where ξ_t is generated randomly at each decision epoch from the uniform distribution on the interval [0, 1], and $^{opt}\hat{p}(a_t|s_t)$ is the estimated optimal decision rule (2.8).

The ϵ-*first strategy* performs exploration during the first ϵh steps, where h is the DM horizon, so initially only random actions are taken. Then the information gathered is exploited during the rest of the dedicated time. The TL decision rule with the ϵ-first exploration applied is for all actions $a_t \in \mathbf{A}$

$$^{\epsilon}p(a_t|s_t) = \begin{cases} \frac{1}{|\mathbf{A}|} & \text{if } t < \epsilon h, \\ {}^{opt}\hat{p}(a_t|s_t) & \text{otherwise,} \end{cases} \tag{3.2}$$

where h is the DM horizon and $\epsilon \in [0, 1]$.

Lastly, the ϵ-*decreasing strategy* is similar to the ϵ-greedy strategy except the exploration rate ϵ decreases over time. This exploration method is implemented in TL as follows

$$^{\epsilon}p(a_t|s_t) = \begin{cases} \frac{1}{|\mathbf{A}|} & \text{if } \xi_t < \epsilon_t, \\ {}^{opt}\hat{p}(a_t|s_t) & \text{otherwise,} \end{cases} \tag{3.3}$$

where ξ_t is randomly generated from a uniform distribution on the interval $[0, 1]$ and $\{\epsilon_t\}_{t=1}^{h}$ is a sequence of numbers from $[0, 1]$ that converges to zero. For instance, this sequence of exploration rates can be defined as $\epsilon_t = \frac{\epsilon_0}{t}$, $\epsilon_0 \in (0, 1]$, [12].

Using any of these strategies requires selecting value of parameter ϵ. This can be done heuristically, or an appropriate value can be determined adaptively during the course of the DM (see [45] for example). However, it usually increases the computational time and the complexity. The choice of parameter ϵ is important as it controls the tradeoff, which is the compromise between exploiting available knowledge and exploring to gather new information. Using the ϵ-decreasing exploration also requires setting the decay rate, i.e. defining the decreasing sequence $\{\epsilon_t\}_{t=1}^{h}$.

The main advantage of these exploration practices is that they are easy to implement and generally do not cause much additional computations.

3.2 Adjusted exploration in the proposed transfer learning

In order to include exploration while preventing over-exploration, we offer a modification of the strategies presented in the previous section. Over-exploration can worsen the DM results when the data are sufficiently good, for example when past objectives are similar to the current objective and imitating past behavior is therefore efficient.

Similarities (2.1) and (2.2) indicate the usefulness of past data for the current ideal model. Small similarities indicate non-optimality of the past decisions with regards to the current DM task. Therefore, it shows a gap in knowledge that suggests the need of exploration. Through exploration, more information about the system can be gathered.

Based on the idea presented in Section 3.1, we define a criterion that helps to recognize whether exploration should be applied. If the average of past $m \in \mathbb{N}$ similarities (2.1), (2.2) is lower than a given threshold from the interval $[0, 1]$, ϵ-exploration is used. Let us denote the threshold by the letter $q \in [0, 1]$. The choice of the optimal decision rule at decision epoch t can be described as

$$p(a_t|s_{t-1}) = \begin{cases} {}^{\epsilon}p(a_t|s_{t-1}), & \text{if } \frac{1}{m}\sum_{\tau=t-m}^{t-1}\omega_\tau < q \\ {}^{opt}\hat{p}(a_t|s_{t-1}), & \text{otherwise,} \end{cases} \tag{3.4}$$

where ${}^{\epsilon}p(a_t|s_{t-1})$ uses ϵ-greedy exploration (3.1), ${}^{opt}\hat{p}(a_t|s_t)$ is the estimate of the optimal decision rule (2.8), $q \in [0, 1]$, and $0 < m < t$.

The threshold q separates similarities into two categories: sufficiently high and low. In case of normalized similarity, see Definition 7, the threshold can be set at a fixed value and be universal across

different problems with data of different quality (suitability for the current goals). A reasonable choice is for example $q = 0.5$. If similarities are not normalized, a mean, median or central value of all similarities can be used as the threshold for each individual problem. However, in that case, if all computed similarities are small, these values are also small and the use of exploration is not sufficient. Similarly, whenever all computed similarities are high, exploration is unreasonable but it is employed in case the threshold is equal to the mean or the median and these values are also high. This suggests that normalization is preferable.

To be able to perform normalization, the maximum possible value of the similarity (the maximum value of the ideal model) has to be determined and that potentially increses computational cost. On the other hand, computing the mean or median requires going through all past observations, and the values would have to be recomputed with each new observation, which also increases the complexity.

Figure 3.1 shows the resulting algorithm of the DM described in Chapter 2 with incorporated exploration and using normalized similarity (2.2).

Data: past data $d_{t-1} = \{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{t-1}$, ideal model ${}^{I}p$, horizon $N > t-1$, $0 < m < t$, $q \in [0,1]$
for $\tau = 1, \ldots, t-1$ **do**
  Compute weight $\omega_\tau \equiv \sigma_\tau = \frac{{}^{I}p(s_\tau, a_\tau | s_{\tau-1})}{\sigma_{max}}$ (2.2);
end
while $t \leq N$ **do**
  Learn the optimal decision rule ${}^{opt}\hat{p}(a_t|s_{t-1})$ (2.8);
  if $\frac{1}{m}\sum_{\tau=t-m}^{t-1} \omega_\tau < q$ **then**
    Generate ξ_t from the uniform distribution on the interval [0,1];
    if $\xi_t < \epsilon$ **then**
      Exploration: Use the random decision rule $p_t(a_t|s_{t-1}) = \frac{1}{|\mathbf{A}|}$;
    else
      Exploitation: Use the optimal decision rule ${}^{opt}\hat{p}(a_t|s_{t-1})$ (2.8);
    end
  else
    Exploitation: Use the optimal decision rule ${}^{opt}\hat{p}(a_t|s_{t-1})$ (2.8);
  end
  Observe new state transition $(s_{t-1}, a_t) \to s_t$;
  Add new observation into the data sequence: $d_t = d_{t-1} \cup \{(s_t, a_t, s_{t-1})\}$;
  Calculate new weight $\omega_t \equiv \sigma_t = \frac{{}^{I}p(s_t, a_t | s_{t-1})}{\sigma_{max}}$ (2.2);
  $t = t + 1$;
end

Figure 3.1: The decision-making algorithm with similarity-based transfer learning and exploration.

Chapter 4

Simulated experiments

In this chapter, the performance of the method of similarity-based transfer learning (TL) presented in Chapter 2 is verified and validated through a series of simulated experiments. It is compared with the performance of the fully probabilistic design (FPD), see Section 1.3.1.

The proposed method of TL was tested with the use of generated data that imitated the agent's observation of a past DM task characterized by unknown DM preferences. These unknown preferences were specified by various types of past ideal model that corresponded to the past DM task. The FPD method was used to generate the past data. The agent then solved the current DM task via transfer learning using the generated past data without knowing what the past DM preferences were or whether they were similar to the agent's current DM preferences defined by the current ideal model. The goal was to determine how the use of various generated data influences the agent's behavior and to compare the results with results of the FPD method. Each experiment was repeated 100 times to ensure that the outcome shown is not dependent on the simulation settings used.

The methods and experiments were implemented in Matlab R2016b®. The seed for reproducibility of results was set to 10. Boxplot figures were generated using Alternative box plot function for Matlab from the IoSR Matlab Toolbox [19].

General settings: The system interacting with the agent was a discrete system with three possible states from the state space $\mathbf{S} = \{s^1, s^2, s^3\}$. The agent could choose between four different actions from the action space $\mathbf{A} = \{a^1, a^2, a^3, a^4\}$. The transition model $p(s_t|a_t, s_{t-1})$, which characterizes the system, was different each time the simulation was repeated because the coefficients of the transition model were generated randomly. Initial state s_0 was also chosen randomly from the uniform distribution. The horizon, i.e. the length of the DM process, was set to $h = 100$ decision epochs. The agent used available past (demonstration) data to find a decision policy optimal with respect to ideal model ${}^I p$. Normalized version of the similarity (2.2) was utilized to weight the observed state-action transitions.

Generating the data: The demonstration data were generated for each simulated experiment and each time the simulation was repeated. The FPD method with full knowledge of the transition model $p(s_t|a_t, s_{t-1})$ was used to simplify the verification of the proposed approach. The DM preferences used in the simulation were specified by the past ideal model, denoted as ${}^I\tilde{p}$. At each decision epoch, the optimal decision policy was determined using the exact solution of the FPD (see Proposition 1). The horizon of the policy optimization was $H = 10$ decision epochs. The resulting closed-loop behavior was observed over $k = 60$ decision epochs, so the past data expressing past experience were $d_{60} = \{(s_\tau, a_\tau, s_{\tau-1})\}_{\tau=1}^{60}$. These data were further used for transfer learning.

4.1 Experiments for DM preferences considering only states

This section presents experiments where the agent's objectives comprised of preferences over system states without preferences over actions. This is a DM objective typical in situations where there are no restrictions on actions. The ideal decision rule (1.10) was a uniform probability mass function: ${}^{I}p(a_t|s_{t-1}) = \frac{1}{|\mathbf{A}|} = 0.25$, for all $a_t \in \mathbf{A}$, $s_{t-1} \in \mathbf{S}$.

Ideal models for demonstration data: Ideals serving for generating the demonstration data are labeled by tilde from here onwards. 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 also uniform.

The first past ideal model, labeled as ${}^{I}\tilde{p}_1$, favored state s^1 and the ideal transition probability was defined as

$$\begin{aligned} {}^{I}\tilde{p}_1(s_t = s^1|a_t, s_{t-1}) &= 0.99998, \\ {}^{I}\tilde{p}_1(s_t \neq s^1|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{4.1}$$

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

The second one, represented by the symbol ${}^{I}\tilde{p}_{1,2}$, was characterized by the preference of states s^1 and s^2, both were favored equally. For all $a_t \in \mathbf{A}$ and all $s_{t-1} \in \mathbf{S}$ it was defined as

$$\begin{aligned} {}^{I}\tilde{p}_{1,2}(s_t = s^1|a_t, s_{t-1}) &= 0.499995, \\ {}^{I}\tilde{p}_{1,2}(s_t = s^2|a_t, s_{t-1}) &= 0.499995, \\ {}^{I}\tilde{p}_{1,2}(s_t = s^3|a_t, s_{t-1}) &= 0.00001. \end{aligned} \tag{4.2}$$

The third past ideal model, symbolized by ${}^{I}\tilde{p}_3$, favored state s^3. The ideal transition model was

$$\begin{aligned} {}^{I}\tilde{p}_3(s_t = s^3|a_t, s_{t-1}) &= 0.99998, \\ {}^{I}\tilde{p}_3(s_t \neq s^3|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{4.3}$$

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

Current ideal model of the agent: The current DM task was characterized by ideal model ${}^{I}p$ with the DM objective to reach state s^1. The ideal transition model was thus the same as ${}^{I}\tilde{p}_1$ (4.1)

$$\begin{aligned} {}^{I}p(s_t = s^1|a_t, s_{t-1}) &= 0.99998, \\ {}^{I}p(s_t \neq s^1|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{4.4}$$

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

The performance of the proposed TL method was measured by *gain*, which was defined as the overall number of occurrences of the targeted state s^1. Prior distribution (2.10) concentration parameters 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 describes the situation where no prior information about the parameter of the closed-loop model is available.

4.1.1 Choice of exploration strategy

As mentioned in Chapter 3, it is important to include some explorative strategy in the model. It helps to overcome a potential problem of missing information in the available data d_{60}. Chapter 3 presents three simple and well-known exploration strategies, and one that was designed on their basis. It is necessary to verify that those exploration strategies behave as expected and choose an appropriate value of the exploration rate ϵ.

Demonstration data with ${}^I\tilde{p}_3$: To imitate the situation where the current DM objectives, described by the ideal model ${}^Ip$ (4.4), are completely different from the past DM objectives, the demonstration data d_{60} for the first three experiments were generated using the ideal transition model ${}^I\tilde{p}_3$ (4.3). The past data thus did not contain all the information needed to achieve the aims defined by ${}^Ip$.

In the first experiment, the ϵ-greedy exploration strategy (3.1) was used, with the value of exploration rate ϵ changing from 0.0 to 0.5. The second experiment was conducted with the ϵ-first exploration strategy (3.2) for changing ϵ. The results of both experiments can be seen in Figure 4.1, which shows a boxplot of the resulting gains, i.e. occurrences of state s^1, for different ϵ. It can be noted that for both exploration strategies, the results improve with growing ϵ until $\epsilon = 0.3$. For higher ϵ, the gains remain roughly the same. The ϵ-first exploration strategy performs slightly better than the ϵ-greedy exploration strategy.

In the third experiment, the ϵ-decreasing exploration strategy (3.3) was applied with the decreasing sequence defined as $\epsilon_t = \frac{\epsilon_0}{t}$. ϵ_0 changed values from 0.0 to 1.0. Results are illustrated in Figure 4.2, it can be seen that the trend of growing gains was different than for the ϵ-greedy exploration and the ϵ-first exploration strategy, see Figure 4.1. Generally, the highest gains were obtained for the highest initial exploration rate $\epsilon_0 = 1$.

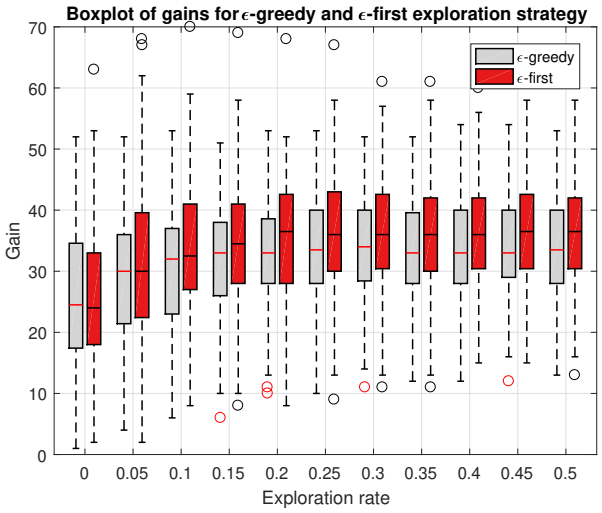


Figure 4.1: Boxplot of gains achieved using the ϵ-greedy exploration and the ϵ-first exploration strategy with changing value of exploration rate ϵ.

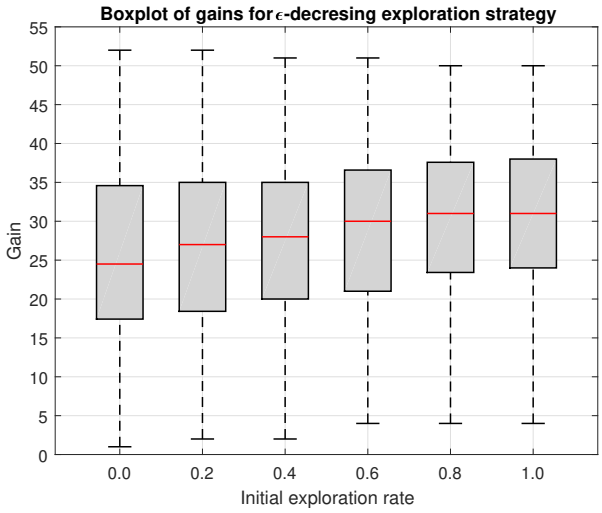


Figure 4.2: Boxplot of gains achieved using the ϵ-decreasing exploration strategy with changing value of initial exploration rate ϵ_0.

Next, all the above mentioned exploration strategies were compared with the adjusted exploration strategy (3.4) and with the case of no exploration, i.e. $\epsilon = 0$. The value of exploration rate was fixed at $\epsilon = 0.3$ for the ϵ-greedy, the ϵ-first and the adjusted exploration strategy, and $\epsilon_0 = 1$ for the ϵ-decreasing exploration strategy. The parameters of the adjusted exploration strategy were set to $q = 0.4$ and $m = 10$

based on simulations described below. q is the threshold determining low average similarity and m is the number of past similarities that are averaged in order to assess the need of employing an exploration strategy (for details see Section 3.2).

Results of the comparison of exploration strategies in a case where demonstration data d_{60} were generated with the ideal model ${}^I\tilde{p}_3$ (4.3) are given in Figure 4.5. It shows that when the data do not contain the information necessary, every exploration strategy helps to improve the gains. The best results were obtained using the ϵ-first exploration strategy, the weakest effect of exploration was produced by the ϵ-decreasing exploration strategy.

Choice of q and m: A series of experiments with different demonstration data was conducted with the aim of choosing the best values of the parameters q and m (see Section 3.2) of the adjusted exploration strategy. The choice of q was based on results of a simulation, in which the adjusted exploration strategy was used with $\epsilon = 0.3$ for changing values of q, and the demonstration data d_{60} were generated using either ${}^I\tilde{p}_3$ (4.3) or ${}^I\tilde{p}_1$ (4.1) while the current ideal was ${}^Ip$ (4.4), i.e. the same as ${}^I\tilde{p}_1$. Similar simulation was done also for changing values of m, namely values $5, 10, 15, 20$. However, the results for all of the values were nearly identical.

As can be seen in Figure 4.3, which shows gains in case of ${}^I\tilde{p}_3$, and in Figure 4.4, which depicts gains in case of ${}^I\tilde{p}_1$, the choice of $q = 0.4$ seems reasonable in both situations. When the demonstration data had missing information, i.e. ${}^I\tilde{p}_3$ was used and so the past and the current DM objectives were entirely different, the gains increased until $q = 0.5$, where they stabilized. When the demonstration data were obtained using the same DM objectives as the current DM objectives, i.e. ${}^I\tilde{p}_1$, so they were informative, the gains decreased slightly with rising value of q.

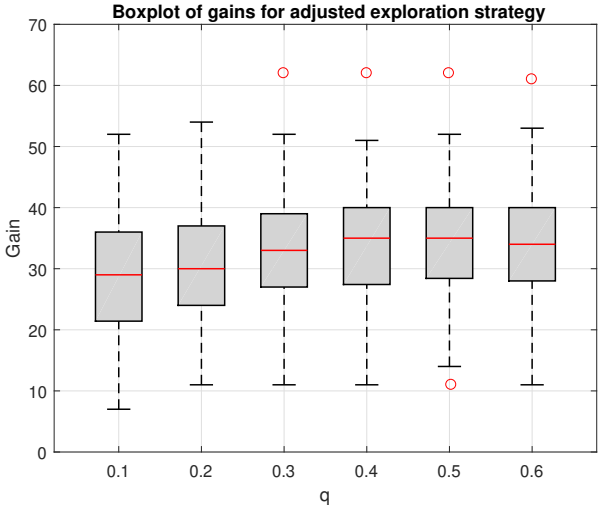


Figure 4.3: Boxplot of gains obtained using the adjusted exploration strategy with changing q, data gathered with ideal model ${}^I\tilde{p}_3$ (the past and the current DM objectives were different).

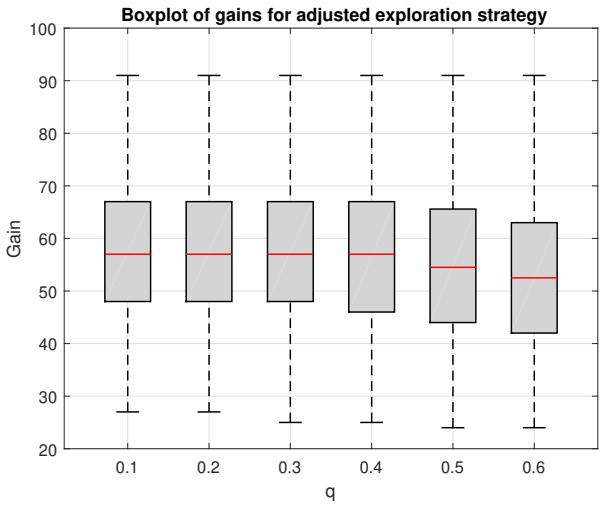


Figure 4.4: Boxplot of gains obtained using the adjusted exploration strategy with changing q, data gathered with ideal model ${}^I\tilde{p}_1$ (the past and the current DM objectives were identical).

Demonstration data with ${}^I\tilde{p}_1$: The last set of experiments presented in this section use demonstration data d_{60} generated with the ideal model ${}^I\tilde{p}_1$ (4.1), which describes DM preferences that are the same as the agent's current DM preferences ${}^Ip$ (4.4). This implies that the data contained sufficient amount of information concerning the goals defined by ${}^Ip$.

Figure 4.6 shows results of the comparison of all considered exploration strategies. The parameters of the adjusted exploration strategy were fixed at $q = 0.4$ and $m = 10$. It can be noted that no exploration, i.e. $\epsilon = 0$, brought the best results while the ϵ-first strategy performed worst. The adjusted exploration strategy came out as the best exploration strategy as it lowered the gains only slightly.

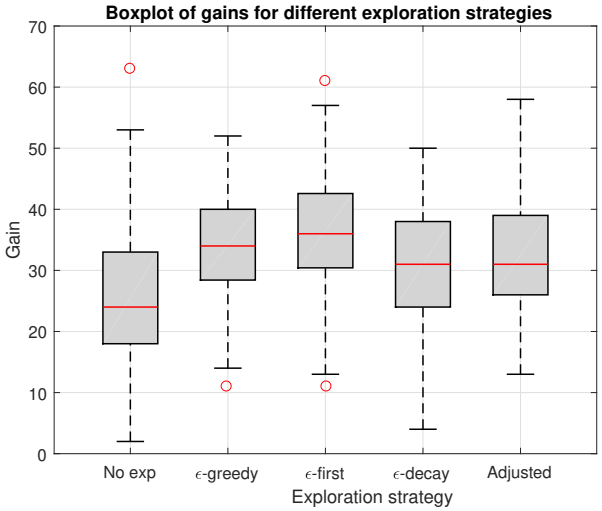


Figure 4.5: Boxplot comparing gains of different exploration strategies, data gathered with ideal model ${}^{I}\tilde{p}_3$ (different from ${}^{I}p$). No exp - no exploration strategy, ϵ-greedy - ϵ-greedy strategy, ϵ-first - ϵ-first strategy, ϵ-decay - ϵ-decreasing strategy, Adjusted - adjusted exploration strategy.

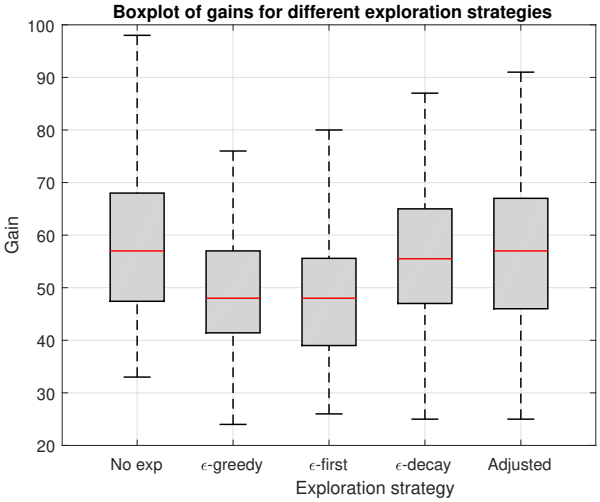


Figure 4.6: Boxplot comparing gains of different exploration strategies, data gathered with ideal model ${}^{I}\tilde{p}_1$ (identical to ${}^{I}p$). No exp - no exploration strategy, ϵ-greedy - ϵ-greedy strategy, ϵ-first - ϵ-first strategy, ϵ-decay - ϵ-decreasing strategy, Adjusted - adjusted exploration strategy.

Comparing boxplots in Figure 4.5 and Figure 4.6 indicates that the adjusted exploration strategy is the most robust exploration strategy with respect to different types of demonstration data. The adjusted exploration strategy is the best choice as it stays active during the whole DM process and is able to adaptively use exploration whenever there is the need for more information. This feature can potentially be of importance for DM tasks with dynamically changing ideal model or system's transition model.

4.1.2 Comparison of the TL method with the FPD method

In this part of the text, results of experiments that compare performance of the FPD method and the proposed TL method are presented. The two methods were also compared to a *random DM policy*, that is a DM 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

$$p(a_t|s_{t-1}) = \frac{1}{|\mathbf{A}|}. \tag{4.5}$$

The DM preferences were focused only on system states.

The FPD method (1.12) 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. The case with complete knowledge of the transition model represents a boundary situation because it is not feasible in real-life applications to fully know the transition model. Learning FPD, i.e. FPD method that learns the unknown transition model in a Bayesian way, was also considered to see a more realistic FPD performance. Bayesian estimation using the same set of observations d_{60} as those available for the TL method was applied to approximate

the transition model. FPD policy (1.12) was optimized over a horizon of $H = 10$ decision epochs in both cases.

The TL method was used either without any exploration (2.8), or with adjusted exploration strategy (3.4). 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 $m = 10$.

To summarize, **the compared methods include**: *i)* the random policy, *ii)* the TL without exploration, *iii)* the TL with the adjusted exploration, *iv)* the FPD without any prior knowledge, *v)* the FPD with full knowledge.

First, Figure 4.7 shows a boxplot representing results of a method comparison where data d_{60} were collected using ideal transition model ${}^{I}\tilde{p}_3$ (4.3), so with completely different objectives than the agent's current DM objectives defined by ${}^{I}p$ (4.4). The boxplot illustrates that when there is no overlap of past and present objectives, the TL performs worse than the random DM policy. When the version of the TL with exploration was adopted, the gains rose slightly above the random DM policy gains, however, they were still considerably worse than the FPD ones.

Second, Figure 4.8 illustrates results of an experiment where ideal model ${}^{I}\tilde{p}_{1,2}$ (4.2) was used while generating the data. The ideal model ${}^{I}\tilde{p}_{1,2}$ expresses DM objectives that partly intersect with the current ones defined by ${}^{I}p$ (4.4). As shown in Figure 4.8, the results improved greatly with observations more appropriate for the agent's aims. The performance of the TL is in general nearly equal to that of the FPD.

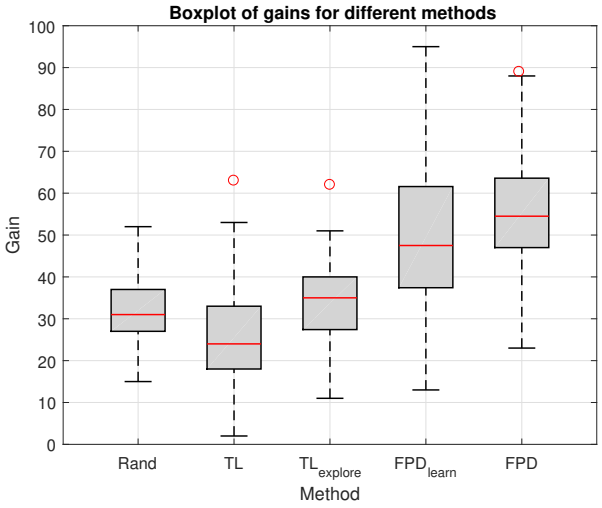


Figure 4.7: Boxplot comparing gains of different methods, data gathered with ideal model ${}^{I}\tilde{p}_3$ (entirely different from ${}^{I}p$). Rand - random policy, TL - TL method, $\text{TL}_{\text{explore}}$ - TL method with exploration, $\text{FPD}_{\text{learn}}$ - learning FPD method, FPD - FPD method with complete knowledge.

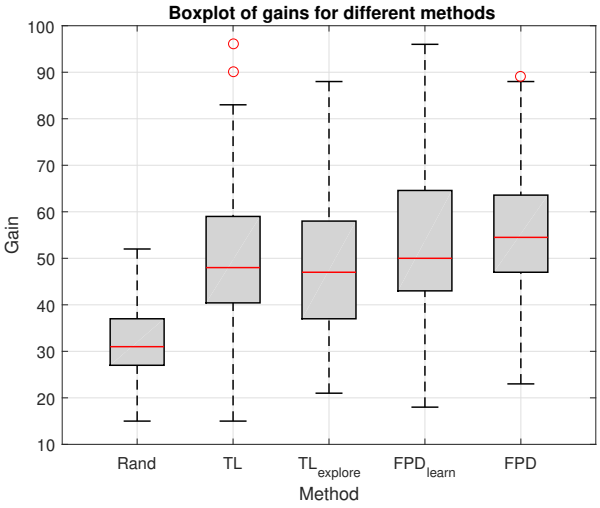


Figure 4.8: Boxplot comparing gains of different methodss, data gathered with ideal model ${}^{I}\tilde{p}_{1,2}$ (partially coincides with ${}^{I}p$). Rand - random policy, TL - TL method, $\text{TL}_{\text{explore}}$ - TL method with exploration, $\text{FPD}_{\text{learn}}$ - learning FPD method, FPD - FPD method with complete knowledge.

Finally, Figure 4.9 represents gains of the compared methods using data d_{60} generated with ${}^{I}\tilde{p}_1$ (4.1), which imitates the case where the past and the current DM objectives are the same. As can be seen in the figure, 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 in this case of past observations being in correspondence with the DM preferences.

Results of the same experiments as in Figures 4.7, 4.8 and 4.9 are shown in Figure 4.10, where gains of the random DM policy (4.5) were subtracted from gains of other methods, i.e. the plot shows benefits

that brought each DM policy in comparison with the random policy. The transition model parameters were different for each simulation so the difficulty of reaching the desired state varied. Figure 4.10 present the results in a more comparative way as it show how much better (worse) than the random DM policy the methods were for each type of the observed data. The results of the FPD method with complete knowledge of the transition model were the same for all three types of demonstration data because the method did not need to use the data to estimate the transition model.

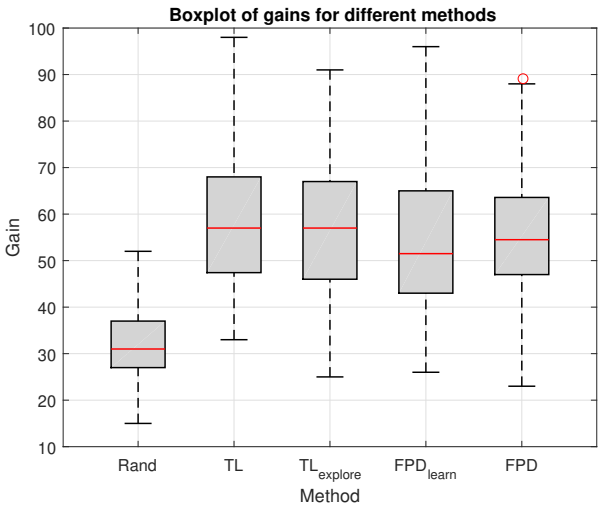


Figure 4.9: Boxplot comparing gains of different methods, data gathered using ideal model ${}^{I}\tilde{p}_1$ (identical to ${}^{I}p$). Rand - random policy, TL - TL method, $\text{TL}_{\text{explore}}$ - TL method with exploration, $\text{FPD}_{\text{learn}}$ - learning FPD method, FPD - FPD method with complete knowledge.

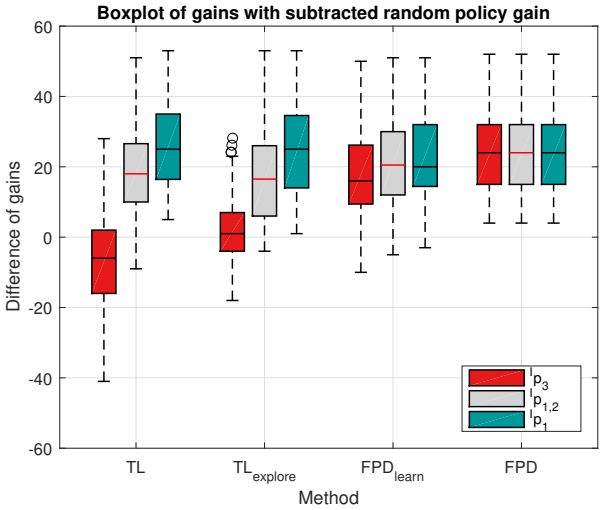


Figure 4.10: Boxplot comparing gains of different methods with gain of random policy subtracted, data gathered with three different ideal models ${}^{I}\tilde{p}$. TL - TL method, $\text{TL}_{\text{explore}}$ - TL method with exploration, $\text{FPD}_{\text{learn}}$ - learning FPD method, FPD - FPD method with complete knowledge.

4.2 Experiments for DM preferences considering both states and actions

Results of simulated experiments where the agent had preferences over states and over actions are presented in this section.

Ideal models for demonstration data: Similarly to Section 4.1, the demonstration data d_{60} were generated using three ideal models, each of them describing different DM preferences.

The first past ideal model ${}^{I}\tilde{p}_1$ favored state s^1 and preferred action a^1 the most and action a^4 the least. The past ideal transition model was defined for all $a_t \in \mathbf{A}$ and all $s_{t-1} \in \mathbf{S}$ as

$$\begin{aligned} {}^{I}\tilde{p}_1(s_t = s^1 | a_t, s_{t-1}) &= 0.99998, \\ {}^{I}\tilde{p}_1(s_t \neq s^1 | a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{4.6}$$

and the past ideal decision rule was defined for all $s_{t-1} \in \mathbf{S}$ as

$$\begin{aligned} {}^{I}\tilde{p}_1(a_t = a^1 | s_{t-1}) &= 0.5, \\ {}^{I}\tilde{p}_1(a_t = a^2 | s_{t-1}) &= 0.245, \\ {}^{I}\tilde{p}_1(a_t = a^3 | s_{t-1}) &= 0.245, \\ {}^{I}\tilde{p}_1(a_t = a^4 | s_{t-1}) &= 0.01. \end{aligned} \tag{4.7}$$

Second past ideal model, symbolized by ${}^{I}\tilde{p}_{1,2}$ and equally favoring states s^1 and s^2 similarly to (4.2), changed the preferences over actions after $\frac{k}{2} = 30$ decision epochs. The ideal transition model did not change and for all $1 \leq t \leq k = 60$, for all $a_t \in \mathbf{A}$, $s_{t-1} \in \mathbf{S}$ it was defined as

$$\begin{aligned} {}^{I}\tilde{p}_{1,2}(s_t = s^1|a_t, s_{t-1}) &= 0.499995, \\ {}^{I}\tilde{p}_{1,2}(s_t = s^2|a_t, s_{t-1}) &= 0.499995, \\ {}^{I}\tilde{p}_{1,2}(s_t = s^3|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{4.8}$$

while the ideal decision rule was for $1 \leq t \leq \frac{k}{2} = 30$ equal to ${}^{I}\tilde{p}_{1,2}(a_t|s_{t-1}) = \frac{1}{|\mathbf{A}|} = 0.25$, for all $a_t \in \mathbf{A}$, $s_{t-1} \in \mathbf{S}$ and for $31 \leq t \leq k = 60$ and for all $s_{t-1} \in \mathbf{S}$ it was

$$\begin{aligned} {}^{I}\tilde{p}_{1,2}(a_t = a^1|s_{t-1}) &= 0.5, \\ {}^{I}\tilde{p}_{1,2}(a_t = a^2|s_{t-1}) &= 0.245, \\ {}^{I}\tilde{p}_{1,2}(a_t = a^3|s_{t-1}) &= 0.245, \\ {}^{I}\tilde{p}_{1,2}(a_t = a^4|s_{t-1}) &= 0.01. \end{aligned} \tag{4.9}$$

The last past ideal model described preference of state s^3 and was identical to ideal model ${}^{I}\tilde{p}_3$ (4.3). The ideal transition model was

$$\begin{aligned} {}^{I}\tilde{p}_3(s_t = s^3|a_t, s_{t-1}) &= 0.99998, \\ {}^{I}\tilde{p}_3(s_t \neq s^3|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{4.10}$$

for all $a_t \in \mathbf{A}$ and for all $s_{t-1} \in \mathbf{S}$. The ideal decision rule was ${}^{I}\tilde{p}_3(a_t|s_{t-1}) = \frac{1}{|\mathbf{A}|} = 0.25$, for all $a_t \in \mathbf{A}$, $s_{t-1} \in \mathbf{S}$ and all $1 \leq t \leq k = 60$.

Current ideal model of the agent: The agent's objectives were identical to the past DM objectives described by the past ideal model ${}^{I}\tilde{p}_1$ (4.6), (4.7), so the agent wanted to reach state s^1 and preferred action a^1 the most and action a^4 the least[1]. The ideal transition model was

$$\begin{aligned} {}^{I}p(s_t = s^1|a_t, s_{t-1}) &= 0.99998, \\ {}^{I}p(s_t \neq s^1|a_t, s_{t-1}) &= 0.00001, \end{aligned} \tag{4.11}$$

for all $a_t \in \mathbf{A}$ and all $s_{t-1} \in \mathbf{S}$. The ideal decision rule was

$$\begin{aligned} {}^{I}p(a_t = a^1|s_{t-1}) &= 0.5, \\ {}^{I}p(a_t = a^2|s_{t-1}) &= 0.245, \\ {}^{I}p(a_t = a^3|s_{t-1}) &= 0.245, \\ {}^{I}p(a_t = a^4|s_{t-1}) &= 0.01, \end{aligned} \tag{4.12}$$

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

Agent's success was measured by *overall gain* defined as $\sum_{\tau=k+1}^{k+h} {}^{I}p(a_\tau|s_\tau)\delta(s_\tau, s^1)$. Essentially, the gain is a weighted number of occurrences of state s^1, where values of the ideal decision rule are used as

[1]Practically such a situation happens when a particular action brings some cost, i.e. in Example 1, Section 1.3: when a^1 corresponds to a treatment that is safe and not very expensive, and a^4 corresponds to an expensive treatment that might have serious side effects.

the weights. No prior information was available, concentration parameters of the prior distribution (2.10) were all equal to

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

All experiments were conducted in the same way as in Section 4.1.2. Results of the TL method (with and without exploration strategy) were compared with results of the FPD method (with and without complete knowledge of the system model). When exploration strategy (3.4) was used, the exploration rate was $\epsilon = 0.3$, the threshold of low average similarity was $q = 0.2$, and the number of preceding similarities to be averaged was $m = 10$. FPD decision policy was optimized over a horizon of $H = 10$ decision epochs.

The value of the threshold q was chosen by conducting similar simulations as in Section 4.1.1. Because the ideal model ${}^{I}p$ (4.11), (4.12) is now more complex and values of similarities are generally smaller, the value $q = 0.4$ that was optimal in the case of preferences only over states, see Section 4.1.1, is not the best choice. In Figures 4.11 and 4.12 we can see gains of the TL method using the adjusted exploration strategy with $\epsilon = 0.3$ and changing values of q for data d_{60} collected using ${}^{I}\tilde{p}_3$ and ${}^{I}\tilde{p}_1$ respectively. The value $q = 0.2$ provides good results for both types of d_{60}.

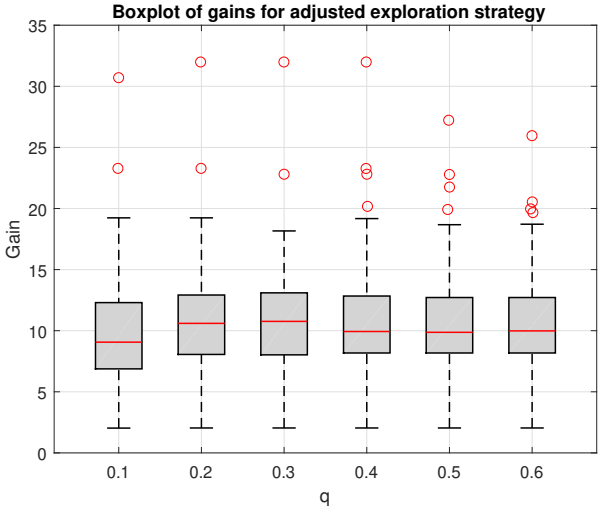


Figure 4.11: Boxplot of gains obtained using the adjusted exploration strategy for different q, data gathered with ideal model ${}^{I}\tilde{p}_3$ (different from ${}^{I}p$).

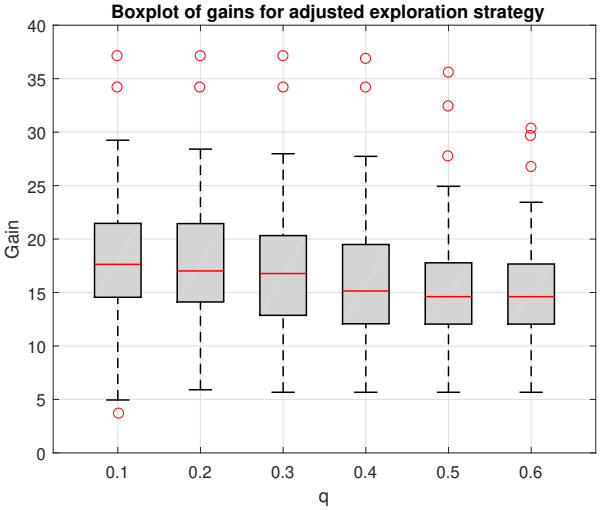


Figure 4.12: Boxplot of gains obtained using the adjusted exploration strategy for different q, data gathered with ideal model ${}^{I}\tilde{p}_1$ (identical to ${}^{I}p$).

Figure 4.13 shows resulting gains of the TL and the FPD method comparison after subtracting gains of the random DM policy (4.5). Comparing Figure 4.13 to Figure 4.10, where only preferences over actions were employed, it can be noted that the results are very similar in terms of how well the methods behave in comparison to each other. This suggests that the TL method maintains the same performance no matter the definition of the ideal model.

Generally, the TL method gains surpassed the FPD method gains in case of data generated using the same ideal model, that is ideal model ${}^{I}\tilde{p}_1$. When the DM preferences were overlapping but not equal (i.e. ${}^{I}\tilde{p}_{1,2}$ was used to generate the data) the results of the TL method were equivalent to the FPD method. The exploration strategy lowered the gains slightly. However, in case the data were generated with a completely different preferences, ${}^{I}\tilde{p}_3$, the gains of the TL method were significantly worse than gains of the FPD method. Exploration strategy helped to overcome the lacking information in the data only to a ceratin extent.

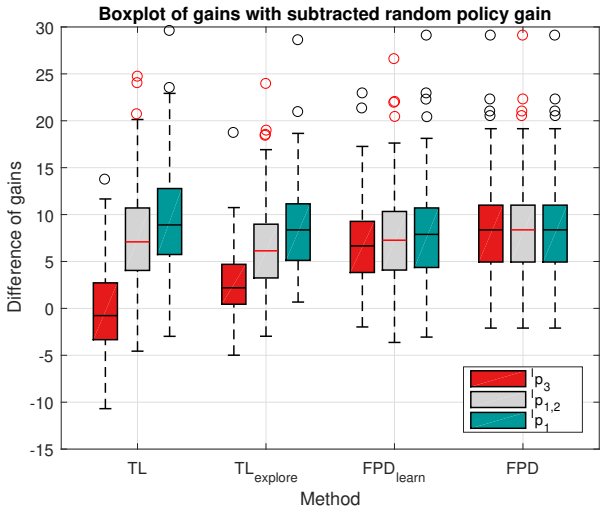


Figure 4.13: Boxplot comparing gains of different methods with gain of random policy subtracted, data gathered with three different ideal models ${}^{I}\tilde{p}$. TL - TL method, $\text{TL}_{\text{explore}}$ - TL method with exploration, $\text{FPD}_{\text{learn}}$ - learning FPD method, FPD - FPD method with complete knowledge.

4.3 Computational complexity

An important aspect of an algorithm is its computational complexity. In order to evaluate the performance of the two considered methods, namely the FPD method and the TL method, the computational complexity of both was analyzed first theoretically and then experimentally.

4.3.1 Theoretical complexity

The complexity of determining one decision rule was estimated using the "big O" notation [4], 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 3.1, 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 (1.12) 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 $O(\max(k, |\mathbf{S}|^2 \cdot |\mathbf{A}|))$.

Looking at Algorithm 3.1 of the TL method, we can differentiate between complexity of computing the first decision rule right after obtaining the sequence of k observations d_k, and every following decision

rule. We can introduce an auxiliary array of a non-normalized estimate of the decision rule, i.e.

$$^{aux}\hat{p}(a_t|s_{t-1}) = \sum_{\tau=1}^{t-1} \omega_\tau \delta(a_t, a_\tau)\delta(s_{t-1}, s_{\tau-1}) + \sum_{s\in\mathbf{S}} \nu_0^{s,a_t|s_{t-1}},$$

which is updated and normalized at each subsequent decision epoch. The update can be written symbolically for $s_t \in \mathbf{S}$ and $a_{t+1} \in \mathbf{A}$ as

$$^{aux}\hat{p}(a_{t+1}|s_t) = {}^{aux}\hat{p}(a_{t+1}|s_t) + \omega_t \delta(a_{t+1}, a_t)\delta(s_t, s_{t-1}).$$

For determining the first ever decision rule, similarities of all past observations are computed and summed. At each of the following decision epochs, similarity ω_t is computed and added to the sum, and $^{aux}\hat{p}(a_{t+1}|s_t)$ is normalized to obtain the decision rule. This update has the asymptotic complexity of $O(|\mathbf{S}|)$ caused by the normalization.

The same can be done for the FPD learning method, where an auxiliary array stores a non-normalized estimate of the transition model. However, the optimal decision rule has to be computed again at each decision epoch. The asymptotic complexity of the update and the calculation of a new optimal decision rule is $O(|\mathbf{S}|^2 \cdot |\mathbf{A}|)$. This suggests that the update of the decision epoch is faster using the TL method than using the learning FPD method.

4.3.2 True complexity

In practice, the coefficients and constants omitted in the theoretical analysis 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. Several experiments were 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 calls a specified function several times and returns the median of the measured elapsed times. In our case, it was used on a function that computes the decision rule using the FPD or the TL method.

The *timeit* function determines the number of repetitions of the specified code automatically to take into account that calling it the first few times is typically more time demanding. 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.

Figure 4.14 shows the median elapsed time of computing the first decision rule after obtaining the data d_k with fixed number of states $|\mathbf{S}| = 3$, fixed number of actions $|\mathbf{A}| = 4$ and for changing number of observations k. It is apparent that in this case of k being much larger than $|\mathbf{S}|$, the elapsed time for both methods shows linear dependence on k only. The sudden jump at the point $k = 12500$ for the FPD method can be explained by the fact that the actual CPU time is considered and it might include some other activity of the system such as memory allocation. Despite this, the linear trend is visible. As mentioned at the beginning of this section, the asymptotic complexity of both methods is the same: $O(\max(k, |\mathbf{S}|^2 \cdot |\mathbf{A}|))$. The real elapsed time is nearly the same for the FPD and the TL methods. It is slightly greater for the FPD method, which is presumably caused by the coefficient H neglected in the computation of the asymptotic complexity.

In Figure 4.15, 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.

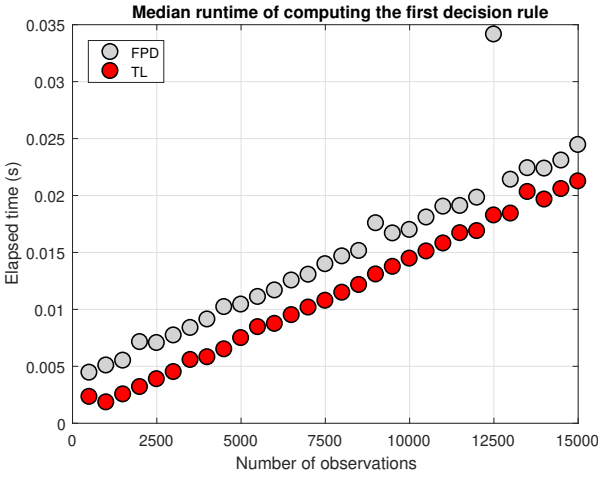


Figure 4.14: Median CPU time required to determine the first decision rule after obtaining the data d_k for growing number of observations k using the FPD learning and the TL method with exploration.

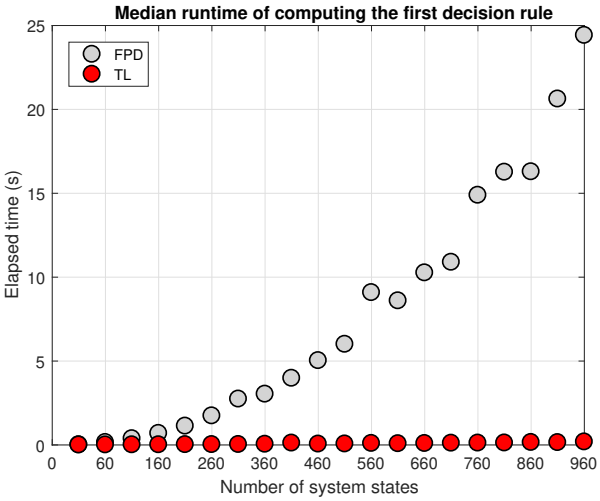


Figure 4.15: Median CPU time required to determine the first decision rule after obtaining the data d_k for growing number of states using the FPD learning and the TL method with exploration.

A graph showing the elapsed time of determining the update of a decision rule with growing number of system states for the TL and the FPD method looks very similar to graph in Figure 4.15, so we show the results for the TL and the FPD method separately. Figure 4.16 presents the median time of estimating the initial decision rule and updating the subsequent decision rule for the FPD method. There is very little difference between the two results. In general, both computations take the same amount of time.

Figure 4.17 shows the median elapsed time of calculating the first decision rule and its update using the TL method. It can be noted that calculating the update is faster, as expected given the difference in the theoretical asymptotic complexity introduced in Section 4.3.1. Comparing Figures 4.16 and 4.17, we can see that the TL method is faster that the FPD method in both computations.

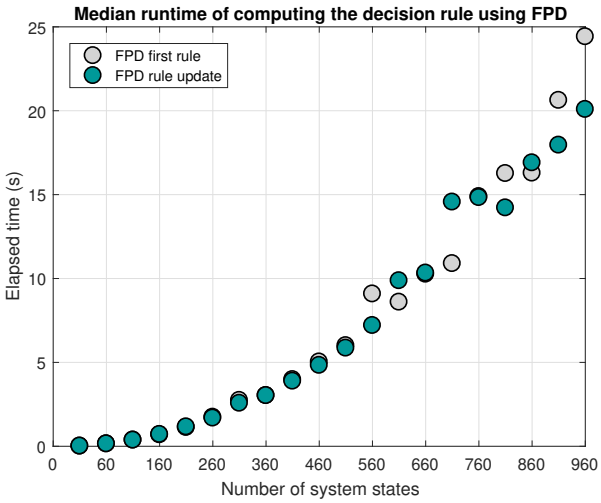


Figure 4.16: Median CPU time required to determine the first decision rule after obtaining the data d_k, and its update for growing number of states using the TL method with exploration.

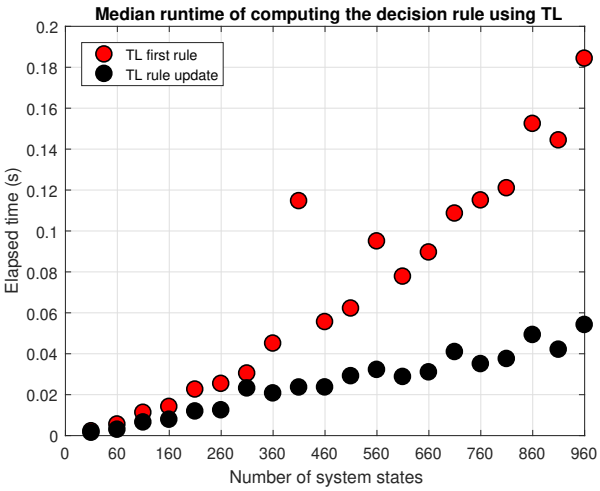


Figure 4.17: Median CPU time required to determine the first decision rule after obtaining the data d_k, and its update for growing number of states using the FPD learning method.

Conclusion

This thesis studied decision-making (DM) under uncertainty within the framework of the fully probabilistic design (FPD), see Section 1.3 for definition, using Markov decision process terms, see Section 1.2. The attention was focused on designing a technique of constructing a decision policy of an agent with predefined DM preferences. The use of the FPD can be limiting due to high computational complexity of solving a higher-dimensional problem. The aim of this thesis was to propose a less-computationally demanding approach using the advantages of the FPD and exploiting data about past DM regarding the same system but with unknown and generally different DM objective. Sequence of system state transitions and actions applied to the system were available to the agent in advance and thus the best experience from the past could be exploited in the current DM task. To exploit the best practise, a transfer learning of an optimal decision policy was applied. The proposed transfer learning defines a degree of similarity of past observations to current objectives (Section 2.1), and uses Bayesian learning to estimate the unknown optimal decision policy (Section 2.2). Different kinds of exploration strategy were introduced in order to overcome a possible lack of information in the past data (demonstration data).

The proposed technique was verified through a series of simulated experiments, see Chapter 4. The results of the experiments show that the performance of the proposed method is comparable to (or it even outperforms) the FPD method in case the demonstration data contain sufficient amount of information about the system. That is when the unknown past preferences are not entirely different from the agent's current objectives. On the other hand, when there is no overlap of DM preferences, the method performs worse than the FPD. Explorative strategy helps to overcome this problem to some extent. A simple complexity analysis indicates that the proposed technique is faster than the FPD solution (Section 4.3).

The main limitation of the similarity-based transfer learning technique is the inability to sufficiently overcome the absence of relevant data. Further research on the subject should investigate new exploration strategies that would faster improve the performance in case of lack of information. Another possible modification is adding a method of avoiding a negative transfer into the learning of the optimal decision policy, that is assessing the available data before the transfer learning is applied and using only the part of the data suitable for the current DM task. A challenge is to keep the computational complexity low while improving the transfer learning technique by adding new features to it.

The method of similarity-based transfer learning could also be expanded to incorporate learning of the unknown past DM preferences. This is a topic studied in inverse reinforcement learning, see [2], [48]. It could help improve the performance in case the current DM preferences are not feasible or are poorly defined by the user.

Bibliography

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

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

[3] C. Agostinelli and L. Greco. Weighted likelihood in Bayesian inference. In *Proceedings of the 46th Scientific Meeting of the Italian Statistical Society*, pages 746–757, 2012.

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

[5] O. Alagoz, H. Hsu, A.J. Schaefer, and M.S. Roberts. Markov decision processes: a tool for sequential decision making under uncertainty. *Medical Decision Making*, 30(4):474–483, 2010.

[6] S.A.M. Almasani, V.I. Finaev, W.A.A. Qaid, and A.V. Tychinsky. The decision-making model for the stock market under uncertainty. *International Journal of Electrical & Computer Engineering (2088-8708)*, 7(5), 2017.

[7] H.B. Ammar, E. Eaton, M.E. Taylor, D.C. Mocanu, K. Driessens, G. Weiss, and K. Tuyls. An automated measure of mdp similarity for transfer in reinforcement learning. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[8] B.D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[9] P. De Beaucorps, T. Streubel, A. Verroust-Blondet, F. Nashashibi, B. Bradai, and P. Resende. Decision-making for automated vehicles at intersections adapting human-like behavior. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 212–217. IEEE, 2017.

[10] F. Le Ber, X. Dolques, L. Martin, A. Mille, and M. Benoît. A reasoning model based on perennial crop allocation cases and rules. In *International Conference on Case-Based Reasoning*, pages 61–75. Springer, 2017.

[11] L. Berec and M. Kárný. Identification of reality in Bayesian context. In M. Kárný and K. Warwick, editors, *Computer Intensive Methods in Control and Signal Processing*, pages 181–193. Birkhäuser, Boston, MA, 1997.

[12] D. Bouneffouf, A. Bouzeghoub, and A.L. Gançarski. Exploration/exploitation trade-off in mobile context-aware recommender systems. In *Australasian Joint Conference on Artificial Intelligence*, pages 591–601. Springer, 2012.

[13] C.T. Chen, A.P. Chen, and S.H. Huang. Cloning strategies from trading records using agent-based reinforcement learning algorithm. In *2018 IEEE International Conference on Agents (ICA)*, pages 34–37. IEEE, 2018.

[14] D.P. de Farias and N. Megiddo. Exploration-exploitation tradeoffs for experts algorithms in reactive environments. In *Advances in neural information processing systems*, pages 409–416, 2005.

[15] T.S. Ferguson. Prior distributions on spaces of probability measures. *The Annals of Statistics*, 2(4):615–629, 1974.

[16] M. H. De Groot. *Optimal Statistical Decisions*. McGraw Hill, New York, 1970.

[17] J. Hoey, T. Schröder, and A. Alhothali. Affect control processes: Intelligent affective interaction using a partially observable markov decision process. *Artificial Intelligence*, 230:134–172, 2016.

[18] Z. Hou, S. Liu, and T. Tian. Lazy-learning-based data-driven model-free adaptive predictive control for a class of discrete-time nonlinear systems. *IEEE transactions on neural networks and learning systems*, 28(8):1914–1928, 2016.

[19] C. Hummersone. Alternative box plot (`https://www.github.com/IoSR-Surrey/MatlabToolbox`). Retrieved March 1, 2020.

[20] M. Kárný and T.V. Guy. Fully probabilistic control design. *Systems & Control Letters*, 55(4):259–265, 2006.

[21] M. Kárný and T. Kroupa. Axiomatisation of fully probabilistic design. *Information Sciences*, 186(1):105–113, 2012.

[22] D. Kasenberg and M. Scheutz. Interpretable apprenticeship learning with temporal logic specifications. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 4914–4921. IEEE, 2017.

[23] I. Kostrikov, K.K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

[24] M. Kárný. Towards fully probabilistic control design. *Automatica*, 32(12):1719–1722, 1996.

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

[26] P. Kulkarni. *Reinforcement and systemic machine learning for decision making*, volume 1. John Wiley & Sons, 2012.

[27] A. Lazaric, M. Restelli, and A. Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 544–551, 2008.

[28] G. Liu, T. Jiang, T.B. Ollis, X. Zhang, and K. Tomsovic. Distributed energy management for community microgrids considering network operational constraints and building thermal dynamics. *Applied energy*, 239:83–95, 2019.

[29] T. Osa, J. Pajarinen, G. Neumann, J.A. Bagnell, P. Abbeel, and J. Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018.

[30] V. Peterka. Bayesian approach to system identification. In P. Eykhoff, editor, *Trends and Progress in System Identification*, pages 239–304. Pergamon Press, Oxford, 1981.

[31] G. Phillips-Wren. Intelligent systems to support human decision making. In *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*, pages 3023–3036. IGI Global, 2017.

[32] G. Phillips-Wren, N. Ichalkaranje, and L.C. Jain. *Intelligent decision making: An AI-based approach*, volume 97. Springer Science & Business Media, 2008.

[33] B. Price and C. Boutilier. Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research*, 19:569–629, 2003.

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

[35] K. Raharjo. *Applying Multi-Armed Bandit on Game Development*. PhD thesis, University of British Columbia, Okanagan, 2016.

[36] M. Ravuri, A. Kannan, G.J. Tso, and X. Amatriain. Learning from the experts: From expert systems to machine-learned diagnosis models. *Machine Learning for Health Care*, 2018.

[37] V. Raykar and P. Agrawal. Sequential crowdsourced labeling as an epsilon-greedy exploration in a markov decision process. In *Artificial intelligence and statistics*, pages 832–840, 2014.

[38] J.A. Recio-Garcia, B. Díaz-Agudo, J.L. Jorro-Aragoneses, and A. Kazemi. Intelligent control system for back pain therapy. In *International Conference on Case-Based Reasoning*, pages 287–301. Springer, 2017.

[39] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[40] Y. Shaposhnik. *Exploration vs. Exploitation: reducing uncertainty in operational problems*. PhD thesis, Massachusetts Institute of Technology, 2016.

[41] J. Štěch, T.V. Guy, B. Pálková, and M. Kárný. Lazy learning of environment model from the past, 2015.

[42] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, 1998.

[43] M.E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

[44] L. Torrey and J. Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global, 2010.

[45] H. Valizadegan, R. Jin, and S. Wang. Learning to trade off between exploration and exploitation in multiclass bandit prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 204–212, 2011.

[46] C.J.C.H. Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, May 1989.

[47] 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*, volume 97 of *Proceedings of Machine Learning Research*, pages 6818–6827, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[48] L. Yu, T. Yu, C. Finn, and S. Ermon. Meta-inverse reinforcement learning with probabilistic context variables. In *Advances in Neural Information Processing Systems 32*, pages 11772–11783. Curran Associates, Inc., 2019.