Simultaneous Estimation of Rewards and Dynamics in Inverse Reinforcement Learning Problems

Michael Herman

Technische Fakultät
Albert-Ludwigs-Universität Freiburg

Dissertation zur Erlangung des akademischen Grades
Doktor-Ingenieur

Betreuer: Prof. Dr. Wolfram Burgard

Simultaneous Estimation of Rewards and Dynamics in Inverse Reinforcement Learning Problems

Michael Herman

Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur
Technische Fakultät, Albert-Ludwigs-Universität Freiburg

Dekan	Prof. Dr. Rolf Backofen
Erstgutachter	Prof. Dr. Wolfram Burgard
Zweitgutachter	Jun.-Prof. Dr. Joschka Bödecker
Tag der Disputation	17. Januar 2020

Zusammenfassung

Da sich die Fähigkeiten autonomer Systemen stetig verbessern, können sie in zunehmend komplexeren Umgebungen immer vielseitigere Aufgaben lösen. Häufig ist es dabei nötig das autonome System an die spezifische Aufgabe oder die jeweilige Umwelt anzupassen, was typischerweise eine umfangreiche Forschung und Entwicklung voraussetzt. Um die Akzeptanz solcher Systeme zu erhöhen, ist es erforderlich deren Einsatz für verschiedene Aufgaben schnell und einfach anhand von Anpassungen der Verhaltensweisen und Ziele durch Nicht-Experten zu ermöglichen. Eine intuitive Art und Weise Aufgaben zu beschreiben ist das Bereitstellen von Demonstrationen erwünschten Verhaltens. Diese Demonstrationen können verwendet werden, um eine Repräsentation der Motivation und des Ziels des Experten zu lernen.

Das Lernen aus Demonstrationen beschreibt eine Klasse von Ansätzen, anhand derer neue Verhaltensweisen trainiert werden können, indem Funktionen und Aufgaben vorgeführt werden, anstatt diese zu programmieren. Zwei Teilbereiche des Lernens aus Demonstrationen sind das Klonen von Verhalten und inverses bestärkendes Lernen. Ansätze aus dem Bereich des Klonens von Verhalten schätzen die Strategie des Experten aus dessen Demonstrationen und lernen somit diesen zu imitieren. Allerdings sind die erlernten Strategien nur geeignet, wenn sich die Umwelt, ihre Dynamik sowie die Aufgabe nicht ändern. Ein populärer Ansatz, der generalisierbarere Repräsentationen lernt, ist das inverse bestärkende Lernen beziehungsweise Inverse Reinforcement Learning (IRL). Dabei wird die Belohnungsfunktion eines Markow-Entscheidungsprozesses aus Demonstrationen eines Experten geschätzt, wobei die Belohnungsfunktion als Motivation oder Ziel interpretiert werden kann. Es existiert eine Vielzahl an Ansätzen des inversen bestärkenden Lernens, die das Problem unter unterschiedlichen Annahmen lösen. Die meisten dieser Ansätze nehmen an, dass ein akkurates Dynamikmodell vorhanden ist, dass das Modell aus Expertendemonstrationen geschätzt werden kann, dass zusätzliche Demonstrationen suboptimalen Verhaltens abgefragt werden können oder dass Heuristiken verwendet werden, um ein nicht vorhandenes Transitionsmodel zu kompensieren. Allerdings werden viele dieser Annahmen häufig verletzt, weil das Dynamikmodel einer

Umwelt sehr komplex sein kann, weil akkurate Modelle häufig nicht vorhanden sind, weil zusätzliche Demonstrationen zu teuer sein können und weil Heuristiken die Schätzung der Belohnungsfunktion verzerren können.

Um Probleme des inversen bestärkenden Lernens unter unbekannten Dynamikmodellen zu lösen, stellen wir einen Ansatz vor, der simultan die Belohnungsfunktion und das Dynamikmodell aus Expertendemonstrationen schätzt. Dies ist möglich, da sowohl die Belohnungsfunktion als auch das Transitionsmodell die Strategie des Experten beeinflussen und daher beide aus Expertendemonstrationen inferiert werden können. Demzufolge enthalten nicht nur die beobachteten Transitionen sondern auch die beobachteten Aktionen Informationen über das Transitionsmodell der Umwelt. Mit dieser Arbeit wird eine neue Problemklasse einer simultanen Schätzung der Belohnungsfunktion und der Dynamik eingeführt. Zudem werden mehrere Lösungen des Problems hergeleitet, die unterschiedliche Annahmen an die Generierung der Strategie des Experten stellen. Die vorgestellten Ansätze werden anhand eines Anschauungsbeispiels, der Navigation auf Basis eines Satellitenbildes, sowie der Navigation eines simulierten Roboters in einem Gangszenario mit Menschen evaluiert. Hierbei zeigen die Ergebnisse, dass das Miteinbeziehen der Schätzung des Transitionsmodells in das inverse bestärkende Lernen zu exakteren Modellen der Dynamik und der Belohnungsfunktion führen.

Abstract

As the capabilities of autonomous systems improve, they can solve more and more tasks in increasingly complex environments. Often, the autonomous system needs adjustments to the specific task or environment, which typically requires extensive research and engineering. By allowing non-experts to adjust the systems to new behaviors and goals, they are faster and easier to deploy for various tasks and environments, which increases their acceptability. An intuitive way to describe a task is to provide demonstrations of desired behavior. These demonstrations can be used to learn a representation of the expert's motivation and goal.

Learning from Demonstration is a class of approaches offering the possibility to teach new behaviors by demonstrating the task instead of programming it directly. Two subfields of Learning from Demonstration are Behavioral Cloning and Inverse Reinforcement Learning (IRL). Approaches from Behavioral Cloning estimate the expert's policy directly from demonstrations and therefore learn to mimic the expert. However, the learned policies are typically only appropriate if the environment, the dynamics, and the task remain unchanged. A popular approach that learns more generalizable representations is Inverse Reinforcement Learning, which estimates the unknown reward function of a Markov Decision Process (MDP) from demonstrations of an expert. Many Inverse Reinforcement Learning approaches exist that solve the problem under various assumptions. Most of them assume the environment's dynamics to be known, that they can be learned from expert demonstrations, that additional samples from suboptimal behavior can be queried, or that appropriate heuristics account for unknown transition models. However, these assumptions are often not satisfied, since transition models of environments can be complex, accurate models may be unknown, querying samples may be too expensive, and heuristics may tamper with the reward estimate.

To solve IRL problems under unknown dynamics, we propose a framework that simultaneously estimates both the reward function and the dynamics from expert demonstrations. This is possible, as both influence the expert's policy and thus the long-term behavior. Therefore, not only the observed transitions of the expert's demonstrations but also the observed actions contain information about the dynamics of the environment. The contribution of this thesis is the formulation of a new problem class, called Simultaneous Estimation of Rewards and Dynamics (SERD). Furthermore, we derive several solutions to this problem under different assumptions on how experts compute their policy. The evaluation on a minimum example, a grid world navigation task, and a high-level navigation task of a simulated robot in a populated hallway shows that incorporating the estimation of the transition model into Inverse Reinforcement Learning yields more accurate models of the dynamics and the reward function.

Acknowledgments

This thesis would have not been possible without the support, guidance, and inspiration of supervisors, colleagues, friends, and family. I am thankful for all the people who made this thesis possible, who gave me the freedom to follow own ideas, and who made it an enjoyable time. First, I would like to thank my doctoral advisor, Wolfram Burgard, for his supervision, his insights and ideas as well as his feedback on research aspects, scientific writing, and presenting scientific results. While being an external PhD student, he has fostered close contact and collaborations with the Autonomous Intelligent Systems lab and made me feel welcome there. I thank Joschka Bödecker, Thomas Brox and Bernhard Nebel for agreeing to be my examination committee. In addition, I would like to thank my supervisors at Bosch, Sheung Ying Yuen-Wille, Volker Fischer, and Tobias Gindele, for their support, constant feedback, and pushing me towards theoretical topics. I am deeply grateful to my supervisor Tobias Gindele, who was asking the right questions, motivating me to target hard ones, being patient, and a friend. Furthermore, I thank Roland Klinnert and Dr. Yasser Jadidi without whom I would not have started a PhD at Bosch.

In addition, I would like to give special thanks to my collaborators at the BCAI, Jörg Wagner, Felix Schmitt, Jens Schreiter, Julia Vinogradska, and Markus Spies, for plenty of fruitful discussion and joint work. Furthermore, I had the chance to supervise several student projects in the area of my PhD topic. Nikolaus Mitchell, Lucas Lutz, Benjamin Coors, Christopher Quignon, David Wagner, Eric Schmidt, Lucas Schweitzer, it was a pleasure to work together with you. Besides working on the PhD topic, I have also enjoyed other activities and events such as being part of the Christmas bands of the AIS and the BCAI. Thank you all for making this time enjoyable.

Finally, I want to thank my family and friends for their support and encouragement, even though that I sometimes had to prioritize the PhD and paper deadlines higher than other social activities. I am grateful to my parents, who have sparked my interest for computer science and electronics and taught me to always give the best. Especially, I would like to thank Maja for always supporting me, giving me enough freedom, encouraging me to finalize my tasks, and teaching me the right balance between work and spare time.

Contents

Chapter 1

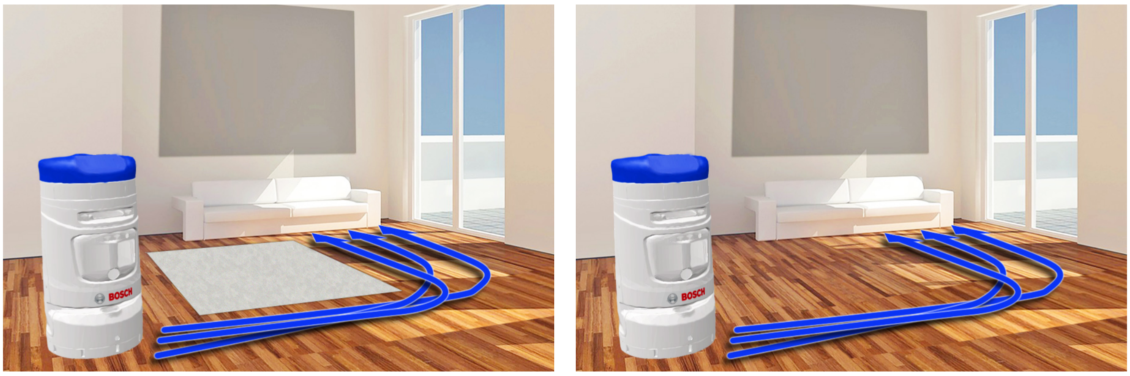
Introduction

The capabilities of autonomous systems are constantly improving, which allows deploying them in more and more complex environments for an increasing number of tasks. Early examples of autonomous robots that operate in populated areas are the museum tour-guide robots MINERVA [100] and RHINO [20] as well as the conference-attending robot GRACE [94]. Recently, the robot Obelix [60] has been successfully navigating autonomously in urban, populated areas and cars are improving in autonomous driving [67]. Besides these problems, robots improve in performing manipulation tasks and are able to tie knots [91], solve pouring tasks [113], grasp various objects [66], and fold towels [71]. A recent work by Levine *et al.*[65] has solved manipulation tasks solely based on images.

Even though these examples show that the abilities of autonomous systems improve, end customers and the industry mainly apply them to highly specific tasks. For example, industrial robots are used for repetitive tasks in assembly lines of manufacturing plants, while domestic robots are typically used for vacuum cleaning or mowing the lawn. The reason for this is that extensive research and development is necessary to enable them solving these tasks. In order to use autonomous systems for more versatile tasks and thereby utilize all their capabilities, it is necessary to provide simple and fast mechanisms to adapt them to new tasks, environments, and behaviors.

An intuitive way to provide a description of a task are demonstrations of the desired behavior. In contrast to directly programming the system behavior, non-experts are often able to demonstrate optimal or close to optimal behavior. Then, these demonstrations can be used to learn a representation of the goal. However, interpreting demonstrations can be difficult, if the environment and its dynamics are (partially) unknown. This is often the case, as dynamics might be complex and environments can change.

Assume that an end consumer buys a mobile robot, that is supposed to solve transporta-



(a) Demonstrations

(b) Wrong transition model

Figure 1.1: Example of misinterpretations of observed behavior due to inaccurate models: Imagine deploying a robot in an unknown environment. (a) A human provides demonstrations of moving to the couch while omitting the carpet, as the human knows that moving over the carpet is slower. If the robot knows that the dynamics differ on the carpet, it can correctly interpret the desired behavior to move fast to the couch. (b) Typically, the environment's dynamics are unknown, if a robot is deployed in a new environment. Since it did not observe any demonstration on the carpet, it cannot naively estimate the dynamics from transitions. If it assumes that the dynamics are equal everywhere, it will wrongly interpret the demonstrations by assuming that it is desirable to walk close to the wall or the windows.

tion tasks in a domestic environment, such as illustrated in Fig. 1.1. After the robot's deployment in the household, the user should program it to transport items to the couch in the back. Since this domestic environment is new to the robot, it can hardly make assumptions about the environment and its dynamics. However, the user knows that the carpet in the middle of the room is reducing the velocity of the robot. Therefore, he provides several demonstrations of navigating on a longer path around the carpet, which results in a faster execution of the transportation task (see Fig. 1.1 (a)). The robot should use these demonstrations in order to learn solving the desired task. If the robot has never observed any demonstration on the carpet, it is not able to estimate the dynamics on the carpet by traditional naive approaches such as supervised learning based ones. However, assumptions about these dynamics influence the interpretation of the expert's demonstrations. If the robot assumes that the dynamics are identical everywhere, it will wrongly interpret the demonstrations by assuming that it is desired to move close to walls or windows (depicted in Fig. 1.1 (b)). It can only find the correct interpretation, if an appropriate model of the environment's dynamics is available (depicted in Fig. 1.1 (a)). This example illustrates the influence of inaccurate models of the dynamics on the interpretability of observed behavior in terms of motivations and goals.

The goal of this thesis is to contribute to the field of learning from demonstrations by examining the influence of (partially) unknown dynamics, uncovering the limitations of state-of-the-art approaches, and proposing solutions to learn from demonstrations if transition models are unknown. We present a new problem class based on Inverse Reinforcement Learning, which extends it by simultaneously estimating a model of the environment's dynamics. By considering that the expert's policy is a result of his knowledge about the environment's dynamics as well as his motivations and goals, we enable the robot to improve transition models from observed behavior of an expert. This even allows drawing conclusions about parts of the transition model that were never observed. An evaluation on a minimum example and a navigation task on discretized satellite images shows improved performance over traditional approaches, by better explaining the expert's behavior and more accurately modeling the dynamics of the environment. Based on a realistic learning scenario of high-level navigation strategies in a simulated, crowded environment we further show that the proposed approach generalizes well from human demonstrations and that the resulting models allow predicting real human behavior more accurately.

1.1 Problem Statement

In many real world applications, accurate models of the environment's dynamics are not available. Nevertheless, methods are necessary for adjusting autonomous systems to new tasks and behaviors. Therefore, one is interested in learning the expert's motivation as well as the environment's dynamics solely from expert demonstrations. We assume full observability of the behavior, which causes the real state and action to be measured without noise. However, the expert's goal and the transition model of the environment may be partially or fully unknown. Furthermore, humans are sub-optimal decision makers, which often yields noisy demonstrations. Hence, it is required to account for these uncertainties, by learning behavioral models that are stochastic.

Learning the expert's motivation and the environment's dynamics allows to make predictions about the expert's decision making. Therefore, they allow for inferring future behavior as well as statistics on time requirements of solving a task. Often, the estimated dynamics and reward functions should transfer to differing initial situations or changing environments. For this reason, generalizable results are important, such that the resulting optimal strategies are able to predict the expert's behavior accurately even in unobserved states.

Therefore, this thesis tries to answer the following questions:

- What are the goals and the motivations of the expert solving a task?
- What are the dynamics of an environment?
- What are the limitations of current Inverse Reinforcement Learning approaches that solve these problems?
- How can we learn motivations, goals, and environment's dynamics from expert demonstrations, if the dynamics of the environment are fully or partially unknown and only expert demonstrations are available?
- Is it possible to estimate the dynamics of an environment from expert demonstrations even for unobserved states and actions?
- How to use additional knowledge for achieving generalizable results that are even applicable in new environments?

We answer these questions by examining current approaches, by revealing their drawbacks, by introducing a new problem class that formalizes these questions, and by deriving a new theoretically founded approach.

1.2 Thesis Statement

The thesis statement is:

> *"It is possible to estimate an expert's motivation and goals as well as the dynamics of the environment solely from expert demonstrations and the estimates can be used to make accurate predictions of future behavior."*

To support this statement, we present a new problem class together with approaches that solve it by learning from goal-directed expert demonstrations. In addition, we provide evaluations, showing that both simulated and real human behavior can be explained by the learned models.

1.3 Concept Overview

Throughout this thesis, we develop a learning algorithm for estimating the motivations and goals of experts as well as the dynamics of the environment solely from expert demonstrations. We build on Markov Decision Processes (Sec. 2.4), which are models for sequential decision making that are parameterized by a reward function, specifying the desirability of states and actions. Therefore, estimating the expert's reward function corresponds to estimating his motivation, which is commonly known as Inverse Reinforcement Learning (Sec. 3). In this thesis, we extend classical IRL by simultaneously estimating the dynamics of the environment.

Fig. 1.2 illustrates an overview of the proposed concept. The bottom level shows the generation of expert demonstrations. Typically, they have some type of prior knowledge about the environment. Therefore, they consider both the environment's dynamics and their motivation, when searching for an optimal strategy to solve a specific problem. Consequently, their policy is a result of both rewards and dynamics. If they apply their policy in a real or simulated environment, demonstrations consist of samples of the expert's policy and samples of the environment's transition model. The second level of Fig. 1.2 indicates the estimation of the unknown parameters of the MDP based on these samples. The problem definition assumes both rewards and dynamics to be unknown beforehand. In contrast to IRL approaches, the approach developed in this thesis considers the bilateral influence of rewards and dynamics on the expert's policy. This allows to learn models of the dynamics that are more accurate then naively estimating them directly from observed transitions. Hence, the proposed method is useful for learning behavioral models of the expert together with the transition model of the environment. Consequently, the approach allows for teaching autonomous systems to behave according to the desired behavior of the expert. Furthermore, the resulting estimates can be used for predicting expert behavior.

This can be advantageous in a variety of applications. For example, it enables to train an autonomous system to solve new tasks in unknown environments solely from expert behavior. By learning succinct representations of the expert's motivation, the resulting estimates are applicable in new or changing environments to compute optimal policies that solve the problem. Furthermore, the approach allows us to learn user models, which enable tracking deviations from typical behavior. For example, a driver assistance system could warn a driver, if the driver's behavior is differing to typical ones.

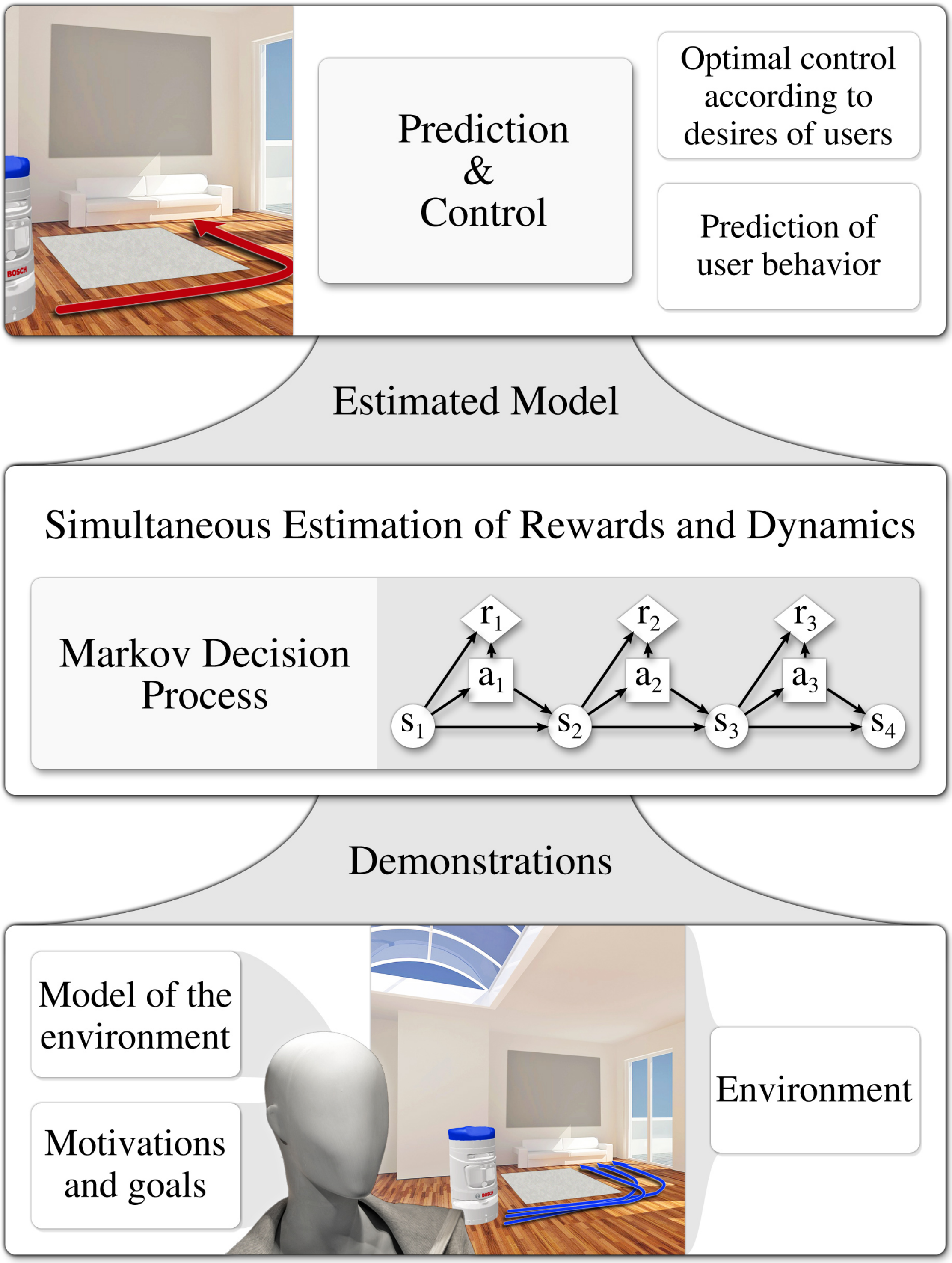


Figure 1.2: Overview of the proposed approach

1.4 Contributions

This thesis contributes to the fields of Learning from Demonstration, Inverse Reinforcement Learning, Machine Learning, and Robotics by proposing a new framework for estimating reward functions and dynamics in Inverse Reinforcement Learning problems if the environment's dynamics are unknown and only expert demonstrations are available. This framework enables to teach robots new behaviors, if the environment is unknown beforehand. This is often the case when end customers deploy a system. In particular the key contributions of this thesis are:

- An examination of state of the art approaches with a discussion about their typical limitations.
- The definition of a new problem class of a simultaneous estimation of rewards and dynamics to overcome those limitations.
- A unified solution to this problem class of which traditional IRL and model estimation in MDPs are special cases.
- The derivations of specific solutions for two different expert policy models with theoretical proofs of the derived approaches.
- Applications of the proposed approaches to the problems of learning to navigate on discretized satellite images and learning high-level navigation strategies in populated environments.

1.5 Publications

This thesis is based on our previous work presented in the following conference proceedings and journals:

- M. Herman, T. Gindele, J. Wagner, F. Schmitt, and W. Burgard. Simultaneous estimation of rewards and dynamics from noisy expert demonstrations. In *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 677–682, April 2016.
- M. Herman, T. Gindele, J. Wagner, F. Schmitt, and W. Burgard. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Artificial Intelligence and Statistics (AISTATS)*, 2016.

- M. Herman, T. Gindele, J. Wagner, F. Schmitt, C. Quignon, and W. Burgard. Learning high-level navigation strategies via inverse reinforcement learning: A comparative analysis. In *Australasian Joint Conference on Artificial Intelligence*, pages 525–534. Springer, 2016.

The following publications that are not included in this thesis also originate from the author's work:

- M. Herman, V. Fischer, T. Gindele, and W. Burgard. Inverse reinforcement learning of behavioral models for online-adapting navigation strategies. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3215–3222. IEEE, 2015.

- J. Wagner, V. Fischer, M. Herman, and S. Behnke. Multispectral pedestrian detection using deep fusion convolutional neural networks. In *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 509–514, April 2016.

- F. Schmitt, H.-J. Bieg, D. Manstetten, M. Herman, and R. Stiefelhagen. Predicting lane keeping behavior of visually distracted drivers using inverse suboptimal control. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, June 2016.

- F. Schmitt, H.-J. Bieg, D. Manstetten, M. Herman, and R. Stiefelhagen. Exact maximum entropy inverse optimal control for modelling human attention switching and control. In *Proc. of the IEEE Conference on Systems, Man and Cybernetics (SMC)*, October 2016.

- F. Schmitt, H.-J. Bieg, M. Herman, and C. Rothkopf. I see what you see: Inferring sensor and policy models of human real-world motor behavior. In *AAAI Conference on Artificial Intelligence*, 2017.

- J. Wagner, V. Fischer, M. Herman, and S. Behnke. Learning semantic prediction using pretrained deep feedforward networks. In *25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, April 2017.

- J. Wagner, V. Fischer, M. Herman, and S. Behnke. Functionally modular and interpretable temporal filtering for robust segmentation. In *29th British Machine Vision Conference (BMVC)*, September 2018.

- U. Baumann, C. Gläser, M. Herman, and J. M. Zöllner. Predicting ego-vehicle paths from environmental observations with a deep neural network. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, 2018.
- U. Baumann, Y.-Y. Huang, C. Gläser, M. Herman, H. Banzhaf, and J. M. Zöllner. Classifying road intersections using transfer-learning on a deep neural network. In *Proc. of the IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

1.6 Outline

This thesis is structured as follows: In Chap. 2, the basic mathematical notation and fundamentals are introduced. Chap. 3 gives an overview about related work and state of the art approaches in the field of Learning from Demonstration and Inverse Reinforcement Learning. The following Chap. 4 examines limitations of state-of-the-art approaches based on a minimum example and introduces the new problem class of a simultaneous estimation of rewards and dynamics. Furthermore, it derives a unified solution based on a maximum a-posteriori formulation of the problem, discusses various priors, and evaluates them on two examples to highlight their differences. Subsequently, Chap. 5 analyzes the proposed approach by applying it to the problem of learning high-level navigation strategies in a densely populated hallway from real human demonstrations. Finally, Chap. 6 concludes the thesis by discussing the proposed approaches and mentioning fields for future research.

Chapter 2

Preliminaries

This thesis proposes an approach for learning behavioral models from demonstrations. Noisy measurements and the stochastic nature of human behavior necessitate the use of probabilistic models, while measures from information theory allow for evaluating random variables. Therefore, this chapter introduces the mathematical notation together with concepts from probability theory [14, 15], information theory [70, 92], and a model for sequential decision making [13, 80].

2.1 Basic Mathematical Notation

Throughout this thesis, we use the following mathematical terminology unless otherwise stated. If functions are defined, the notation indicates their type of output.

Notation	Meaning
x, y	Scalar values
$\boldsymbol{x}, \boldsymbol{y}$	Vectors
$\boldsymbol{x}^\intercal, \boldsymbol{y}^\intercal$	Transposed vectors
$\boldsymbol{X}, \boldsymbol{Y}$	Matrices
$\boldsymbol{X}^\intercal, \boldsymbol{Y}^\intercal$	Transposed matrices
$\mathcal{X}, \{x_1, x_2, \dots, x_n\}$	Set
$\mathcal{X}, [x_1, x_2)$	Space
X, Y	Random variables
$\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$	Initial guesses of $\boldsymbol{x}, \boldsymbol{y}$
$\hat{\boldsymbol{x}}, \boldsymbol{y}^*$	Optimum values of $\boldsymbol{x}, \boldsymbol{y}$
$P(X{=}x) = P(x)$	Discrete probability distribution
$p(X{=}x) = p(x)$	Probability density function

Table 2.1: Basic mathematical notation

2.2 Probability Theory

Many real world processes are stochastic, which means that they are random and precise predictions are not possible. However, stochastic models can often describe such processes and enable statistical analysis. A probability space is defined by $\Xi = \langle \Omega, S, P \rangle$ with the set of possible outcomes Ω, the σ-algebra S which is a collection of considered events, and the probability measure P which assigns probabilities to the events in S. Based on this definition, a random variable X is a measurable function $X : \Omega \to \mathcal{X}$ from the sample space Ω to a measurable space $\mathcal{X}$ [5]. If the measurable space $\mathcal{X}$ is discrete, samples from this random variable are distributed according to the probability distribution $P(X = x)$. Hence, the probabilities of this distribution sum up to one:

$$\sum_{x \in \mathcal{X}} P(X{=}x) = 1. \tag{2.1}$$

In case of a continuous measurable space (e.g. $\mathcal{X} = \mathbb{R}$), the probability density function $p(X{=}x)$ specifies the relative likelihood for a random variable to take a given value x. Thus, integrating $p(X{=}x)$ over the space $\mathcal{X}$ results in one:

$$\int_{x \in \mathcal{X}} p(X{=}x) dx = 1. \tag{2.2}$$

A joint probability distribution is denoted by $P(X{=}x, Y{=}y)$. If two random variables X and Y are independent, their joint probability distribution decomposes into

$$P(X{=}x, Y{=}y) = P(X{=}x) P(Y{=}y). \tag{2.3}$$

In contrast, if two random variables are conditionally dependent, knowledge about either of them provides information about the other:

$$P(X{=}x, Y{=}y) = P(X{=}x \mid Y{=}y) P(Y{=}y) \tag{2.4}$$

with $P(X{=}x \mid Y{=}y)$ being the probability of x given that y has occurred. Finally, the Bayes theorem relates the conditional probability distributions:

$$P(X{=}x \mid Y{=}y) = \frac{P(Y{=}y \mid X{=}x) P(X{=}x)}{P(Y{=}y)} \tag{2.5}$$

A causally conditioned probability $P(\boldsymbol{X}{=}\boldsymbol{x} \,/\!/\, \boldsymbol{Y}{=}\boldsymbol{y})$ specifies a conditional probability where each $\boldsymbol{x}_t$ is only conditioned on a portion of $\boldsymbol{y}_{1:t}$:

$$P(\boldsymbol{X}{=}\boldsymbol{x} \mathbin{/\!\!/} \boldsymbol{Y}{=}\boldsymbol{y}) = \prod_{t=1}^{T} P(\boldsymbol{X}_t{=}\boldsymbol{x}_t \mid \boldsymbol{Y}_{1:t}{=}\boldsymbol{y}_{1:t}, \boldsymbol{X}_{1:t-1}{=}\boldsymbol{x}_{1:t-1}). \tag{2.6}$$

The expectation of a random variable is the value to which the arithmetic mean converges if the number of samples approaches infinity. For discrete random variables the expectation is given by

$$\mathbb{E}\left[X\right] = \sum_{x \in \mathcal{X}} p(X{=}x)x. \tag{2.7}$$

Sometimes the expectation of a function $f(X)$ of a random variable X is of interest. This corresponds to the expected value of $f(x)$ under the probability distribution $p(X{=}x)$:

$$\mathbb{E}\left[f(X)\right] = \sum_{x \in \mathcal{X}} p(X{=}x)f(x). \tag{2.8}$$

Respectively, the expectation of a continuous random variable X with probability density function $p(X{=}x)$ is given by

$$\mathbb{E}\left[X\right] = \int_{x \in \mathcal{X}} p(X{=}x)x \; dx \tag{2.9}$$

and the expected value of function $f(X)$ is

$$\mathbb{E}\left[f(X)\right] = \int_{x \in \mathcal{X}} p(X{=}x)f(x) \; dx. \tag{2.10}$$

2.3 Information Theory

The information theory by Shannon [92] studies how to quantify, store, and communicate information. He introduced a famous measure, called entropy $H\left[X\right]$, which quantifies the amount of uncertainty of a random variable:

$$\begin{aligned} H\left[X\right] &= \mathbb{E}_{P(x)}\left[-\log(P(x))\right] && (2.11)\\ &= -\sum_{x \in \mathcal{X}} P(x) \log P(x). && (2.12)\end{aligned}$$

Accordingly, the entropy is zero, if the probability distribution is deterministic, because no information needs to be encoded. In all other cases, the entropy is a positive value.

The higher it is, the less informative is a probability distribution. Hence, the entropy of the random variable of a coin flip is lower than of rolling a dice. Since the entropy quantifies uncertainty, an outcome of a random variable with higher entropy contains more information. This definition of the entropy can be extended to continuous probability distributions:

$$H\left[X\right] = -\int_{x\in\mathcal{X}} p(x)\log p(x)\, dx. \tag{2.13}$$

Furthermore, the joint entropy measures the entropy of pairs of random variables X, Y:

$$H\left[X,Y\right] = \mathbb{E}_{P(x,y)}\left[-\log(P(x,y))\right]. \tag{2.14}$$

The conditional entropy [23] measures the average amount of required information to recover X if Y is known and thus extends information theory to conditional probability distributions:

$$\begin{aligned} H\left[X|Y\right] &= \mathbb{E}_{P(x,y)}\left[-\log(P(x|y))\right] && (2.15)\\ &= H\left[X,Y\right] - H\left[Y\right]. && (2.16)\end{aligned}$$

This measure can also be applied to causally conditioned probability distributions:

$$\begin{aligned} H\left[\boldsymbol{X} \,/\!/\, \boldsymbol{Y}\right] &= \mathbb{E}_{P(x,y)}\left[-\log(P(\boldsymbol{x} \,/\!/\, \boldsymbol{y}))\right] && (2.17)\\ &= \sum_{t=1}^{T} H\left[\boldsymbol{X}_t|\boldsymbol{Y}_{1:t}, \boldsymbol{X}_{1:t-1}\right]. && (2.18)\end{aligned}$$

The relative entropy (Kullback-Leibler divergence) [59] measures the additional amount of required information to compress data from a probability distribution $P(X)$ if the encoding $Q(X)$ is used:

$$\begin{aligned} H\left[P||Q\right] &= D_{KL}\left(P(x)||Q(x)\right) && (2.19)\\ &= \mathbb{E}_{P(x)}\left[\log\left(\frac{P(x)}{Q(x)}\right)\right]. && (2.20)\end{aligned}$$

This measure often serves as a distance metric between probability distributions. However, it is not symmetric and it does not satisfy the triangle inequality, which makes the relative entropy a semi-quasimetric.

2.4 Markov Decision Process

Since the dynamics of real world environments are typically partly random, models for sequential decision making need to take this randomness into account. Therefore, it is often not sufficient to compute an optimal plan that solves a problem, because the stochastic environment may cause to result in a state for which the plan does not provide any advice how to solve the problem. A mathematical framework for modeling decision-making under partly random outcomes is the Markov Decision Process [80]. A discounted, infinite horizon MDP is a discrete-time stochastic process, which is defined by a tuple $M = \langle S, A, P(s'|s,a), \gamma, R, P(s_0)\rangle$ with the variables defined as given by Tab. 2.2.

Fig. 2.1 illustrates an exemplary MDP with two states and three actions. In the first state, an agent can choose from two actions, while in the second state, only action c is available. As soon as an agent reaches the second state, he is never able to transition back to the first one. We will use this example throughout the thesis to discuss properties of the proposed approach.

Solving an MDP corresponds to finding a strategy for the decision-maker. Typically, this strategy should maximize the expected, cumulated, discounted reward yielding optimal behavior. Policies are the mathematical formulation of strategies, which are conditional probability distributions $\pi\left(a|s\right) = P\left(a|s\right)$, specifying the probability that an agent chooses action a in state s. The policy defines an action for every state of the state space. As a consequence, it is suitable for stochastic environments, since the policy provides optimal actions for all possible outcomes. The optimal policy $\pi^*\left(a|s\right)$

Notation	**Definition**
S	State space with states $s \in S$.
A	Action space with states $a \in A$.
$P\left(s'|s,a\right)$	Dynamics, which specify the probability to transition to s' if action a is applied in state s.
$\gamma \in [0,1)$	Discount factor.
$R : S \times A \to \mathbb{R}$	Reward function assigning a scalar reward to state-action tuples.
$P(s_0)$	Initial state probability distribution.

Table 2.2: Definition of a Markov Decision Process

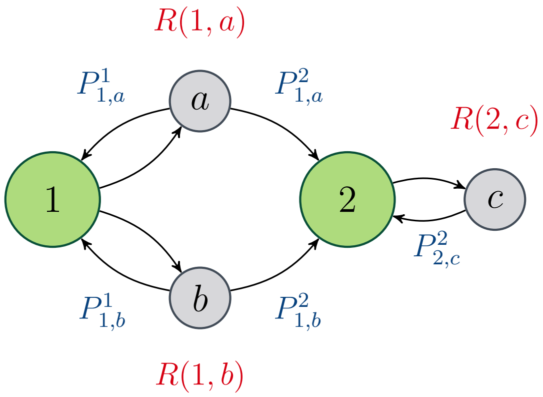


Figure 2.1: Exemplary Markov Decision Process with two states $s \in S = \{1, 2\}$ that are indicated in green, three actions $a \in A = \{a, b, c\}$ that are indicated in gray, transition probabilities $P^{s'}_{s,a} = P(s'|s,a)$ that are colored in blue, and rewards $R(s,a)$ that are illustrated in red. In state 1, an agent can choose from two available actions a and b, while transitions can occur to all states from the environment. As soon as the agent is in state 2, only action c is available and the agent will never be able to transition to state 1 again.

according to a reward function $R(s,a)$ maximizes the expected, cumulated, discounted reward (return). These optimal policies are often deterministic and can be expressed by a map $\pi_d : S \to A$, as all non-deterministic policies can only achieve a similar or a lower expected return. The value function specifies the expected, cumulated, and discounted reward for starting in a certain state s and executing actions according to the policy π:

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t, a_t\right) | s_0 = s, \pi\right]. \tag{2.21}$$

It is equally possible to define a state-action-value function (Q-function), specifying the expected, discounted, cumulated reward for starting in state s, picking action a and afterwards acting according to the policy π:

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t, a_t\right) | s_0 = s, a_0 = a, \pi\right]. \tag{2.22}$$

If a policy is known, the following fixed point equation computes the corresponding value- and Q-function, by solving:

$$V_i^\pi\left(s\right) = \mathbb{E}_{\pi(a|s)}\left[R\left(s,a\right) + \gamma \sum_{s' \in S} P\left(s'|s,a\right) V_{i-1}^\pi(s')\right] \qquad \forall s \in S. \tag{2.23}$$

Often the optimum policy $\pi^*\left(a|s\right)$ as well as the optimum value function are of interest.

Then, the Bellman equation [13] allows for finding a solution to the infinite horizon optimum control problem:

$$Q_i(s,a) = \left[R\left(s,a\right) + \gamma \sum_{s' \in S} P\left(s'|s,a\right) V_{i-1}(s') \right] \qquad \forall s \in S, a \in A, \tag{2.24}$$

$$V_i\left(s\right) = \max_a \left[Q_i(s,a)\right] \qquad \forall s \in S. \tag{2.25}$$

By repeatedly applying Eq. (2.24) and Eq. (2.25), both Q- and value-function converge $Q^*(s,a) = Q_\infty(s,a)$ and $V^*(s,a) = V_\infty(s)$, since it is a fixed point equation. From the optimal Q-function $Q^*(s,a)$ Eq. (2.24), it is possible to derive the optimal policy, by greedily choosing the action that promises the highest expected, cumulated, discounted reward:

$$\pi^*(s) = \operatorname*{argmax}_a Q^*(s,a). \tag{2.26}$$

Chapter 3

Inverse Reinforcement Learning

The following chapter introduces the field of Learning from Demonstration, gives a general overview about Inverse Reinforcement Learning, and presents exemplary problems that these methods can solve more efficiently. It reviews existing approaches, discusses their differences, and points out open problems. Subsequently, several of these approaches are explained in more detail to provide the theoretical foundation of this thesis.

Due to the increasing capabilities of autonomous systems, there is an emerging necessity for programming techniques that allow for efficient adjustment of the system's behavior. By enabling an autonomous system to be reparameterized to new behaviors, tasks, and environments, its application possibilities are highly increased. An intuitive and easy way to describe a task is to provide demonstrations of desired behavior. These demonstrations can then be used to learn strategies that solve the desired problem. A famous class of approaches offering such possibilities is Learning from Demonstration (LfD) [4, 7, 24, 35], which summarizes methods for teaching new behaviors and tasks by demonstration

(a) Autonomous Driving [56] **(b)** Manipulation tasks [3] **(c)** Activity forecasting [48]

Figure 3.1: Examples for different tasks and applications that have to some extent been solved by Learning from Demonstration.

instead of programming them directly. Famous examples of systems that have been successfully taught by human demonstrations are for example cars that drive and change lanes autonomously (Fig. 3.1a), robots that solve manipulation tasks (Fig. 3.1b), and prediction systems that forecast human behavior (Fig. 3.1c).

3.1 Survey on Inverse Reinforcement Learning

Common LfD methods belong to the fields of Behavioral Cloning (BC) [7] and Inverse Reinforcement Learning (IRL) [85, 93]. The goal of BC is to mimic the expert's behavior by estimating his policy directly from demonstrations (illustrated in Fig. 3.3a). This is possible, since expert demonstrations consist of both samples of the expert's policy and samples of the environment's dynamics, as illustrated in Fig. 3.2. Most BC approaches use supervised learning [10] to train a policy, which predicts the expert's action in a certain state, by matching the model to the observed behavior. By applying the predicted actions from the learned policies, the model should result in a similar behavior to the one of the expert when being deployed in a similar environment. For example, prior work uses BC for learning to drive [17, 79] or to fly [87]. However, a good parameterization and learning of policies is often not obvious, as it should generalize with high accuracy even to unobserved states, which differ to the states in the training set. Hence, the resulting policies are often only appropriate in states that are similar to the ones demonstrated by the expert.

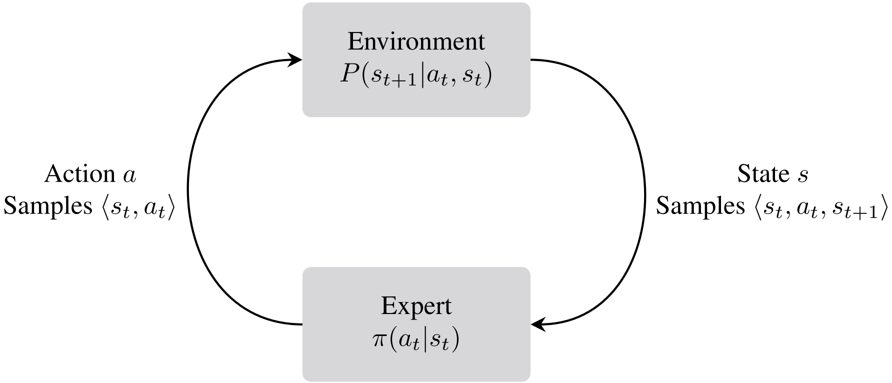
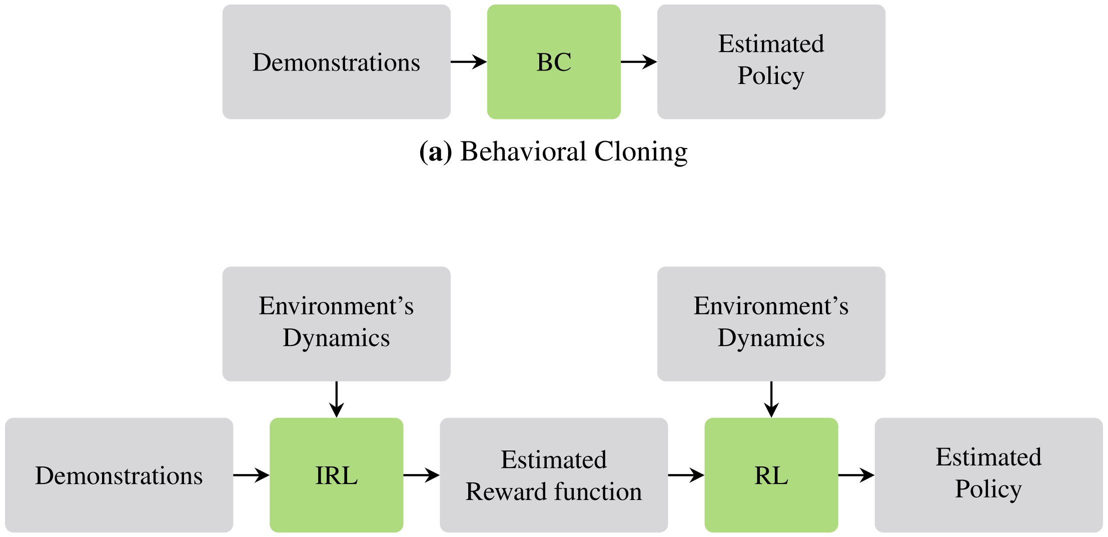

Environment
$P(s_{t+1}|a_t, s_t)$
Action a
Samples $\langle s_t, a_t \rangle$
State s
Samples $\langle s_t, a_t, s_{t+1} \rangle$
Expert
$\pi(a_t|s_t)$

Figure 3.2: Expert demonstrations are often generated by applying the expert's policy in the real or simulated environment. Typically, the expert provides demonstrations, by constantly taking decisions in an environment that reacts by changing the state according to the underlying environment's dynamics. Hence, an expert demonstration consists of sequential samples from the expert's policy and samples from the environment's dynamics.



(a) Behavioral Cloning

(b) Apprenticeship Learning via Inverse Reinforcement Learning

Figure 3.3: Learning from Demonstration through Behavioral Cloning or Inverse Reinforcement Learning. (a) Approaches from the field of Behavioral Cloning mimic the expert by learning a policy that behaves according to the observed behavior. (b) Inverse Reinforcement Learning based imitation learning approaches estimate the expert's reward function and use it as a representation of the expert's motivation. This reward function can then be used to recover a policy.

Imagine a human expert, who is training a car to drive autonomously. Since the human driver only provides examples of optimal or nearly-optimal behavior, it is unlikely to observe undesirable scenarios, such as hazardous situations that could result in accidents, as drivers anticipate them. However, the strategies that are necessary for successfully solving these problems can include complex strategies, such as evasive maneuvers and full braking. Generalizing to these strategies can be hard as real world observations rarely include them. Thus, BC often requires a massive amount of demonstrations that cover all parts of the state space that can occur. Furthermore, if the environment, the transition model, or the task changes, the estimated policies may not be optimal any more or even become unsuitable at all.

Therefore, another line of work proposes a class of approaches to obtain results that are more generalizable. In contrast to BC, IRL [76] estimates the underlying motivation and goals of an expert instead of copying its behavior directly. Those motivations and goals allow inferring optimal actions for unobserved states, new environments, or differing dynamics (illustrated in Fig. 3.3b). IRL has been successfully applied to several problems, such as learning to predict mouse pointing targets [120], cooperative human navigation strategies [52, 54, 55], lane keeping behavior [88, 89], parking lot navigation [2], or

Problem: Inverse Reinforcement Learning

Given:

- MDP $M \setminus R$ without the reward function
- Demonstrations D

Determine:

- Reward function R

Table 3.1: Definition of the Inverse Reinforcement Learning problem class.

driving styles [1, 38, 56, 62, 98].

In many of those problems, an intelligent agent [86] needs to make sequential decisions under partly random outcomes, which can be modeled as MDPs [13]. Given an environment and a reward function, solving an MDP corresponds to deriving optimal policies, e.g., by Reinforcement Learning (RL) [97]. However, if the environments or the problems are complex, specifying reward functions that yield a desired behavior can be difficult. Therefore, Russell [85] proposed IRL, which describes the problem of recovering the unknown reward function of an MDP from the fully known policy or from observed behavior of an agent acting according to some policy. Many publications [81] interpret the reward function as the most succinct representation of the expert's objective describing his motivations or goals. Tab. 3.1 characterizes the IRL problem class as defined in [76, 85], while the set of expert demonstrations D is given by

$$D = \{\tau_1, \tau_2, \ldots, \tau_N\} \tag{3.1}$$

with trajectories τ being defined as

$$\tau = \left\{ \langle s_0^\tau, a_0^\tau \rangle, \langle s_1^\tau, a_1^\tau \rangle, \ldots, \left\langle s_{T_\tau}^\tau, a_{T_\tau}^\tau \right\rangle \right\} . \tag{3.2}$$

Such demonstrations consist of samples $\langle s_t^\tau, a_t^\tau \rangle$ of the expert's policy $\pi(a|s)$ as well as samples $\left\langle s_t^\tau, a_t^\tau, s_{t+1}^\tau \right\rangle$ from the environment's dynamics $P(s_{t+1}^\tau | s_t, a_t)$. The goal of IRL is to estimate the agent's reward function $R(s, a)$, which explains the observed behavior in the demonstrations. Typically, this results in an optimization problem, minimizing some loss $R^* = \operatorname{argmin}_R L(\pi_E, \pi_{\boldsymbol{\theta}})$ between the expert's π_E and the learner's policy $\pi_{\boldsymbol{\theta}}$. If the agent always acts optimally, the solution set of reward functions [76], that might have caused the expert's behavior, is given by

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{t}R_{E}\left(s_{t},a_{t}\right)\mid\pi_{E}\right]\geq\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{t}R_{E}\left(s_{t},a_{t}\right)\mid\pi\right]\qquad\forall\pi. \tag{3.3}$$

With other words, this indicates that under the unknown, optimal expert's reward function R_E no other policy π results in a higher expected reward than the optimal expert's policy π_E. However, this problem formulation has several drawbacks: the whole expert's policy must be known, the expert must be indeed optimal, and there exist many reward functions that solve the problem, resulting in reward function ambiguities including degenerate solutions such as

$$R(s,a)=0\qquad\forall s,a. \tag{3.4}$$

Especially, if the state- and action-space is large or infinite, many approaches reduce the reward function space by expressing the reward as a linear function of weighted features $R(s,a)=\boldsymbol{\theta}^{\intercal}\boldsymbol{f}(s,a)$ with state- and action-dependent features $\boldsymbol{f}:S\times A\rightarrow\mathbb{R}^{d}$ (see [76]). In addition, this representation allows us to generalize reward functions to unobserved states and actions or new environments, if it is possible to compute the features for them. One might argue that linear reward functions might be too unexpressive to model complex motivations and goals. However, it is possible to add a large number of arbitrary features, which allows to adapt the reward function complexity. In this case, estimating the reward function corresponds to learning the optimal feature weights $\boldsymbol{\theta}^{*}$. Using this parameterization, Eq. (3.3) can be reduced to

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{t}\boldsymbol{\theta}_{E}{}^{\intercal}\boldsymbol{f}\left(s_{t},a_{t}\right)\mid\pi_{E}\right]\geq\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{t}\boldsymbol{\theta}_{E}{}^{\intercal}\boldsymbol{f}\left(s_{t},a_{t}\right)\mid\pi\right]\qquad\forall\pi \tag{3.5}$$

$$\boldsymbol{\theta}_{E}{}^{\intercal}\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{t}\boldsymbol{f}\left(s_{t},a_{t}\right)\mid\pi_{E}\right]\geq\boldsymbol{\theta}_{E}{}^{\intercal}\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{t}\boldsymbol{f}\left(s_{t},a_{t}\right)\mid\pi\right]\qquad\forall\pi \tag{3.6}$$

$$\boldsymbol{\theta}_{E}{}^{\intercal}\boldsymbol{\phi}\left(\pi_{E}\right)\geq\boldsymbol{\theta}_{E}{}^{\intercal}\boldsymbol{\phi}\left(\pi\right)\qquad\forall\pi \tag{3.7}$$

with $\boldsymbol{\phi}\left(\pi\right)=\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{t}\boldsymbol{f}\left(s_{t},a_{t}\right)\mid\pi\right]$ being the expected, cumulated, discounted features (feature expectation) values under policy π. Furthermore, this overcomes the necessity of knowing the whole expert's policy, since it suffices to estimate the expert's feature expectation by approximating it from a set of demonstrations $\widetilde{\boldsymbol{\phi}}=\frac{1}{|D|}\sum_{\tau\in D}\sum_{t=0}^{T_{\tau}}\gamma^{t}\boldsymbol{f}\left(s_{t}^{\tau},a_{t}^{\tau}\right)$. Most IRL approaches have in common that they search for a reward function for which the resulting policy matches the one of the expert. Abbeel and Ng [1] have shown that under the linear parameterization of the reward function it is sufficient to match the models expected cumulated features to the ones of the expert. Then, Eq. (3.7) guarantees that both, expert and model, perform similar in terms of the expected cumulated rewards.

Many approaches propose solutions to the IRL problem. Ng and Russel [76] point out that many reward functions including degenerate solutions may exist. They suggest adding penalty terms to overcome the impact of the ambiguities and formulate a linear program to find a feasible reward function. Abbeel and Ng [1] introduced apprenticeship learning to estimate a reward function under which the demonstrated behavior of the expert is optimal. They propose an approach of matching feature expectations between the observed policy and the learner's one. A similar idea by Ratliff *et al.*[83] formulates a solution as a Maximum Margin Planning (MMP) approach. The approach by Syed and Schapire [98] uses a game-theoretic framework to find a policy that can even outperform the expert's one. The LPAL algorithm by Syed *et al.*[99] extends the previous approach by proposing a linear programming formulation of the problem.

Another class of approaches uses Bayesian methods to infer the reward function of the expert from observed behavior. Baker *et al.*[8] propose a Bayesian approach that models the expert's policy as a Boltzmann distribution over state-action values. Babes-Vroman *et al.*[6, 107] propose to maximize the likelihood of the observed trajectories with respect to the reward function parameters and show that by using an expectation maximization (EM) based approach it is possible to learn from demonstrations with varying goals or objectives. Neu and Szepesvári [74] use a similar policy and suggest a natural gradient approach to minimize the squared loss between the estimated expert's and the learner's policy. Therefore, they derive the gradient of the converged Q-function with respect to the reward function parameters. Ramachandran and Amir [81] introduce a Bayesian model of the IRL problem, which allows incorporating prior knowledge and evidence from the expert to derive a posterior distribution over rewards. Their approach has been further generalized by Rothkopf and Dimitrakakis [84]. Neu and Szepesvári [75] derive a unified framework for IRL algorithms, which illustrates similarities and differences between various approaches, and Choi and Kim [21] show that a large set of IRL approaches solve a maximum a posteriori (MAP) estimation problem for identifying reward function. Furthermore, Choi and Kim [22] propose an IRL approach for partially observable MDPs (POMDPs) that allows to learn from experts that are not able to measure the true environmental state but rather a noisy estimate of it.

Usually humans generate expert demonstrations. Since they are rarely able to provide entirely optimal demonstrations, recent work derives approaches that can learn from noisy or sub-optimal demonstrations. Ziebart *et al.*[116, 119] propose a probabilistic model of maximum (causal) entropy (ME / MCE) to capture stochastic behavior while matching feature expectations in finite horizon problems. Bloem and Bambos [16] extend the approach to infinite time horizons. The class of maximum (causal) entropy IRL algorithms has been applied to a variety of different problems, such as inferring decisions of taxi drivers [117], predicting human trajectories by learning their motion behavior [54], probabilistic mouse pointing target prediction [120], learning to navigate in human crowds [37], or teaching an autonomous car adaptive driving behavior [38].

Many IRL approaches require expensive iterative computations that are often intractable, when applied to infinite spaces. Therefore, several approximate IRL methods have been proposed to allow them to be used in continuous state- and action-spaces. Ziebart [116] has shown that in the case of linear dynamics with Gaussian noise and a quadratic cost function, the Maximum Causal Entropy (MCE) Inverse Optimal Control (IOC) problem yields a closed-form solution. The approach by Boularias *et al.*[18] estimates the models feature expectation by approximating a high dimensional distribution over trajectories. However, this requires computing an intractable integral over the continuous space of trajectories, which they approximate via importance sampling. The guided cost learning approach by Finn *et al.*[30] extends the previous importance sampling based approach, by using policy optimization to slowly adapt the proposal distribution over trajectories towards the modeled one. In contrast, Kretzschmar *et al.*[51] use Hamiltonian Markov Chain Monte Carlo sampling to approximate the feature expectation of the model. The continuous IOC approach by Levine *et al.*[62] uses a Laplace approximation around trajectories, which locally models the distribution as a Gaussian. Thus, their approach learns local reward functions, which may not be suitable for globally optimal decision making. Huang *et al.*[44] propose a Maximum Entropy IOC variant, which estimates the models policy based on a soft value function approximation. Afterwards, their approach approximates the models feature expectation by Monte Carlo rollouts with the resulting policy.

The linear formulation of the reward function requires to define a set of features. While this allows incorporating knowledge into learning, it requires domain experts to hand-engineer appropriate features that are able to capture all possible goals and objectives. Several approaches try to mitigate the assumption of a linear reward function. Ratliff *et al.*[82] suggest to use boosting for learning a nonlinear function from a set of base features. Levine *et al.*[63] propose an approach that constructs reward features as logical conjunctions from a base feature set and simultaneously learns a reward function that explains the expert's behavior. For learning arbitrary nonlinear reward functions over features, Levine *et al.*[64] derive an approach based on Gaussian Processes and Wulfmeier *et al.*[112] propose to use fully convolutional neural networks. In [30], Finn *et al.*show how to learn nonlinear reward functions with deep neural network as function approximators from visual features trained via unsupervised learning. In contrast to previous approaches, this overcomes specifying hand-engineered features at all.

Most IRL approaches are computationally expensive and return a reward function as a solution, which cannot be directly applied for imitating the expert, since a policy would be needed. To overcome this burden, Ho and Ermon [42] propose Generative Adversarial Imitation Learning (GAIL), an approach that is derived from Maximum Causal Entropy IRL, but which directly trains a policy and overcomes learning the reward function. They show that for a specific type of regularization their approach reduces to a training procedure that is similar to the one of Generative Adversarial Networks (GAN). Furthermore, Finn *et al.*[29] show that sampling-based Maximum Entropy IRL approaches are mathematically equivalent to GANs with a specific choice of a discriminator. Since the GAN-based training needs to make rollouts in the environment, a massive amount of demonstrations is necessary to train both previous approaches. To reduce the number of system interactions Baram *et al.*[9] propose a model-based variant of GAIL, which initially trains a model of the dynamics based on the available samples of the transition model and subsequently learns an imitation policy, while optimizing full trajectories by differentiating through the fixed model of the dynamics. Recently, these GAN-based imitation learning approaches have been applied to several problems such as learning to drive on highways [57] or for third person imitation learning [95].

Except for the new GAN-based approaches, which no longer train a reward function, many of the traditional IRL approaches repeatedly apply a Reinforcement Learning (RL) algorithm to find optimal policies as part of the IRL algorithm. Solving the RL problem typically requires a known model of the system dynamics. Therefore, most of the IRL approaches assume that the dynamics are either given or can be estimated well enough from the given expert demonstrations in advance. However, estimating the transition model from goal-oriented expert demonstrations can be inaccurate, as rational decisions bias observed behavior towards desired states and actions. Therefore, large parts of the state- and action-space are rarely or never observed. For example, a human expert navigating a robot in a populated hallway will omit approaching humans too close or crashing into them. Consequently, estimating the dynamics of these situations is not possible without incorporating prior knowledge.

To enable IRL under unknown transition models, several approaches propose model-free methods. Tossou and Dimitrakakis [103] derive an approach based on a least squares approximation of the Q-function. It omits the need of solving the forward problem, and is applicable under unknown dynamics. Klein *et al.*[49] use similar assumptions and reformulate IRL to a problem of structured classification. The approach of Boularias *et al.*[18] minimizes the relative entropy between the probability distribution of trajectories following a baseline policy and the distribution under the learned policy, while matching expected feature counts. These model-free IRL variants either require expert demonstrations that are rich of observed transitions or additional samples from an arbitrary policy. However, coming back to the hallway navigation example, gathering robot demonstrations might bother humans or harm the environment. Therefore, it is often not feasible to obtain additional informative observations.

To account for this drawback we propose an approach called Simultaneous Estimation of Rewards and Dynamics (SERD) [40], which simultaneously optimizes the experts reward function and the environment's dynamics from expert demonstrations. This is possible as the expert's policy is a result of his reward function and his belief about the environment's dynamics.

3.2 Fundamentals

In the following, we explain Policy Matching IRL [74], Maximum (Causal) Entropy IRL [116, 119], and Relative Entropy (REIRL) [18] in more details, since we use them for the derivation of the proposed SERD approach and for comparison in the evaluation.

3.2.1 Policy Matching IRL

Neu and Szepesvári [74] propose a gradient based approach for apprenticeship learning using IRL, by assuming that the expert behaves optimally according to some unknown reward function. In contrast to classical approaches, where the goal was to find the reward function that the expert was optimizing, Policy Matching IRL (PM IRL) uses the reward function as a parameterization of the policy. Then, their algorithm aims to find a reward function, such that the resulting policy matches the observed behavior of the expert. Therefore, they propose to minimize the expectation of the squared loss between the learner's $\pi_{\boldsymbol{\theta}}$ and the expert's policy π_E

$$L(\pi_{\boldsymbol{\theta}}) = \frac{1}{2} \sum_{s \in S, a \in A} \mu_E(s) \left(\pi_{\boldsymbol{\theta}}(a \mid s) - \pi_E(a \mid s) \right)^2 \tag{3.8}$$

with respect to the empirical estimate of the probability distribution of states $\mu_E(s)$ and $\pi_E(a \mid s)$ being the full known expert's policy or an empirical estimate of it. Neu and Szepesvári propose to use a gradient-based method for minimizing the loss function in Eq. (3.8). By applying the chain rule, this results in the following gradient:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\theta}} L(\pi_{\boldsymbol{\theta}}) &= \frac{\partial L(\pi_{\boldsymbol{\theta}})}{\partial \pi_{\boldsymbol{\theta}}} \frac{\partial \pi_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} && (3.9) \\ &= \sum_{s \in S, a \in A} \mu_E(s) \left(\pi_{\boldsymbol{\theta}}(a \mid s) - \pi_E(a \mid s) \right) \frac{\partial \pi_{\boldsymbol{\theta}}(a \mid s)}{\partial \boldsymbol{\theta}} && (3.10) \end{aligned}$$

The computation of the policy's gradient $\frac{\partial \pi_{\boldsymbol{\theta}}(a|s)}{\partial \boldsymbol{\theta}}$ requires specifying a parametric function for the modeled policy. Since the expert is assumed to be optimal, a natural choice would be the optimal policy from Eq. (2.26), which is greedy and thus always chooses the action with maximum expected, cumulated, discounted reward $Q^*(s, a)$. However, the derivative of the argmax function is either zero or not differentiable and therefore can hardly be used for a gradient-based optimization. Instead, Neu and Szepesvári suggest using a Boltzmann distribution over actions to model the policy

$$\pi_{\boldsymbol{\theta}}(a \mid s) = \frac{\exp\left\{ \beta Q^*_{\boldsymbol{\theta}}(s, a) \right\}}{\sum_{a' \in A} \exp\left\{ \beta Q^*_{\boldsymbol{\theta}}(s, a') \right\}} \tag{3.11}$$

with the optimal, converged action-value function $Q^*_{\boldsymbol{\theta}}(s, a')$ and the inverse "temperature" β, which specifies how close $\pi_{\boldsymbol{\theta}}(a \mid s)$ is to the optimal policy. The corresponding gradient of this policy parameterization is given by:

$$\begin{aligned} \frac{\partial}{\partial \theta_k} \pi_{\boldsymbol{\theta}}(a \mid s) &= \pi_{\boldsymbol{\theta}}(a \mid s) \frac{\partial \ln\left\{ \pi_{\boldsymbol{\theta}}(a \mid s) \right\}}{\partial \theta_k} && (3.12) \\ &= \pi_{\boldsymbol{\theta}}(a \mid s) \beta \left(\frac{\partial Q^*_{\boldsymbol{\theta}}(s, a)}{\partial \theta_k} - \sum_{a' \in A} \pi_{\boldsymbol{\theta}}(a' \mid s) \frac{\partial Q^*_{\boldsymbol{\theta}}(s, a')}{\partial \theta_k} \right). && (3.13) \end{aligned}$$

The derivative of the policy in Eq. (3.13) depends on the partial derivative of the optimal Q-function $Q^*_{\boldsymbol{\theta}}(s, a')$, since it is also a function of the parameters $\boldsymbol{\theta}$. Neu and Szepesvári [74] show that the gradient of the converged Q-function exists and that it can be calculated almost everywhere, if the reward function is differentiable with respect to $\boldsymbol{\theta}$ with uniformly bounded derivatives. Furthermore, they derive a fixed-point equation whose solution is the gradient of the optimal Q-function $\frac{\partial Q^*_{\boldsymbol{\theta}}(s,a)}{\partial \theta_k}$ except on a set of measure zero. Therefore, they define the following operator S_π, which acts over the space of functions $\phi : |S| \times |A| \rightarrow \mathbb{R}^{|\Theta|}$ with the numbers of parameters $|\Theta|$, by

$$\left(S_{\pi^*}\phi_{\boldsymbol{\theta}}\right)(s,a) = \left(\frac{\partial r_{\boldsymbol{\theta}}(s,a)}{\partial \boldsymbol{\theta}}\right)^{\mathsf{T}} + \gamma \sum_{s' \in S} P(s' \mid s,a) \sum_{a' \in A} \pi^*_{\boldsymbol{\theta}}(a' \mid s)\phi_{\boldsymbol{\theta}}(s',a'). \quad (3.14)$$

where $\pi^*_{\boldsymbol{\theta}}$ is the greedy policy with respect to the optimal Q-function $Q^*_{\boldsymbol{\theta}}(s,a)$. They show, that by repeatedly applying this operator to some initial gradient, it will converge to the true gradient of the optimal Q-function. However, it should be noted that this is a subgradient, since the optimal Q-function from Eq. (2.24) and Eq. (2.25) incorporates the max-function, which may not be differentiable everywhere.

3.2.2 Maximum Entropy IRL

Since humans are rarely able to demonstrate entirely optimal behavior, approaches are necessary that can cope with stochastic or noisy demonstrations. The PM IRL approach allows learning from stochastic demonstrations, by minimizing an error between observed behavior and a stochastic model. However, due to the policy structure and the training method, it might be unsuitable to model real human behavior, since it is often unknown how humans behave. Consequently, the specified policy models might underfit, resulting in reward functions that are not solving the task. In contrast, if the model overfits, the learned reward function might not generalize and would only be accurate, if the environment and the observed states do not change.

Revisiting Eq. (3.7), every reward function yielding feature expectations that are identical to the ones of the expert will always have similar performance to him under this reward. At the same time, many different stochastic policies and thus probability distributions over trajectories are able to match feature expectations.

To solve the ambiguities of learning a reward function and choosing an appropriate distribution, Ziebart *et al.*[118] propose the Maximum Entropy IRL (ME IRL) approach. By employing the principle of maximum entropy [45], they search for a maximally uninformative probability distribution over trajectories under the constraint to match expected feature counts. This yields a distribution that is no more committed to any trajectory than the feature matching constraint requires.

Therefore, Ziebart *et al.*[118] formulate the following optimization problem:

$$\underset{P(\tau)}{\operatorname{argmax}} \; H\left[P(\tau)\right] \tag{3.15}$$

$$\text{s.t.} \;\; \mathbb{E}_{P(\tau)}\left[\phi_i(\tau)\right] = \widetilde{\phi}_i \qquad \forall i \tag{3.16}$$

$$\sum_{\tau \in \mathcal{T}} P(\tau) = 1 \tag{3.17}$$

$$P(\tau) \geq 0 \qquad \forall \tau \tag{3.18}$$

with $\mathcal{T}$ being the set of all valid trajectories, $\phi_i(\tau)$ being the i-th cumulated, discounted feature along trajectory τ, and $\widetilde{\phi}_i$ being the empirical expectation of the i-th feature under the expert's policy. The solution to this optimization problem is a probability distribution over trajectories that fulfills all mentioned criteria. Since the objective and the constraint space are both convex, optimization techniques [19] exist to solve the problem. The method of Lagrangian multipliers [25] results in the following Lagrangian:

$$\begin{aligned}\lambda\left(P, \boldsymbol{\theta}, \eta\right) = &- \sum_{\tau \in \mathcal{T}} P(\tau) \log P(\tau) - \sum_i \theta_i \left(\sum_{\tau \in \mathcal{T}} P(\tau)\phi_i(\tau) - \widetilde{\phi}_i\right) \\ &+ \eta \left(\sum_{\tau \in \mathcal{T}} P(\tau) - 1\right)\end{aligned} \tag{3.19}$$

Due to the Karush-Kuhn-Tucker conditions, the Lagrangian $\lambda\left(P, \boldsymbol{\theta}, \eta\right)$ is differentiable with respect to the unknown probability distribution $P(\tau)$, which can be set equal to zero:

$$\frac{\partial \lambda\left(P, \boldsymbol{\theta}, \eta\right)}{\partial P(\tau)} = \log\left(P(\tau)\right) - \sum_i \theta_i \phi_i(\tau) + \eta + 1 = 0 \tag{3.20}$$

By solving the equation for $P(\tau)$ and making use of $\sum_{\tau \in \mathcal{T}} P(\tau) = 1$, the distribution over trajectories according to the optimization problem in Eq. (3.27) results in:

$$P_{\boldsymbol{\theta}}(\tau) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(\boldsymbol{\theta}^\intercal \boldsymbol{\phi}(\tau)\right) \tag{3.21}$$

with the partition function $Z(\boldsymbol{\theta}) = \sum_{\tau \in \mathcal{T}} \exp\left(\boldsymbol{\theta}^\intercal \boldsymbol{\phi}(\tau)\right)$. Eq. (3.21) specifies a Boltzmann distribution over trajectories and implies that trajectories with a higher cumulated, discounted reward $R_{\boldsymbol{\theta}}(\tau) = \boldsymbol{\theta}^\intercal \boldsymbol{\phi}(\tau)$ are more likely. Interestingly, the reward function naturally became a linear combination of weights and features even though this has not been particularly specified. This is rather a direct consequence of the optimization problem, where the parameters $\boldsymbol{\theta}$ are the Lagrangian multipliers of the feature matching constraints.

Ziebart *et al.*[118] derive a gradient-based method from the dual problem, which optimizes the parameters $\boldsymbol{\theta}$ to ensure that the constraints are satisfied. Therefore, the Lagrangian $\lambda\left(P, \boldsymbol{\theta}, \eta\right)$ is differentiated with respect to the parameters $\boldsymbol{\theta}$, yielding

$$\frac{\partial\lambda\left(P, \boldsymbol{\theta}, \eta\right)}{\partial\boldsymbol{\theta}} = \widetilde{\boldsymbol{\phi}} - \sum_{\tau\in\mathcal{T}} P_{\boldsymbol{\theta}}(\tau)\boldsymbol{\phi}(\tau). \tag{3.22}$$

This gradient is the difference between the expert's empirical feature expectation and the expected, cumulated, discounted feature counts of the model. Hence, optimizing the parameters of the reward function according to the gradient causes the feature expectations of the expert and the model to match at the maxima. Furthermore, Ziebart *et al.*[118] show that the parameters can also be learned by maximizing the log likelihood of the demonstrations with respect to the feature weights $\boldsymbol{\theta}$ under the probability distribution over trajectories $P_{\boldsymbol{\theta}}(\tau)$. Then, the resulting gradient is identical to the one of the Maximum Entropy optimization problem in Eq. (3.22).

The ME IRL approach has been applied to a variety of reward learning problems and different variants have shown to outperform traditional approaches, especially, when learning from human demonstrations. In addition, Ziebart [116] provides worst-case guarantees, yielding a robust approach for learning reward functions from expert demonstrations.

3.2.3 Maximum Discounted Causal Entropy IRL

The statistical model of the ME IRL approach assumes the a priori availability of all necessary side information. However, the properties of many real world problems do not satisfy this assumption, as they include elements of interaction and feedback. For example, an agent acting in an environment with stochastic dynamics, cannot foresee how the state will change until it actually happened. In such cases, it is necessary to model the causal relationship between available information and decisions appropriately. Therefore, Ziebart *et al.*propose an extension to the ME IRL approach, called Maximum Causal Entropy IRL (MCE IRL) [119], in which they model sequentially revealed side information by causally conditioned probabilities. For solving infinite horizon problems Bloem and Bambos [16] as well as Zhou *et al.*[115] introduce Maximum Discounted Causal Entropy IRL. Since the causal entropy from Eq. (2.18) can be infinite in the infinite horizon case, they propose a finite variant with an exponentially discounted contribution of future causal entropy:

$$H_\beta \left[\boldsymbol{A} \,/\!/\, \boldsymbol{S}\right] = \mathbb{E}_{P(s_0)P(s_{t+1}|s_t,a_t),\pi} \left[\sum_{t=0}^{\infty} -\beta^t \log \pi\left(a_t \mid s_t\right)\right]. \tag{3.23}$$

Based on this definition, Bloem and Bambos define the Maximum Discounted Causal Entropy IRL (MDCE IRL) problem in a similar way as the ME IRL problem. The goal of the following optimization problem is to find a maximum discounted causal entropy policy under the constraint to match the model's feature expectation to the empirical one of the expert:

$$\underset{\pi}{\operatorname{argmax}} \;\; H_\beta \left[\boldsymbol{A} \,/\!/\, \boldsymbol{S}\right] \tag{3.24}$$

$$\text{s.t.} \;\; \mathbb{E}_{P(s_0)P(s'|s,a),\pi} \left[\sum_{t=0}^{\infty} \gamma^t f_i(s_t,a_t)\right] = \widetilde{\phi}_i \qquad \forall i \tag{3.25}$$

$$\sum_{a\in A} \pi(a \mid s) = 1 \qquad \forall s \tag{3.26}$$

$$P(\tau) \geq 0 \qquad \forall a, s \tag{3.27}$$

The general formulation in [16] allows different discount factors for the discounted causal entropy and the reward. However, since differing discount factors would result in non-stationary policies, they assume that $\beta = \gamma$. The formulation in Zhou *et al.*[115] directly applies the same type of discounting to the causal entropy and feature expectations. Again, it can be shown that the general problem is convex [16, 116] and can be solved with standard convex optimization methods. To solve the MDCE IRL problem, they derive a stationary soft value iteration, which shares similarities with Eq. (2.24) and Eq. (2.25). This results in the following soft state-action value function

$$Q^s_{\boldsymbol{\theta}}(s,a) = \boldsymbol{\theta}^T \boldsymbol{f}(s,a) + \gamma \sum_{s'\in S} P\left(s' \mid s,a\right) V_{\boldsymbol{\theta}}(s') \tag{3.28}$$

and soft value function

$$V^s_{\boldsymbol{\theta}}(s) = \log\left(\sum_{a\in A} \exp\left(Q^s_{\boldsymbol{\theta}}(s,a)\right)\right). \tag{3.29}$$

Similar to the Bellman equation, the soft value iteration is a fixed point equation, which is proven by Bloem and Bambos [16]. It should be noted that this softened version is differentiable and uses a smooth approximation of the maximum function, which guarantees that the soft value will always be higher or equal than the maximum of its individual Q-values $V_i^s(s) \geq \max_a Q_i^s(s,a)$. The resulting soft value function yields a stochastic policy, which is a Boltzmann distribution over soft Q-values:

$$\begin{aligned} \pi_{\boldsymbol{\theta}}(s,a) &= \exp(Q^s_{\boldsymbol{\theta}}(s,a) - V^s_{\boldsymbol{\theta}}(s)) \\ &= \frac{\exp(Q^s_{\boldsymbol{\theta}}(s,a))}{\sum_{a' \in A} \exp\left(Q^s_{\boldsymbol{\theta}}(s,a')\right)}. \end{aligned} \tag{3.30}$$

A notable property of this type of policy is that it chooses actions with a higher soft value more frequently. Estimating the expert's reward function corresponds to finding the feature weights under which the stochastic policy Eq. (3.30) explains the observed behavior. Hence, one can factorize the probability of a single trajectory τ given the start state distribution, the stochastic policy and the probability of transitions:

$$P(\tau \mid M, \boldsymbol{\theta}) = \; P(s_0^\tau) \prod_{t=0}^{T_\tau - 1} \left[\pi_{\boldsymbol{\theta}}\left(s_t^\tau, a_t^\tau\right) P\left(s_{t+1}^\tau \mid s_t^\tau, a_t^\tau\right)\right]. \tag{3.31}$$

Based on this definition and assuming independent trajectories, the likelihood of the demonstrations in D results in

$$P(D \mid M, \boldsymbol{\theta}) = \prod_{\tau \in D} P(\tau \mid M, \boldsymbol{\theta}). \tag{3.32}$$

Then, appropriate feature weights can be learned by maximizing the log likelihood of the demonstrations with respect to the feature weights $\boldsymbol{\theta}$ under the maximum causal entropy policy distribution $\pi_{\boldsymbol{\theta}}(s,a)$ from Eq. (3.30):

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \, J_{\boldsymbol{\theta}}\left(D\right) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log P(D \mid M, \boldsymbol{\theta}). \tag{3.33}$$

According to [16], the gradient-based optimization of the maximum entropy and the maximum likelihood estimate of the feature weights yields gradients that are the difference between empirical feature counts of the expert $\widetilde{\boldsymbol{\phi}}$ and the expected feature counts of the model $\boldsymbol{\phi}(\pi_{\boldsymbol{\theta}})$:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}\left(D\right) = \widetilde{\boldsymbol{\phi}} - \boldsymbol{\phi}(\pi_{\boldsymbol{\theta}}). \tag{3.34}$$

The guarantee of the consistency between maximum entropy and maximum likelihood estimates only holds true, if the empirical expert's policy matches the true one and inference uses the real underlying transition model. However, this is typically only true if the number of expert demonstrations approaches infinity. Otherwise, the result of the maximum likelihood estimation from Eq. (3.33) can differ from the estimate, produced by following the maximum entropy gradient from Eq. (3.34). Ziebart *et al.*[121] derive probabilistic bounds on the approximation error of the feature expectation, which enable allowing a small amount of slack by relaxing the constraints of the maximum causal entropy problem. This results in a regularized maximum causal likelihood estimation problem.

3.2.4 Relative Entropy IRL

Many problems exist for which accurate models of the environment are not available and it is not feasible to provide a large amount of expert demonstrations from the whole state space. Since most IRL approaches assume a model of the dynamics to be known, they cannot be applied to those types of problems. Furthermore, it is often not possible to estimate the environment's transition model directly from the observed demonstrations, as they only cover a small part of the state space. To overcome those drawbacks, Boularias *et al.*[18] propose a model-free IRL approach, called Relative Entropy IRL (REIRL). It extends ME IRL [118] to problems where expert demonstrations are not covering the whole state space and a model of the environment's dynamics is not available. Instead, system interaction is required for querying an additional set of demonstration from an arbitrary, known policy from the environment.

For this purpose, they propose to minimize the relative entropy between the agent's probability distribution over trajectories $P(\tau)$ and some baseline distribution $Q(\tau)$ under the feature matching constraints:

$$\operatorname*{argmin}_{P(\tau)} \quad H\left[P(\tau)||Q(\tau)\right] \tag{3.35}$$

$$\text{s.t.} \quad \left|\sum_{\tau\in\mathcal{T}} P(\tau)\phi_i(\tau) - \widetilde{\phi}_i\right| \leq \epsilon_i \qquad \forall i \tag{3.36}$$

$$\sum_{\tau\in\mathcal{T}} P(\tau) = 1 \tag{3.37}$$

$$P(\tau) \geq 0 \qquad \forall\tau \tag{3.38}$$

with the set of all possible trajectories $\mathcal{T}$ and $Q(\tau)$ being the probability distribution over trajectories if some baseline policy π_Q is applied in the environment. Hoeffding's bound

[43] allows for computing the thresholds ϵ_i for the i-th feature dimension, as suggested in [18]. This problem formulation differs from ME IRL by exchanging the entropy in the primal problem by the Kullback-Leibler divergence [58]. Furthermore, it is required to specify a baseline policy π_Q, which causes the probability distribution over trajectories $Q(\tau)$. Therefore, the optimization problem searches for a probability distribution $P(\tau)$ that is as close as possible to $Q(\tau)$ under the relative entropy measure while satisfying the feature matching constraints.

If a uniform baseline distribution $Q(\tau) = C$ is used, the REIRL optimization problem equals to ME IRL, since the uniform distribution is the one with maximum entropy among all distribution. Then, minimizing the relative entropy between $H\left[P(\tau)||Q(\tau)\right]$ corresponds to maximizing the entropy of $P(\tau)$:

$$
\begin{aligned}
& H\left[P(\tau)||Q(\tau)\right] && (3.39)\\
=& \sum_{\tau \in \mathcal{T}} P(\tau) \log \frac{P(\tau)}{C} && (3.40)\\
=& \sum_{\tau \in \mathcal{T}} P(\tau) \log P(\tau) - \log C && (3.41)\\
=& -H(P(\tau)) - \log C. && (3.42)
\end{aligned}
$$

Hence, the primal problem of both optimization problems would be equal up to a constant. As a consequence, REIRL is similar to ME IRL when using maximum entropy baseline policies for the baseline distribution $Q(\tau)$. However, in general REIRL allows to use arbitrary baseline distributions and thus allows to incorporate prior information into learning by either assuming the baseline policy π_Q or the corresponding probability distribution over trajectories $Q(\tau)$ to be known.

Boularias *et al.*[18] derive a solution to the optimization problem by using the calculus of variations [31]. It shares similarities to the solutions of ME IRL and MDCE IRL and is given by:

$$
P(\tau \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} Q(\tau) \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\tau)\right) \tag{3.43}
$$

with the partition function $Z(\boldsymbol{\theta})$ and the cumulated, discounted feature counts $\boldsymbol{\phi}(\tau)$ of trajectory τ. Again, it is possible to include the constant term into the partition function $Z(\boldsymbol{\theta})$, if the baseline distribution is uniform $Q(\tau) = C$. Then, the resulting probability distribution over trajectories $P(\tau \mid \boldsymbol{\theta})$ in Eq. (3.43) is identical to the one of ME IRL in Eq. (3.21).

Boularias *et al.*show that the following gradient estimates the parameters $\boldsymbol{\theta}$ by subgradient ascent:

$$\frac{\partial g}{\partial \boldsymbol{\theta}} = \widetilde{\boldsymbol{\phi}} - \sum_{\tau \in \mathcal{T}} P(\tau \mid \boldsymbol{\theta}) \boldsymbol{\phi}(\tau) - \boldsymbol{\alpha} \circ \boldsymbol{\epsilon} \tag{3.44}$$

with $\forall \alpha_i = \operatorname{sign} \theta_i$, thresholds $\boldsymbol{\epsilon}$, and $\circ$ indicating the Hadamard product. As in the previous approaches, at the optimum this gradient yields feature matching. However, in contrast to ME IRL or MDCE IRL the last term incorporates some slack for matching features. The exact computation of this gradient is only practical in small or discrete state- and action-spaces. If the set of possible trajectories $\mathcal{T}$ becomes large or infinite, estimating the partition function $Z(\boldsymbol{\theta})$ can be computationally expensive or unfeasible at all. Furthermore, its direct computation typically requires the dynamics of the system to be known to perform exact inference.

In contrast, Boularias *et al.*[18] propose to approximate the probability distribution of demonstrations under the learner's model $P(\tau \mid \boldsymbol{\theta})$ by importance sampling from an arbitrary policy π_a. Furthermore, they omit the need of a model of the dynamics, since the arbitrary demonstrations $\tau_N^{\pi_a}$ consist of samples of the environment's dynamics. Thus, the approximate sample-based gradient of REIRL is

$$\begin{aligned} \frac{\partial g}{\partial \boldsymbol{\theta}} &= \widetilde{\boldsymbol{\phi}} - \sum_{\tau \in \mathcal{T}} \frac{Q(\tau) \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\tau)\right)}{\sum_{\eta \in \mathcal{T}} Q(\eta) \exp\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\eta)\right)} \boldsymbol{\phi}(\tau) - \boldsymbol{\alpha} \circ \boldsymbol{\epsilon} & (3.45) \\ &= \widetilde{\boldsymbol{\phi}} - \frac{\sum_{\tau \in \tau_N^{\pi_a}} \frac{U_{\pi_Q}(\tau)}{U_{\pi_a}(\tau)} \exp\left(\boldsymbol{\theta}^\intercal \boldsymbol{\phi}(\tau)\right) \boldsymbol{\phi}(\tau)}{\sum_{\tau \in \tau_N^{\pi_a}} \frac{U_{\pi_Q}(\tau)}{U_{\pi_a}(\tau)} \exp\left(\boldsymbol{\theta}^\intercal \boldsymbol{\phi}(\tau)\right)} - \boldsymbol{\alpha} \circ \boldsymbol{\epsilon}. & (3.46) \end{aligned}$$

with the joint probability of actions under the baseline policy $U_{\pi_Q}(\tau)$, the joint probability of actions under the arbitrary policy $U_{\pi_a}(\tau)$, and $U_\pi(\tau) = \prod_{t=0}^{T_\tau - 1} \pi\left(a_t^\tau \mid s_t^\tau\right)$. The proposed approach uses Monte Carlo integration to compute the partition function of $P(\tau \mid \boldsymbol{\theta})$. Consequently, a reliable estimate in the high dimensional space of trajectories often requires a large number of demonstrations. REIRL requires to sample these demonstrations from the real underlying dynamics of the environment.

Often, only expert demonstrations are available and system interaction is not possible. Then, additional demonstrations can only be generated artificially. By estimating a model of the environment's dynamics directly from the observed transitions, it can serve as a surrogate of the real environment. Afterwards, the learned transition model allows for inference or for sampling. However, since expert demonstrations often only cover a small part of the state space, the learned model might be inaccurate for unobserved states or actions.

3.3 Summary

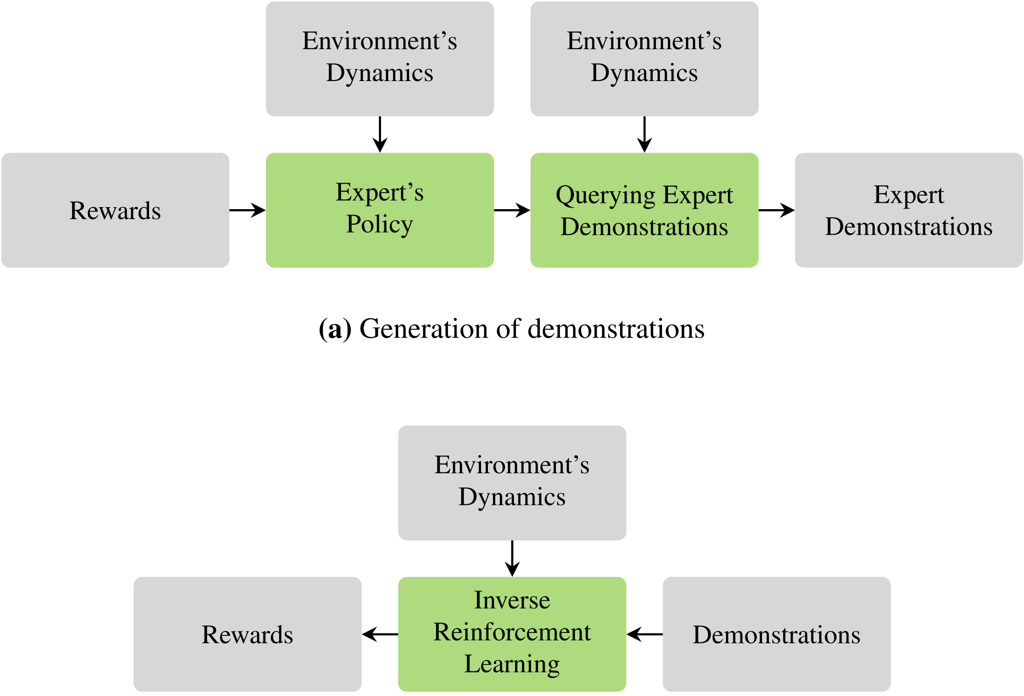
This chapter gave an overview on Learning from Demonstration and especially Inverse Reinforcement Learning. It outlined the history of IRL approaches and pointed out their differences and similarities. As a basis for the approaches presented in this thesis, we presented some methods in more detail. Many IRL approaches assume that the stochastic system dynamics are known, can be estimated from expert demonstrations, or additional samples from the transition model can be queried in states and actions that have been rarely observed. However, these assumptions are often not satisfied, as in many real world problems accurate transition models are not accessible and querying samples is too expensive. In those cases, algorithms are necessary that either can cope with inaccurate transition models or are able to improve the wrong model.

Chapter 4

Simultaneous Estimation of Rewards and Dynamics

Many IRL approaches assume the environment's dynamics to be known or additional samples to be available for learning. This is often not the case, but expert demonstrations consist of samples from the real or simulated dynamics of the environment as well as samples from the expert's policy. Since expert policies depend on the stochastic system dynamics and the reward function, demonstrated behavior of an expert allows estimating both. In this chapter, we outline limitations of current IRL approaches under unknown dynamics based on a minimum example, we introduce a new problem class that is able to overcome those limitations to a certain degree, and we provide a solution to this problem class based on maximum a-posteriori (MAP) estimation of rewards and dynamics. Furthermore, we derive solutions for two different policy models, discuss priors than can be used for incorporating knowledge into learning, and evaluate the approach based on two examples.

As the capabilities of intelligent systems (e.g. robots) improve, they can solve more and more complex problems. Consequently, the number of such systems performing tasks in populated areas is constantly increasing. However, to ensure a fast and easy deployment in various environments, it is necessary to provide simple programming approaches for non-experts to parameterize new behaviors, goals or tasks. A promising approach for teaching systems a certain behavior is Inverse Reinforcement Learning (IRL), which estimates the underlying reward function of a Markov Decision Process (MDP) from



(a) Generation of demonstrations

(b) Inverse Reinforcement Learning

Figure 4.1: (a) An expert computes his policy by considering his reward function as well as the environment's dynamics. When applying this policy in a real or a simulated environment, generated demonstrations consist of samples of the expert's policy and samples of the environment's dynamics. (b) Most of the IRL approaches assume that besides expert demonstrations a transition model of the environment is available to estimate the expert's reward function. Approaches that avoid the need of a model of the dynamics typically require additional samples from a non-optimal policy or apply heuristics.

observed behavior of an expert.

In the previous chapter, we have outlined the state of the art of IRL. However, current approaches have a couple of limitations. Fig. 4.1 illustrates the generation of expert demonstrations and IRL as block diagrams. Most of the model-based approaches need to solve the RL problem repeatedly as part of solving the IRL problem, which usually requires an accurate model of the environment's dynamics. Typically, they assume that this model is known or that it can be accurately estimated from observed transitions directly. However, these approaches often ignore that the provided demonstrations are the result of an expert's policy, which biases them towards high expected cumulated rewards, while many states and actions are rarely or never observed. When naively estimating a model of the dynamics from these demonstrations, it is likely to be inaccurate for rare states and actions. While the model-free approaches omit the need of a model of the dynamics, they have stronger requirements on the provided demonstrations, by using them directly as a proxy of the transition model. For example, some approaches require that the observed demonstrations consist of enough samples of the environment's dynamics to accurately estimate the reward function. Other approaches need access to the environment to generate additional demonstrations or introduce heuristics to account for unobserved transitions. However, sampling demonstrations from the environment can be costly, realistic simulators often do not exist, and defining appropriate heuristics is difficult for many environments. Consequently, model-free approaches may tend to estimate wrong reward functions.

Only little research has been conducted on the robustness of IRL against wrong transition models, even though Ramachandran and Amir pointed it out already in [81]. Since most IRL approaches assume the given transition model to be true, they might draw wrong conclusions. However, both the reward function and the dynamics of the environment influence the expert's policy. Therefore, the frequency of state-action pairs carries information about both. To the best of our knowledge, the first publication that exploits one side of this influence was by Phatak *et al.*[78], which estimates the system dynamics from expert demonstrations under a known reward function. Recently, Golub *et al.*[33] proposed an approach that estimates the human's belief about the system dynamics from expert demonstrations under a known reward function.

In this chapter, we integrate the estimation of the environment's dynamics into IRL. We call this problem class Simultaneous Estimation of Rewards and Dynamics (SERD) [39, 40] and exploit the fact that a policy has generated the expert's demonstrations, which is the result of both rewards and dynamics. To solve it, we propose a unified framework of maximum a-posteriori inference of rewards and dynamics, which includes IRL and MDP model estimation as special problem cases. We provide a general gradient-based solution and show derivations of this gradient from the unified framework under different assumptions about the expert's policy generation. Furthermore, we introduce and discuss different priors to avoid overfitting. To clarify characteristics and properties of IRL and SERD, we present a minimum example showing the problem of interpreting demonstrations in light of the bilateral influence of rewards and dynamics onto the policy. We use this example as well as a grid-world navigation task based on satellite images for evaluation, showing that the proposed approach outperforms other classical ones, by improving the estimate of the dynamics, the estimate of the reward function, as well as the sample efficiency of learning.

4.1 Limitations of IRL approaches

To clarify properties of current IRL approaches, we present a minimum example that illustrates the problem of interpreting demonstrations in light of the bilateral influence of rewards and dynamics onto the policy, if the dynamics of the environment are unknown to the learner. The minimum example is depicted in Fig. 4.2 and consists of two states $S \in \{1, 2\}$ and three actions $A \in \{a, b, c\}$. An agent is allowed to choose between action a or b in state 1, while only action c is available in state 2. As soon as the agent reaches state 2, he is not able to transit back to state 1, since only action c is available with the single successor state 2. Consider that an expert chooses actions from an optimal policy, providing the two demonstrations from Fig. 4.2:

$$\tau_b = \{\langle s_0{=}1, a_0{=}a\rangle, \langle s_1{=}1, a_1{=}a\rangle, \langle s_2{=}2, a_2{=}c\rangle\}, \tag{4.1}$$

$$\tau_r = \{\langle s_0{=}1, a_0{=}a\rangle, \langle s_1{=}2, a_1{=}c\rangle, \langle s_2{=}2, a_2{=}c\rangle\}, \tag{4.2}$$

colored in blue and red. Both demonstrations are three steps long and start in state 1, where always action a is executed. Most IRL approaches assume prior knowledge about the dynamics of the environment. If this is not the case, many of them propose to estimate a model from the observed transitions in the demonstrations. A naive transition model of the environment can be estimated by computing the frequency of transitions conditioned on the previous state and action. Such a naive estimate is depicted in blue in Fig. 4.3 (b) and (c). However, the demonstrations only cover action a and tell nothing about the transition

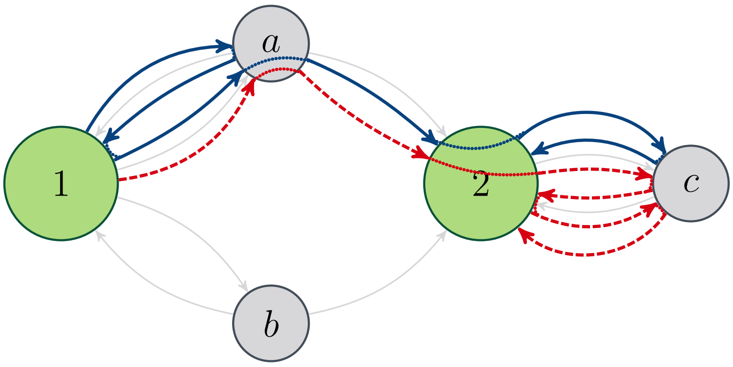
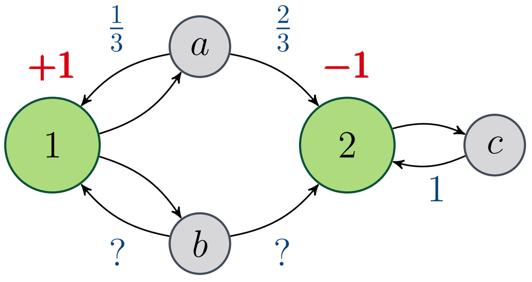


Figure 4.2: Minimum example of IRL with two different expert demonstrations (colored in blue and red). Both demonstrations always start in state 1, where the expert prefers action a, while action b is never chosen. As soon as an agent is in the second state, he cannot transition back to state 1. The goal of IRL is to recover the unknown reward function that caused the demonstrations. However, unknown transition models might complicate interpreting the observed behavior.



(a) Option 1

(b) Option 2

Figure 4.3: Minimum example of IRL based on the previously introduced MDP. Traditional IRL approaches require the dynamics to be known. Therefore, a naive estimate of the dynamics from the given demonstrations is illustrated in (a), and (b) (depicted in blue) . Since action b has never been observed, the transition probability to the successor states cannot be computed. To reduce the complexity of this example, only state-dependent reward functions (depicted in red) are considered $\forall a \in A, s \in S : R(s, a) = R(s)$. Which of the two options for the reward functions (a) or (b) is more likely?

probabilities of action b. Hence, it is not possible to specify the transition probabilities to successor states after executing action b in state 1. Estimating the expert's reward function corresponds to specifying which of the two states has the higher immediate reward, since the actual value has no influence on the optimal, deterministic policy. To reduce the complexity of the IRL example, only state-dependent reward functions are considered $\forall a \in A, a' \in A, s \in S : R(s, a) = R(s, a')$ in this minimum example, which are depicted in red. Fig. 4.3 (a) and (b) show two options for the reward function. Is option 1 or 2 more likely to be the true reward function of the expert, who provided the demonstrations shown in Fig. 4.2?

Assuming that the expert chooses his actions to maximize the expected, cumulated, discounted reward, interpreting his motivation is equivalent to arguing about both his reward function and the environment's dynamics:

- **Option 1:** Since the first state has a higher reward, the expert chooses the action that has the higher likelihood of staying in state 1. Hence, action a is chosen, if the transition probability to stay in state 1 of action a is higher than that of action b: $P(1 \mid 1, b) < P(1 \mid 1, a) \approx \frac{1}{3}$. Otherwise, the expert would have chosen action b.

- **Option 2:** If the second state is more preferable due to a higher reward, the expert chose action a, because its likelihood to transition to state 2 is higher than that of action b: $P(2 \mid 1, b) < P(2 \mid 1, a) \approx \frac{2}{3}$.

Interestingly, in an informal survey most humans assume that option 2 is the expert's reward function. However, as we have shown, both options can explain the expert's behavior equally well and wrong assumptions about the unknown dynamics would result in possibly wrong interpretations. Only by considering that the policy is the result of the reward function and the dynamics one can find both options. If unknown or inaccurate dynamics are used for IRL, this has to be taken into account. This example illustrates several aspects of IRL that most available approaches do not address:

Expert demonstration bias:

- Models of the dynamics that are estimated naively from observed transitions of an expert are likely to be inaccurate or partially unknown, since **expert demonstrations are biased** towards high values.

- The state-action tuples in the demonstrations are samples from the **expert's policy**. Therefore, they incorporate information about the dynamics, which influenced the policy.

Ambiguities:

- (Partially) unknown dynamics can introduce **ambiguities**.

- The additional ambiguities can only be found by estimating the rewards as well as the dynamics and considering their **bilateral influence** on the policy.

- **Prior information** can resolve ambiguities.

- **Inaccurate transition models** may tamper with the reward estimates.

In summary, if an agent acts according to some (sub-) optimal policy in an MDP, it is possible to infer better dynamics models to a certain degree, by exploiting the fact that policies consider the rewards and the transition model. In previous work, we have introduced this new problem class and proposed first approaches for solving the SERD problem [39, 40]. This can be advantageous, because a simultaneous estimation captures the bilateral influence of the transition model and the reward function on the policy. We propose a unified approach based on a maximum a-posteriori formulation to resolve ambiguities and to prevent overfitting, due to training a large amount of parameters from a possibly small number of demonstrations. Furthermore, we derive models for different problem classes and discuss several priors.

4.2 Problem Formulation

We define an extension to the classical IRL problem, where neither the reward function, nor the dynamics are known and both should be estimated from demonstrations. Furthermore, we consider an aspect that many available approaches neglect: the expert may have imperfect knowledge about the environment. If this is the case, IRL approaches might wrongly interpret the expert's behavior yielding inaccurate or unfeasible reward functions.

Fig. 4.4 illustrates the process of generating demonstrations and indicates the encoded information. If an expert provides demonstrations in a real or simulated environment, the individual samples $\left\langle s_t^\tau, a_t^\tau, s_{t+1}^\tau \right\rangle$ allow for estimating the dynamics of the environment. In addition, the tuples $\langle s_t^\tau, a_t^\tau \rangle$ are samples of the expert's policy. Since his reward function and his internal model of the environment's dynamics influenced his policy, these samples give information about both.

By considering this influence, it is even possible to estimate the expert's internal model of the environment's dynamics, covering the agent's belief about how the world behaves [33]. In addition, this allows to argue about reasons for suboptimal expert behavior. Furthermore, for some problems we can make assumptions on the similarity between the expert's model of the environment's dynamics and the real transition model. They could be completely identical, they could be identical in certain areas of the state- and action-space, or it could be possible to constrain their difference in terms of some measure (e.g. Kullback-Leibler divergence). Then, it is possible to estimate more accurate transition models, by considering both sources of information: the transitions in the demonstrations and the influence of the environment's dynamics onto the expert's policy.

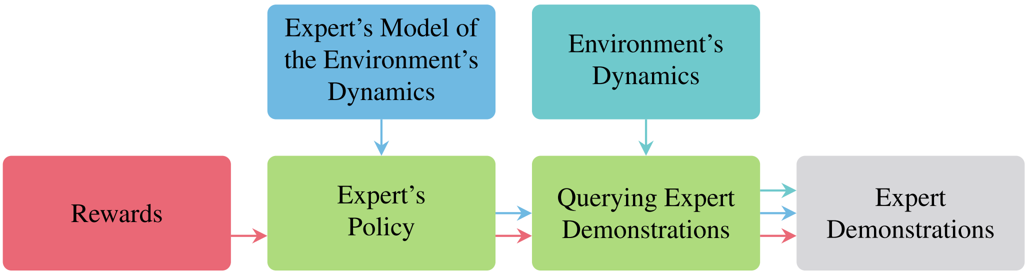

Expert's Model of the Environment's Dynamics
Environment's Dynamics
Rewards
Expert's Policy
Querying Expert Demonstrations
Expert Demonstrations

Figure 4.4: The expert generates his policy by considering his reward function and his model of the environment's dynamics. However, this model can be inaccurate and might tamper with the expert's policy such that it is suboptimal in the real environment. If this policy is applied in a real or a simulated environment, demonstrations consist of samples of the expert's policy and samples of the environment's dynamics. Hence, samples are influence by the expert's reward function, his model of the environment's dynamics and its real transition model.

Problem: Simultaneous Estimation of Rewards and Dynamics

Given:

- MDP without the reward function and dynamics:
  $M \setminus \{R, P\left(s' \mid s, a\right), P_E\left(s' \mid s, a\right)\}$
- Demonstrations $D = \{\tau_1, \tau_2, \ldots, \tau_N\}$ with trajectories
  $\tau = \left\{\left(s_0^\tau, a_0^\tau\right), \left(s_1^\tau, a_1^\tau\right), \ldots, \left(s_{T_\tau}^\tau, a_{T_\tau}^\tau\right)\right\}$ of an expert acting in M based on a policy that depends on $R(s, a)$, $P_E\left(s' \mid s, a\right)$, and $g(Q)$

Determine:

- Expert's reward function $R(s, a)$
- Expert's estimate of the dynamics $P_E\left(s' \mid s, a\right)$
- Real dynamics $P\left(s' \mid s, a\right)$
- Stochastic policy model $\pi = g(Q)$

Table 4.1: Definition of the Simultaneous Estimation of Rewards and Dynamics problem class.

In order to consider a possibly inaccurate belief on the environment's dynamics, our problem definition differentiates between the expert's estimate of the system's dynamics $P_E\left(s' \mid s, a\right)$ and the true transition model of the environment $P\left(s' \mid s, a\right)$. Typically, the stochastic policy $\pi = g(Q)$ is a function of the state-action values Q^* and possible additional, unknown parameters. This allows incorporating the expected, cumulated, discounted future reward into the policy, by modeling stochasticity that depends on the difference between the Q-values of different actions. Altogether, Tab. 4.1 characterizes the SERD problem class, which specifies the problem of estimating the expert's reward function, his belief about the environment's dynamics, the real dynamics of the environment, as well as his stochastic policy model solely from expert demonstrations, while assuming that the expert computed his policy based on his reward function and his belief about the environment's dynamics.

To solve this problem, we assume to learn parameters of the problem statement's parameterizable functions, distributions, or mappings simultaneously from expert demonstrations D. Therefore, we introduce the following parameters:

$\boldsymbol{\theta}_R$ Feature weights of the reward function $R(s,a)$,
$\boldsymbol{\theta}_{T_E}$ Parameters of the expert's transition model $P_{\boldsymbol{\theta}_{T_E}}$,
$\boldsymbol{\theta}_T$ Parameters of the real transition model $P_{\boldsymbol{\theta}_T}$,
$\boldsymbol{\theta}_P$ Parameters of the expert's policy $\pi = g(\boldsymbol{\theta}_P, Q_{\boldsymbol{\theta}})$.

4.3 Unified SERD approach

We propose to solve the SERD problem by formulating a maximum a-posteriori optimization problem. This allows to incorporate prior information to resolve ambiguities and to prevent overfitting. The MAP estimate of the parameters $\boldsymbol{\theta} = \begin{pmatrix}\boldsymbol{\theta}_R^\intercal & \boldsymbol{\theta}_{T_E}^\intercal & \boldsymbol{\theta}_T^\intercal & \boldsymbol{\theta}_P^\intercal\end{pmatrix}^\intercal$ is given by

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \operatorname*{argmax}_{\boldsymbol{\theta}} \log P(\boldsymbol{\theta} \mid D, M) && (4.3)\\
&= \operatorname*{argmax}_{\boldsymbol{\theta}} \log \frac{P(D \mid M, \boldsymbol{\theta}) P(\boldsymbol{\theta})}{P(D \mid M)} && (4.4)\\
&= \operatorname*{argmax}_{\boldsymbol{\theta}} \log \underbrace{P(D \mid M, \boldsymbol{\theta})}_{\text{likelihood}} + \log \underbrace{P(\boldsymbol{\theta})}_{\text{prior}} && (4.5)\\
&= \operatorname*{argmax}_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(D)\,. && (4.6)
\end{aligned}$$

If the observed trajectories are independent and identically distributed, the likelihood of the demonstrations D decomposes to

$$P(D \mid M, \boldsymbol{\theta}) = \prod_{\tau \in D} P(s_0^\tau) \prod_{t=0}^{T_\tau - 1} \left[\pi_{\boldsymbol{\theta}}\left(a_t^\tau \mid s_t^\tau\right) P_{\boldsymbol{\theta}_T}\left(s_{t+1}^\tau \mid s_t^\tau, a_t^\tau\right)\right]. \qquad (4.7)$$

Then, the gradient of the objective $J_{\boldsymbol{\theta}}(D)$ allows solving the optimization problem via gradient-ascent:

$$\begin{aligned}
\frac{\partial}{\partial \theta_i} J_{\boldsymbol{\theta}}(D) =& \sum_{\tau \in D} \sum_{t=0}^{T_\tau - 1} \left[\frac{\partial}{\partial \theta_i} \log \pi_{\boldsymbol{\theta}}\left(a_t^\tau \mid s_t^\tau\right) + \frac{\partial}{\partial \theta_i} \log P_{\boldsymbol{\theta}_T}\left(s_{t+1}^\tau \mid s_t^\tau, a_t^\tau\right)\right] && (4.8)\\
&+ \frac{\partial}{\partial \theta_i} \log P(\boldsymbol{\theta}). && (4.9)
\end{aligned}$$

Here, the policy $\pi_{\boldsymbol{\theta}}(a \mid s)$ depends on all parameters except $\boldsymbol{\theta}_T$, because the expert derived his policy based on his reward function and his estimate of the environment's dynamics. In contrast, the observed transitions occur in the real or simulated environment. Therefore, the transition model $P_{\boldsymbol{\theta}_T}(s' \mid s, a)$ only depends on $\boldsymbol{\theta}_T$. If $P_{\boldsymbol{\theta}_{T_E}}$ and $P_{\boldsymbol{\theta}_T}$ do not share any parameters, the gradient simplifies to:

$$\frac{\partial}{\partial\theta_i} J_{\boldsymbol{\theta}}(D) = \begin{cases} \sum\limits_{\tau\in D}\sum\limits_{t=0}^{T_\tau-1} \frac{\partial}{\partial\theta_i}\log\pi_{\boldsymbol{\theta}}\left(a_t^\tau \mid s_t^\tau\right) + \frac{\partial}{\partial\theta_i}\log P(\boldsymbol{\theta}) & \text{if } i\in\Psi_{\neg T} \\ \sum\limits_{\tau\in D}\sum\limits_{t=0}^{T_\tau-1} \frac{\partial}{\partial\theta_i}\log P_{\boldsymbol{\theta}_T}\left(s_{t+1}^\tau \mid s_t^\tau, a_t^\tau\right) + \frac{\partial}{\partial\theta_i}\log P(\boldsymbol{\theta}) & \text{if } i\in\Psi_T \end{cases} \tag{4.10}$$

with Ψ_X being the set of all parameter indices belonging to parameter type $X \in \{R, T_E, T, P\}$ and the negation $\Psi_{\neg X}$ being the set of all parameter indices except the ones of parameter type X. By modeling the expert's belief about the environment's dynamics and the true dynamics with individual sets of parameters, the only coupling between them might exist via the log prior term. Hence, one can introduce a prior that penalizes differences between the expert's transition model and the true dynamics. Sec. 4.5 gives an overview about various priors for regularization purposes or for incorporating knowledge into learning.

We remain with deriving the gradient of the log likelihood of the policy and the dynamics. Since the model of the environment's dynamics is a design choice and problem-dependent, we exemplarily derive its gradient in the evaluation section of SERD and in the following chapter based on an application example. Therefore, we now provide the gradient of the log policy $\frac{\partial}{\partial\theta_i}\log\pi_{\boldsymbol{\theta}}\left(a_t^\tau \mid s_t^\tau\right)$, which is given by

$$\frac{\partial}{\partial\theta_i}\log\pi_{\boldsymbol{\theta}}\left(a_t^\tau \mid s_t^\tau\right) = \frac{\frac{\partial}{\partial\theta_i}\pi_{\boldsymbol{\theta}}\left(a \mid s\right)}{\pi_{\boldsymbol{\theta}}\left(a \mid s\right)}. \tag{4.11}$$

Finally, it is necessary to compute the partial derivative of the policy $\frac{\partial}{\partial\theta_i}\pi_{\boldsymbol{\theta}}\left(a \mid s\right)$ with respect to the parameters of the stochastic policy model as well as to all other parameters. Again, we express the gradient as partial derivatives with respect to the individual parameter types

$$\frac{\partial}{\partial\theta_i}\pi_{\boldsymbol{\theta}}\left(a \mid s\right) = \begin{cases} \frac{\partial}{\partial\theta_i} g(\boldsymbol{\theta}_P, Q_{\boldsymbol{\theta}}) & \text{if } i\in\Psi_P \\ \sum\limits_{\substack{s'\in S\\ a'\in A}} \frac{\partial g(\boldsymbol{\theta}_P, Q_{\boldsymbol{\theta}})(s,a)}{\partial Q_{\boldsymbol{\theta}}(s',a')}\frac{\partial}{\partial\theta_i} Q_{\boldsymbol{\theta}}(s',a') & \text{if } i\in\Psi_R\cup\Psi_{T_E} \\ 0 & \text{if } i\in\Psi_T. \end{cases} \tag{4.12}$$

However, to compute the a-posteriori log likelihood and its gradient, it is necessary to specify a model of the policy as well as the Q-function that it is based on. Furthermore, it requires both a differentiable policy model and Q-function. In general, various policy models could be applied depending on the problem that is solved. Sec. 4.4 provides derivations of some common policy models used in IRL problems. In addition, it shows the computation of the corresponding gradients, provides proofs for the derivations and compares the policy models with respect to their differences and their computation time when using SERD.

Finally, Alg. 1 summarizes the proposed gradient-based solution of the SERD problem. It assumes that an MDP is given without known parameters of the rewards, the real environment's dynamics, the expert's belief about it, or the policy model. Furthermore, it requires a set of expert demonstrations, a step size for gradient ascent, and an initial parameter vector. The SERD algorithm estimates the optimal parameters that maximize the log likelihood of the expert demonstrations. Since it is beneficial to start with reasonable parameter guesses, we propose to use an *InitialDynamicsEstimator*, which estimates an initial model of the environments dynamics directly from the observed transitions. Afterwards, the iterative gradient-based optimization starts with this initial parameterization. The first step solves the forward problem by applying *QIteration*, which computes the state-action value function. This allows extracting the *Policy* according to the chosen policy model. To perform an update step, it is necessary to compute the gradient of the objective, which is the log likelihood of the demonstrations. This requires deriving the gradient of the Q-function via *QGradient*. After that, it uses *ObjGradient* to calculate the gradient of the objective for updating the parameters.

4.4 Policies

The previous chapter introduced a unified solution to the SERD problem. However, the computation of the gradient of the policy in Eq. (4.12) requires to define a model for the stochastic policy $g(\boldsymbol{\theta}_P, Q_{\boldsymbol{\theta}})$ as well as the type of Q-function $Q_{\boldsymbol{\theta}}$. In Chap. 3, we gave an overview about learning from demonstration together with detailed reviews of several famous IRL approaches. Typically, these approaches make different assumptions on the expert's behavior and thus model different types of policies. Furthermore, their policies model behavior from different Q-functions. For example, PM IRL uses the optimal Q-function $Q^*_{\boldsymbol{\theta}}(s, a)$ according to Eq. (2.24), while MDCE IRL uses a softened one $Q^{\mathrm{s}}_{\boldsymbol{\theta}}(s, a)$ which is specified in Eq. (3.28). This section derives the gradients of the MDCE and PM policy models, which have been successfully applied to various IRL problems. In both cases, we assume that the reward function is a linear combination of parameters and features $R(s, a) = \boldsymbol{\theta}_R{}^{\mathsf{T}} \boldsymbol{f}(s, a)$.

Algorithm 1 Unified SERD algorithm

```
Require: MDP without known parameters of rewards, dynamics, or policy model
    $M \setminus \{R, P_T, P_{T_E}, g(\boldsymbol{\theta}_P, Q_{\boldsymbol{\theta}})\}$, expert demonstrations $D$, initial parameter set $\bar{\boldsymbol{\theta}}$, step
    size $\alpha : \mathbb{N}_+ \to \mathbb{R}_+$
Ensure: Approximately optimal parameters $\boldsymbol{\theta}$
  $t \leftarrow 0$
  $\boldsymbol{\theta}_0 \leftarrow \bar{\boldsymbol{\theta}}$
  $\boldsymbol{\theta}_0 \leftarrow$ InitialDynamicsEstimator($M$, $D$, $\boldsymbol{\theta}_0$)
  while not converged do
      $\boldsymbol{Q_\theta} \leftarrow$ QIteration($M$, $\boldsymbol{\theta}_t$)                         ▷ Eq. (2.24) or Eq. (3.28)
      $\boldsymbol{\pi_\theta} \leftarrow$ Policy($M$, $\boldsymbol{Q_\theta}$, $g(\boldsymbol{\theta}_P, Q_{\boldsymbol{\theta}})$)          ▷ Eq. (2.26) or Eq. (3.30)
      $\nabla \boldsymbol{Q_\theta} \leftarrow$ QGradient($M$, $\boldsymbol{Q_\theta}$, $\boldsymbol{\pi_\theta}$, $\boldsymbol{\theta}_t$)            ▷ Eq. (4.21) or Eq. (4.66)
      $\nabla J_{\boldsymbol{\theta}} \leftarrow$ ObjGradient($M$, $D$, $\boldsymbol{Q_\theta}$, $\boldsymbol{\pi_\theta}$, $d\boldsymbol{Q_\theta}$)                  ▷ Eq. (4.9)
      $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \alpha(t)\nabla L_{\boldsymbol{\theta}}$
      $t \leftarrow t + 1$
  end while
```

4.4.1 Maximum Discounted Causal Entropy SERD (MDCE SERD)

Since humans are rarely able to demonstrate entirely optimal behavior, approaches are necessary that can cope with stochastic or noisy demonstrations. However, it is often unknown how humans behave and how their policy can be appropriately modeled. Therefore, IRL and SERD require robust solutions that learn generalizable representations that do not overestimate individual trajectories, especially if the expert demonstration set is small. In Sec. 3.2.3, we gave an overview on MDCE IRL, which is an infinite horizon variant of MCE IRL. The approaches by Ziebart *et al.*[119] as well as Bloem and Bambos [16] achieve robustness by learning policies with maximum causal entropy under constraints to match features between the model and the expert behavior. This yields a distribution that is as uninformative as possible, while guaranteeing to satisfy the desired constraints. Consequently, it decouples the learned model from the demonstrations by only matching certain aspects and not trying to copy the observed trajectories. Hence, the solutions to their maximum causal entropy problems are gradient-based methods with a gradient that is the difference between the expert's empirical feature expectation and the expected features of the model $\frac{\partial\lambda(P,\boldsymbol{\theta},\eta)}{\partial\boldsymbol{\theta}} = \widetilde{\boldsymbol{\phi}} - \sum_{\tau\in\mathcal{T}} P_{\boldsymbol{\theta}}(\tau)\boldsymbol{\phi}(\tau)$. It should be noticed that these solutions only require the expert's empirical feature expectation to be known, while it does not further use the expert demonstrations for optimization. However, they also show that the maximum likelihood solution, which directly optimizes the log likelihood of the observed demonstrations, is identical to the MDCE solution with a feature matching gradient, if the transition model is available and the number of expert demonstrations approaches infinity. This indicates that in the limit the ML solution may be as robust as the MDCE solution. As the unified SERD approach formulates a MAP problem, which incorporates the likelihood of the demonstrations, the question arises whether it still learns robust and accurate models, when modeling behavior with an MDCE policy. In this section, we derive a SERD solution based on the MDCE policy, which we call MDCE SERD. It extends the MDCE IRL problem, by simultaneously learning the parameters of a model of the environment's dynamics and the expert's belief about it. In the following, we derive a solution for the MDCE SERD problem, analyze this solution, and provide proofs for the differentiability and convergence.

In MDCE IRL, the policy model $\pi = g(Q^s_{\boldsymbol{\theta}})$ takes the form of a Boltzmann distribution over soft Q-values from Eq. (3.28) with no additional policy parameters:

$$\pi_{\boldsymbol{\theta}}(a \mid s) = g(Q^s_{\boldsymbol{\theta}})(s,a) = \frac{\exp(Q^s_{\boldsymbol{\theta}}(s,a))}{\sum_{a'\in A}\exp\left(Q^s_{\boldsymbol{\theta}}(s,a')\right)}. \tag{4.13}$$

Deriving the SERD gradient Eq. (4.12) of the MDCE policy model [40] requires to compute the partial derivative of $g(Q^s_{\boldsymbol{\theta}})(s,a)$ with respect to $Q^s_{\boldsymbol{\theta}}(s,a)$:

$$\frac{\partial g(Q^s_{\boldsymbol{\theta}})(s,a)}{\partial Q^s_{\boldsymbol{\theta}}(s',a')} = \begin{cases} \frac{\exp\left(Q^s_{\boldsymbol{\theta}}(s,a)\right)\left(\sum_{\alpha\in A\setminus a}\exp\left(Q^s_{\boldsymbol{\theta}}(s,\alpha)\right)\right)}{\left(\sum_{\alpha\in A}\exp\left(Q^s_{\boldsymbol{\theta}}(s,\alpha)\right)\right)^2} & \text{if } s=s', a=a' \\ -\frac{\exp\left(Q^s_{\boldsymbol{\theta}}(s,a)\right)\exp\left(Q^s_{\boldsymbol{\theta}}(s,a')\right)}{\left(\sum_{\alpha\in A}\exp\left(Q^s_{\boldsymbol{\theta}}(s,\alpha)\right)\right)^2} & \text{if } s=s', a\neq a' \\ 0 & \text{otherwise.} \end{cases} \tag{4.14}$$

When substituting the partial derivative of $g(Q_{\boldsymbol{\theta}})(s,a)$ with respect to $Q_{\boldsymbol{\theta}}(s,a)$ into Eq. (4.12), the partial derivative of the policy $\frac{\partial}{\partial\theta_i}\pi_{\boldsymbol{\theta}}\left(a \mid s\right)$ further simplifies to the weighted and expected gradient of the Q-function:

$$
\begin{aligned}
&\frac{\partial}{\partial \theta_i} \pi_{\boldsymbol{\theta}}\left(a \mid s\right) && (4.15)\\
&\quad= \sum_{\substack{s' \in S \\ a' \in A}} \frac{\partial g(\boldsymbol{\theta}_P, Q_{\boldsymbol{\theta}})(s,a)}{\partial Q_{\boldsymbol{\theta}}(s',a')} \frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s',a') && (4.16)\\
&\quad= \pi_{\boldsymbol{\theta}}(a \mid s) \left[\sum_{a' \in A \setminus a} \pi_{\boldsymbol{\theta}}(a' \mid s) \frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s,a) - \sum_{a' \in A \setminus a} \pi_{\boldsymbol{\theta}}(a' \mid s) \frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s,a') \right] && (4.17)\\
&\quad= \pi_{\boldsymbol{\theta}}(a \mid s) \left[\frac{\partial}{\partial \theta_i} Q^s_{\boldsymbol{\theta}}(s,a) - \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s)} \left[\frac{\partial}{\partial \theta_i} Q^s_{\boldsymbol{\theta}}(s,a') \right] \right]. && (4.18)
\end{aligned}
$$

It follows that the gradient of the policy depends on the gradient of the state-action value function $\frac{\partial}{\partial \theta_i} Q^s_{\boldsymbol{\theta}}(s,a)$. Furthermore, it requires to derive the converged soft Q- and soft value-function for extracting the stochastic policy, which is necessary for computing the expected gradient $\mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s)} \left[\frac{\partial}{\partial \theta_i} Q^s_{\boldsymbol{\theta}}(s,a') \right]$.

Finally, the partial derivative of the soft Q-function from Eq. (3.28) with respect to θ_i results in:

$$
\begin{aligned}
\frac{\partial}{\partial \theta_i} Q^s_{\boldsymbol{\theta}}(s,a) =& \frac{\partial}{\partial \theta_i} \boldsymbol{\theta}_R^{\intercal} \boldsymbol{f}(s,a) && (4.19)\\
&+ \gamma \sum_{s' \in S} \left[\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_{T_E}}\left(s' \mid s,a\right) \right) V_{\boldsymbol{\theta}}\left(s'\right) \right] && (4.20)\\
&+ \gamma \sum_{s' \in S} \left[P_{\boldsymbol{\theta}_{T_E}}\left(s' \mid s,a\right) \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s')} \left[\frac{\partial}{\partial \theta_i} Q^s_{\boldsymbol{\theta}}(s',a') \right] \right]. && (4.21)
\end{aligned}
$$

As the MDCE policy does not introduce additional policy parameters, we take the partial derivative with respect to the three remaining individual parameter types (the feature weights, the expert's dynamics parameters, and the parameters of the true environment's dynamic) yields a more simple solution that is easier to interpret:

$$
\frac{\partial}{\partial \theta_i} Q^s_{\boldsymbol{\theta}}(s,a) =
\begin{cases}
f_i(s,a) + \gamma \sum_{s' \in S} \Big[P_{\boldsymbol{\theta}_{T_A}}\left(s' \mid s,a\right) & \text{if } i \in \Psi_R \\
\quad \cdot \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s')} \left[\frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s',a') \right] \Big] & \\
\gamma \sum_{s' \in S} \left[\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_{T_A}}\left(s' \mid s,a\right) \right) V_{\boldsymbol{\theta}}(s') \right. & \text{if } i \in \Psi_{T_E} \\
\quad \left. + P_{\boldsymbol{\theta}_{T_A}}\left(s' \mid s,a\right) \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s')} \left[\frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s',a') \right] \right] & \\
0 & \text{if } i \in \Psi_T.
\end{cases}
\qquad (4.22)
$$

This soft Q-gradient can be solved exactly, as it is a linear equation system. However, solving large linear equation systems exactly (e.g. via LU decomposition [32]) can be computationally expensive. Instead, it is often beneficial to choose an iterative approach [96] (see Section 4.4.3). When interpreting Eq. (4.21) as an operator and recursively applying it to an arbitrary initial gradient, it will converge to the true gradient, as it is a fixed point equation. If such an approximate solution is sufficient, it can highly increase the computational efficiency. Furthermore, Alg. 1 requires computing the Q-gradient iteratively with slightly changed parameters in every pass of the inner loop. If the new Q-gradient computation is initialized with the result from the preceding run, it might further reduce the required number of Q-gradient iterations to convergence to a good solution.

In the following, we will prove the correctness and convergence of the proposed algorithm. The gradient-based optimization method requires the policy to be differentiable. However, it depends on the converged soft Q-function. Therefore, we will show that the soft Q-iteration is a fixed point equation and that the converged soft Q-function is differentiable with respect to all parameters. Afterwards, we will show that the provided soft Q-gradient iteration is a contraction mapping with only one fixed point. Consequently, the proposed gradient-based method yields valid gradients for optimizing the log likelihood of the demonstrations.

Soft Q-iteration is a Contraction Mapping

We need to show that the soft Q-iteration is a fixed point iteration with only one fixed point, since this is a requirement of our algorithm. Bloem *et al.*[16] have shown that the soft value iteration operator is a contraction mapping. Hence, we need to prove that the same holds for the soft Q-iteration operator $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q})$. Therefore, we adjust their proof to be valid for the Q-iteration.

The soft Q-iteration operator is defined as

$$T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q})(s,a) = \boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s,a) + \gamma \sum_{s' \in S} \left[P_{\boldsymbol{\theta}_{T_A}}(s' \mid s,a) \operatorname*{softmax}_{a' \in A}\left(Q\left(s',a'\right)\right)\right] \tag{4.23}$$

with the function

$$\operatorname*{softmax}_{x_i \in \boldsymbol{x}}(x_i) = \log\left(\sum_{i=1}^{N} \exp\left(x_i\right)\right). \tag{4.24}$$

We will begin with deriving proofs for necessary auxiliary definitions and lemmata. In order to argue about the monotonicity of multidimensional functions, we introduce a partial order on $\mathbb{R}^{A\times B}$. Then, we derive a required property of the $\operatorname{softmax}$ function first and prove the monotonicity of the operator $T^{soft}_{\boldsymbol{\theta}}(\boldsymbol{Q})(s,a) : \mathbb{R}^{|S|\times|A|} \to \mathbb{R}^{|S|\times|A|}$ with respect to the partial order afterwards.

Definition 4.4.1. *For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{A\times B}$ with $A, B \in \mathbb{N}^+$, the partial order $\preceq$ is defined as $\boldsymbol{x} \preceq \boldsymbol{y} \Leftrightarrow \forall a \in A, b \in B : x_{a,b} \leq y_{a,b}$.*

Lemma 4.4.2. *The* $\operatorname{softmax}$ *function has the property that for any $\boldsymbol{x} \in \mathbb{R}^N$ and $d \in \mathbb{R}$ it holds that $\operatorname*{softmax}_{x_i\in\boldsymbol{x}}(x_i + d) = \operatorname*{softmax}_{x_i\in\boldsymbol{x}}(x_i) + d$.*

Proof. It is easy to show that the $\operatorname{softmax}$ function allows to extract additive constants d from all elements x_i:

$$\begin{aligned}\operatorname*{softmax}_{x_i\in\boldsymbol{x}}(x_i + d) &= \log\left(\sum_{i=1}^{N} \exp\left(x_i + d\right)\right) && (4.25)\\ &= \log\left(\exp\left(d\right)\sum_{i=1}^{N} \exp\left(x_i\right)\right) && (4.26)\\ &= \log\left(\sum_{i=1}^{N} \exp\left(x_i\right)\right) + \log\left(\exp\left(d\right)\right) && (4.27)\\ &= \log\left(\sum_{i=1}^{N} \exp\left(x_i\right)\right) + d && (4.28)\\ &= \operatorname*{softmax}_{x_i\in\boldsymbol{x}}(x_i) + d && (4.29)\end{aligned}$$

□

Lemma 4.4.3. *The soft Q-iteration operator $T^{soft}_{\boldsymbol{\theta}}(\boldsymbol{Q})(s,a)$ is monotone, satisfying $\forall \boldsymbol{Q}_m, \boldsymbol{Q}_n \in \mathbb{R}^{|S|\times|A|} : \boldsymbol{Q}_m \preceq \boldsymbol{Q}_n \to T^{soft}_{\boldsymbol{\theta}}(\boldsymbol{Q}_m) \preceq T^{soft}_{\boldsymbol{\theta}}(\boldsymbol{Q}_n)$.*

Proof. The partial derivative of the $T^{soft}_{\boldsymbol{\theta}}(\boldsymbol{Q})(s,a)$ with respect to a single value $Q(s_i, a_i)$ is

$$\frac{\partial}{\partial Q(s_i,a_i)} T^{soft}_{\boldsymbol{\theta}}(\boldsymbol{Q})(s,a) = \gamma P_{\boldsymbol{\theta}_{T_A}}\left(s_i \mid s,a\right) \frac{\exp\left(Q\left(s_i,a_i\right)\right)}{\sum_{a_j\in A}\exp\left(Q\left(s_i,a_j\right)\right)}. \tag{4.30}$$

From the definition of the MDP it follows that $\gamma \in [0,1)$ and the probability distribution $P_{\boldsymbol{\theta}_{T_A}}\left(s_i \mid s,a\right) \in [0,1]$. As $\forall x_i \in \mathbb{R} : \exp(x_i) \in (0,+\infty)$, all terms of the partial derivative $\frac{\partial}{\partial Q(s_i,a_i)} T^{soft}_{\boldsymbol{\theta}}(\boldsymbol{Q})(s,a)$ are positive or zero, which finishes the proof that $\frac{\partial}{\partial Q(s_i,a_i)} T^{soft}_{\boldsymbol{\theta}}(\boldsymbol{Q})(s,a) \geq 0$. □

Based on Lemma 4.4.2 and 4.4.3, we can derive the proof that the soft Q-iteration is a contraction mapping with only one fixed point. Therefore, we adopt the proof of Bloem and Bambos [16] for the value iteration and adjust it, such that it applies for the Q-iteration.

Theorem 4.4.4. *The soft Q-iteration operator $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q})(s,a)$ is a contraction mapping with only one fixed point. Therefore, it is Lipschitz continuous $||T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_m) - T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n)||_\infty \leq L||\boldsymbol{Q}_m - \boldsymbol{Q}_n||_\infty$ for all $\boldsymbol{Q}_m, \boldsymbol{Q}_n \in \mathbb{R}^{|S|\times|A|}$ with a Lipschitz constant $L \in [0,1)$.*

Proof. Consider $\boldsymbol{Q}_m, \boldsymbol{Q}_n \in \mathbb{R}^{|S|\times|A|}$. There exists a distance d under the supremum norm, for which $\exists d \in \mathbb{R}_0^+ : ||\boldsymbol{Q}_m - \boldsymbol{Q}_n||_\infty = d$ holds and therefore

$$-d\mathbf{1} \preceq \boldsymbol{Q}_m - \boldsymbol{Q}_n \preceq d\mathbf{1} \tag{4.31}$$

with $\mathbf{1} = (1)_{k,l}$, where $1 \leq k \leq |S|, 1 \leq l \leq |A|$. Since d bounds the components of the vector difference $\boldsymbol{Q}_m - \boldsymbol{Q}_n$, $\boldsymbol{Q}_m \preceq \boldsymbol{Q}_n + d\mathbf{1}$ and $\boldsymbol{Q}_n \preceq \boldsymbol{Q}_m + d\mathbf{1}$ hold due to the symmetric definition. In both cases, the monotonicity condition of Lemma 4.4.3 is satisfied, which allows for the following inequality: $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_m) \preceq T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n + d\mathbf{1})$. By applying Lemma 4.4.2, it follows that $\forall s \in S, a \in A$

$$\begin{aligned}
&T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_m)(s,a) && (4.32)\\
&\quad\leq T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n + d\mathbf{1})(s,a) && (4.33)\\
&\quad= \boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s,a) + \gamma \sum_{s'\in S}\left[P_{\boldsymbol{\theta}_{T_A}}(s' \mid s,a) \operatorname*{softmax}_{a'\in A}\left(Q\left(s',a'\right)+d\right)\right] && (4.34)\\
&\quad= \boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s,a) + \gamma \sum_{s'\in S}\left[P_{\boldsymbol{\theta}_{T_A}}(s' \mid s,a) \operatorname*{softmax}_{a'\in A}\left(Q\left(s',a'\right)\right)+d\right] && (4.35)\\
&\quad= \boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s,a) + \gamma \sum_{s'\in S}\left[P_{\boldsymbol{\theta}_{T_A}}(s' \mid s,a) \operatorname*{softmax}_{a'\in A}\left(Q\left(s',a'\right)\right)\right]+\gamma d && (4.36)\\
&\quad= T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n)(s,a) + \gamma d && (4.37)
\end{aligned}$$

In matrix notation, this results in $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_m) \preceq T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n) + \gamma d\mathbf{1}$. The symmetric definition of $-d\mathbf{1} \preceq \boldsymbol{Q}_m - \boldsymbol{Q}_n \preceq d\mathbf{1}$, implies that $\boldsymbol{Q}_n \preceq \boldsymbol{Q}_m + d$ and consequently it follows that $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n) \preceq T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_m) + \gamma d\mathbf{1}$. To finish the proof, it is required to show that the soft Q-iteration operator is Lipschitz continuous with $L \in [0,1)$. Combining the related inequations of the operator yields

$$-\gamma d\mathbf{1} \preceq \; T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_m) - T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n) \quad \preceq \gamma d\mathbf{1} \tag{4.38}$$

$$||T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_m) - T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n)||_\infty \leq \gamma d \tag{4.39}$$

$$||T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_m) - T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}_n)||_\infty \leq \gamma ||\boldsymbol{Q}_m - \boldsymbol{Q}_n||_\infty . \tag{4.40}$$

This proves that the soft Q-iteration operator $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q})$ is Lipschitz continuous with a Lipschitz constant $L = \gamma$ and $\gamma \in [0, 1)$, resulting in a contraction mapping. As this holds for the whole input space of $\mathbb{R}^{|S|\times|A|}$, two points would always contract, so there cannot exist two fixed points. □

Differentiability of the converged soft Q-function

Theorem 4.4.5. *The converged soft Q-function $\tilde{Q}^s(s, a)$ is differentiable with respect to the parameters $\boldsymbol{\theta}$.*

Proof. Previously, we have proven that the soft Q-iteration operator $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}^s) : \mathbb{R}^{S\times A} \mapsto \mathbb{R}^{S\times A}$ is a fixed-point equation with

$$T_{\boldsymbol{\theta}}^{soft}(Q^s(s', a'))[s, a] = \boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s, a) + \gamma \sum_{s' \in S} \left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) \log \left(\sum_{a' \in A} \exp(Q^s(s', a')) \right) \right] .$$

It converges to a fixed-point $\tilde{\boldsymbol{Q}}^s$ given $\gamma \in [0, 1)$, by repeatedly applying the operator to an arbitrary initial $\boldsymbol{Q}_0^s$. If the transition model $P_{\boldsymbol{\theta}_{T_E}}$ is differentiable with respect to $\boldsymbol{\theta}$, then $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}^s)$ is differentiable with respect to both $\boldsymbol{Q}^s$ and $\boldsymbol{\theta}$, since it is a composition of differentiable functions. Then, we can apply the implicit function theorem [50] to compute the derivative $\frac{\partial}{\partial \boldsymbol{\theta}} \tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}^s$, which is given by the equation

$$T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}^s)(s, a) - Q^s(s, a) = 0. \tag{4.41}$$

According to the theorem, the derivative $\frac{\partial}{\partial \boldsymbol{\theta}} \tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}^s$ exists, if the Jacobian $\frac{\partial}{\partial \boldsymbol{Q}^s}[T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}^s) - \boldsymbol{Q}^s]$ is invertible at $\tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}^s$. If this is the case, the derivative is given by

$$\frac{\partial}{\partial \boldsymbol{\theta}} \tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}^s = \left(\frac{\partial}{\partial \boldsymbol{Q}^s} [T_{\boldsymbol{\theta}}^{soft}(.) - .] \right)^{-1} \frac{\partial}{\partial \boldsymbol{\theta}} T_{\boldsymbol{\theta}}^{soft}(.) \; (\tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}^s). \tag{4.42}$$

The partial derivative of the operator $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}^s)$ is

$$\frac{\partial}{\partial Q^s(s_i,a_i)} T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}^s)(s,a) = \gamma P_{\boldsymbol{\theta}_{T_E}}(s_i \mid s,a) \frac{\exp\left(Q^s\left(s_i,a_i\right)\right)}{\sum_{a_j \in A} \exp\left(Q^s\left(s_i,a_j\right)\right)}. \tag{4.43}$$

Therefore, the Jacobian of $T_{\boldsymbol{\theta}}^{soft}(\boldsymbol{Q}^s) - \boldsymbol{Q}^s$ is

$$\frac{\partial}{\partial \boldsymbol{Q}^s}[T_{\boldsymbol{\theta}}^{soft}(\tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}^s) - \tilde{\boldsymbol{Q}}_{\theta}^s]([s,a],[s',a']) \tag{4.44}$$

$$= -\delta(a'=a, s'=s) + \gamma P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a) \tag{4.45}$$

$$\cdot \frac{1}{\sum_{a'' \in A} \exp(Q^s(s',a''))} \exp(Q^s(s',a')) \tag{4.46}$$

$$= -\delta(a'=a, s'=s) + \gamma \underbrace{\pi(a' \mid s') P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a)}_{M_{[s,a],[s',a']}}. \tag{4.47}$$

It holds $1 > \gamma \geq ||\gamma \boldsymbol{M}||_\infty$ in the $L\infty$ induced matrix-norm defined as $||A||_\infty := \max_x \frac{|Ax|_\infty}{|x|_\infty} = \max_i \sum_j |A_{i,j}|$, as

$$\max_{[s,a]} \sum_{[s',a']} |\gamma M_{[s,a],[s',a']}| = \max_{[s,a]} \sum_{[s',a']} |\gamma \pi(a' \mid s') P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a)| = \gamma. \tag{4.48}$$

It follows that $(\gamma \boldsymbol{M} - I)^{-1}$ exists and is given by the *Neuman* operator-series $-\sum_{i=0}^{\infty} (\gamma \boldsymbol{M})^i$. [114]. The fact that the Jacobian is invertible proves that the derivative of the converged soft Q-function $\frac{\partial}{\partial \boldsymbol{\theta}} \tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}^s$ with respect to $\boldsymbol{\theta}$ exists. □

Monotonicity of the soft Q-gradient operator

We define the soft Q-gradient operator $U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}) \in \mathbb{R}^{|S| \times |A| \times |\Psi|}$ as:

$$U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi})(s,a,i) = \frac{\partial}{\partial \theta_i} \boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s,a) \tag{4.49}$$

$$+ \gamma \sum_{s' \in S} \left[\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a) \right) V_{\boldsymbol{\theta}}^{soft}(s') \right] \tag{4.50}$$

$$+ \gamma \sum_{s' \in S} \left\{ P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a) \cdot \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s')} \left[\Phi(s',a',i) \right] \right\}, \tag{4.51}$$

for all $s \in S, a \in A$, parameter dimensions $i \in \Psi$ with $\Psi = \{1, \ldots, \dim(\boldsymbol{\theta})\}$ and the gradient $\Phi(s,a,i) = \frac{\partial}{\partial \theta_i} Q_\theta^s(s,a)$. In order to argue about the monotonicity of multidimensional functions, we introduce a partial order on $\mathbb{R}^{A \times B \times C}$.

Definition 4.4.6. *For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{A \times B \times C}$ with $A, B, C \in \mathbb{N}^+$, the partial order $\preceq$ is defined as $\boldsymbol{x} \preceq \boldsymbol{y} \Leftrightarrow \forall a \in A, b \in B, c \in C : x_{a,b,c} \leq y_{a,b,c}$.*

Lemma 4.4.7. *The soft Q-gradient iteration operator $U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi})(s, a, i)$ is monotone, satisfying $\forall \boldsymbol{\Phi}_m, \boldsymbol{\Phi}_n \in \mathbb{R}^{|S| \times |A| \times |\Psi|} : \boldsymbol{\Phi}_m \preceq \boldsymbol{\Phi}_n \rightarrow U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m) \preceq U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n)$.*

Proof. The partial derivative of $U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi})(s, a, i)$ with respect to a single value $\Phi(s_k, a_k, k)$ is

$$\frac{\partial}{\partial \Phi(s_k, a_k, k)} U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi})(s, a, i) \tag{4.52}$$

$$= \frac{\partial}{\partial \Phi(s_k, a_k, k)} \gamma \sum_{s' \in S} \left\{ P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) \, \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s')} \left[\Phi(s', a', i) \right] \right\} \tag{4.53}$$

$$= \gamma P_{\boldsymbol{\theta}_{T_E}}(s_k \mid s, a) \, \pi_{\boldsymbol{\theta}}(a_k \mid s_k) \, \delta(i = k). \tag{4.54}$$

Since the definition of the MDP ensures that $\gamma \in [0, 1)$, the probability distributions $\pi_{\boldsymbol{\theta}}(a_k \mid s_k) \in [0, 1]$, as well as $P_{\boldsymbol{\theta}_{T_E}}(s_i \mid s, a) \in [0, 1]$, and $\delta(\cdot) \in \{0, 1\}$, all terms of the partial derivative $\frac{\partial}{\partial \Phi(s_k, a_k, k)} U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi})(s, a, i)$ are positive or zero. Hence, it follows that $\frac{\partial}{\partial \Phi(s_k, a_k, k)} U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi})\,(s, a, i) \geq 0$. □

Soft Q-gradient operator is a fixed point equation

Theorem 4.4.8. *The soft Q-gradient iteration operator $U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi})(s, a, i)$ is a contraction mapping with only one fixed point. Therefore, it is Lipschitz continuous $||U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m) - U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n)||_\infty \leq L||\boldsymbol{\Phi}_m - \boldsymbol{\Phi}_n||_\infty$ for all $\boldsymbol{\Phi}_m, \boldsymbol{\Phi}_n \in \mathbb{R}^{|S| \times |A| \times |\Psi|}$ with a Lipschitz constant $L \in [0, 1)$.*

Proof. Consider $\boldsymbol{\Phi}_m, \boldsymbol{\Phi}_n \in \mathbb{R}^{|S| \times |A| \times |\Psi|}$. There exists a distance d for which $\exists d \in \mathbb{R}_0^+ : ||\boldsymbol{\Phi}_m - \boldsymbol{\Phi}_n||_\infty = d$ holds and therefore $-d\mathbf{1} \preceq \boldsymbol{\Phi}_m - \boldsymbol{\Phi}_n \preceq d\mathbf{1}$ with $\mathbf{1} = (1)_{k,l,m}$, where $1 \leq k \leq |S|, 1 \leq l \leq |A|, 1 \leq m \leq |\Psi|$. If d is added to every element of $\boldsymbol{\Phi}_n$, it is guaranteed that $\boldsymbol{\Phi}_m \preceq \boldsymbol{\Phi}_n + d\mathbf{1}$. This satisfies the monotonicity condition of Lemma

4.4.7: $U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m) \preceq U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n + d\mathbf{1})$. Then, it follows $\forall s \in S, a \in A, i \in \Psi$:

$$
\begin{aligned}
& U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m)(s,a,i) && (4.55)\\
&\quad \leq U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n + d\mathbf{1})(s,a,i) && (4.56)\\
&\quad = \frac{\partial}{\partial \theta_i}\boldsymbol{\theta}_R^{\intercal}\boldsymbol{f}(s,a) + \gamma \sum_{s' \in S}\left[\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a)\right) V_{\boldsymbol{\theta}}(s')\right] && (4.57)\\
&\qquad + \gamma \sum_{s' \in S}\left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a)\, \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s')}\left[\Phi(s',a',i) + d\right]\right] && (4.58)\\
&\quad = \frac{\partial}{\partial \theta_i}\boldsymbol{\theta}_R^{\intercal}\boldsymbol{f}(s,a) + \gamma \sum_{s' \in S}\left[\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a)\right) V_{\boldsymbol{\theta}}(s')\right] && (4.59)\\
&\qquad + \gamma \sum_{s' \in S}\left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a)\, \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a' \mid s')}\left[\Phi(s',a',i)\right]\right] + \gamma d && (4.60)\\
&\quad = U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n)(s,a,i) + \gamma d && (4.61)
\end{aligned}
$$

In the previous notation, this yields $U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m) \preceq U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n) + \gamma d\mathbf{1}$. Since d is defined symmetrically, it equally holds that $\boldsymbol{\Phi}_n \preceq \boldsymbol{\Phi}_m + d$ and $U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n) \preceq U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m) + \gamma d\mathbf{1}$. Then, the Lipschitz continuity of the soft Q-gradient iteration follows from combining these inequations:

$$
\begin{aligned}
-\gamma d\mathbf{1} \preceq \quad & U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m) - U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n) \quad \preceq \gamma d\mathbf{1}\\
& ||U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m) - U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n)||_\infty \leq \gamma d\\
& ||U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_m) - U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi}_n)||_\infty \leq \gamma ||\boldsymbol{\Phi}_m - \boldsymbol{\Phi}_n||_\infty
\end{aligned}
$$

This proves that the soft Q-gradient iteration operator $U_{\boldsymbol{\theta}}^{soft}(\boldsymbol{\Phi})(s,a,i)$ is Lipschitz continuous with a Lipschitz constant $L = \gamma$ with $\gamma \in [0,1)$, resulting in a contraction mapping. As this holds for the whole input space $\mathbb{R}^{|S| \times |A| \times |\Psi|}$, two points will always contract. As a consequence, there cannot exist two fixed points. □

4.4.2 Policy Matching SERD (PM SERD)

In the previous section, we have derived the MDCE SERD approach, which models a maximum entropy policy that matches feature expectations, yielding a maximally uninformative distribution. This is reasonable, since the real expert's policy model is often unknown and more informative models would more likely overfit. However, if it is possible to make assumptions about the expert's decision making, the expert's behavior could be explained more accurately. In general, many different types of policies are applicable. Since PM IRL has proven successful in several applications, this section introduces Policy Matching SERD (PM SERD) [39], which is based on Apprenticeship Learning using IRL and Gradient Methods by Neu and Szepesvári [74] (see Sec. 3.2.1). Similar to the previous approach, we model behavior by a Boltzmann distribution over optimal Q-values from Eq. (2.24). This implies that the expert is able to compute the optimum Q-values of the available actions, but he is not able to execute them accurately. In addition, we preserve the *temperature* θ_P as a parameter of the policy, which scales its randomness. This result in the following policy model of PM SERD:

$$\pi_{\boldsymbol{\theta}}(a \mid s) = g(Q)(s,a) = \frac{\exp(\frac{1}{\theta_P} Q_{\boldsymbol{\theta}}(s,a))}{\sum_{a' \in A} \exp\left(\frac{1}{\theta_P} Q_{\boldsymbol{\theta}}(s,a')\right)}. \tag{4.62}$$

It should be noted that other policy models could be introduced here. For example, the Boltzmann distribution produces nonzero probabilities for all actions, that have a Q-value $Q_{\boldsymbol{\theta}}(s,a) > -\infty$. However, this model assumption may be inappropriate, if experts never choose useless or bad actions. The partial derivative of this policy $\frac{\partial}{\partial \theta_i} \pi_{\boldsymbol{\theta}}\left(a_t^\tau \mid s_t^\tau\right)$ is

$$\frac{\partial}{\partial \theta_i} \pi_{\boldsymbol{\theta}}\left(a \mid s\right) = \begin{cases} \pi_{\boldsymbol{\theta}}\left(a \mid s\right) \frac{1}{\theta_P} \left[\frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s,a) - \mathbb{E}_{\pi_{\boldsymbol{\theta}}(a'|s)}\left[\frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s,a')\right]\right] & \text{if } i \in \Psi_{\neg P} \\ \pi_{\boldsymbol{\theta}}\left(a \mid s\right) \frac{1}{\theta_P{}^2} \left[\mathbb{E}_{\pi_{\boldsymbol{\theta}}(a'|s)}\left[Q_{\boldsymbol{\theta}}(s,a')\right] - Q_{\boldsymbol{\theta}}(s,a)\right] & \text{if } i \in \Psi_P \end{cases} \tag{4.63}$$

with the set of policy parameter indices Ψ_P. It is very similar to the solution of MDCE SERD, but it differs by the additional parameter θ_P, by the corresponding partial derivative, and by depending on the optimal Q-function. Exactly as in the previous approach, the gradient of the policy depends on the gradient of the converged state-action value function $\frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s,a)$. However, the $\max$-function is not differentiable everywhere. Therefore, the resulting gradient can be a sub-derivative. In the following, we will provide the partial derivative of the Q-function with respect to θ_i, which we call Q-gradient.

$$\frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s,a) = \frac{\partial}{\partial \theta_i} \boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s,a) \tag{4.64}$$

$$+ \gamma \sum_{s' \in S} \left[\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_{T_E}}\left(s' \mid s,a\right)\right) V_{\boldsymbol{\theta}}\left(s'\right)\right] \tag{4.65}$$

$$+ \gamma \sum_{s' \in S} \left[P_{\boldsymbol{\theta}_{T_E}}\left(s' \mid s,a\right) \frac{\partial}{\partial \theta_i} Q_{\boldsymbol{\theta}}(s', \pi_{\boldsymbol{\theta}}^*\left(s'\right))\right] \tag{4.66}$$

This gradient shares similarities with the gradient of the approach from Neu and Szepesvári [74] and extends it by the partial derivative of $Q_{\boldsymbol{\theta}}(s, a)$ with respect to the transition model parameters. The Q-gradient is identical to the soft Q-gradient of MDCE SERD, when using a deterministic, optimal policy to compute the expectation of Eq. (4.21). This is especially interesting, since the formulation of the traditional Q-function differs from the softened one. In the following, we will provide proofs for the correctness and convergence of PM SERD.

Subdifferentiability of the converged Q-function

Bellman [13] has shown that the value iteration [97] is a fixed point equation. Hence, the Q-iteration operator $T_{\boldsymbol{\theta}}(\boldsymbol{Q}) : \mathbb{R}^{S \times A} \mapsto \mathbb{R}^{S \times A}$ is a fixed-point equation, too, with

$$T_{\boldsymbol{\theta}}(Q(s', a'))[s, a] = \boldsymbol{\theta}_R^{\intercal} \boldsymbol{f}(s, a) + \gamma \sum_{s' \in S} \left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) \max_{a' \in A} Q(s', a') \right].$$

It converges to a fixed-point $\boldsymbol{Q}^*$ given $\gamma \in [0, 1)$, by repeatedly applying the operator to an arbitrary initial $\boldsymbol{Q}_0$. However, the mapping from parameters $\boldsymbol{\theta}$ to the fixed-point $\boldsymbol{Q}^*$ is non-differentiable, as the operator assumes a greedy strategy for choosing actions Eq. (2.26). For computing derivatives of the converged Q-function $\tilde{\boldsymbol{Q}}^*$, Neu and Szepesvári [74] show that the 'derivatives can be calculated almost everywhere'. Therefore, they use the concept of subdifferentials:

Definition 4.4.9 (Fréchet Subdifferentials)**.** *Let W be a Banach space, W^* be its topological dual. The Fréchet subdifferential of $f : W \to \mathbb{R}$ at $w \in W$, denoted by $\partial^- f(w)$ is the set of $w^* \in W^*$ such that*

$$\liminf_{h \to 0, h \neq 0} \|h\|^{-1} \left[f(w + h) - f(w) - \langle w^*, h \rangle \right] \geq 0. \tag{4.67}$$

Consequently, the Fréchet subdifferential is the set of all subgradients that fulfill the criterion in Eq. (4.67). In addition, Neu and Szepesvári [74] derive some properties of functions $f(w)$ that involve a maximum or a linear combination, such as in the operator $T_{\boldsymbol{\theta}}(\boldsymbol{Q})$, based on the analysis of Kruger [53].

Proposition 4.4.10. *Let $\{f_i\}_{i \in I}$ be a family of real-valued functions defined over W and let $f(w) = \max_{i \in I} f_i(w)$. Then, if $w^* \in \partial^- f_i(w)$ and $f_i(w) = \max f(w)$, then $w^* \in \partial^- f(w)$. If $f_1, f_2 : W \to \mathbb{R}, \alpha_1, \alpha_2 \geq 0$, then $\alpha_1 \partial^- f_1 + \alpha_2 \partial^- f_2 \subset \partial^- (\alpha_1 f_1 + \alpha_2 f_2)$.*

This proposition shows that the Fréchet subdifferential contains all subgradients $w^* \in \partial^- f_i(w)$ of the individual components $f_i(w)$ as long as they fulfill $f_i(w) = \max f(w)$. As $T_{\boldsymbol{\theta}}(\boldsymbol{Q})$ contains a maximum, this also holds for the operator itself as well as its fixed-point $\boldsymbol{Q}^*$. For proving that a subdifferential exists for the fixed point, Neu and Szepesvári [74] extract a proof by Penot [77].

Proposition 4.4.11. *Assume that $\{f_n\}_n$ is a sequence of real-valued functions over W which converge to some function f pointwise. Let $w \in W, w_n^* \in \partial^- f_n(w)$ and assume that $\{w_n^*\}$ is weak*-convergent to w^* and is bounded. Then, $w^* \in \partial^- f(w)$, if the following holds at w: For any $\epsilon > 0$, there exists some index $N > 0$ and a real number $\delta > 0$ such that for any $n \geq N, h \in B_W(0, \delta)$,*

$$f_n(w + h) \geq f_n(w) + \langle w_n^*, h \rangle - \epsilon \|h\|. \tag{4.68}$$

This solution by Pernot [77] summarizes conditions under which 'taking a derivative and a limit is interchangeable'. Hence, it is required that the Q-function is a contraction mapping, that the partial derivative is weak*-convergent, and that it is bounded.

Lipschitz continuity of the converged Q-function

As Lipschitz continuity allows to deduce differentiability properties, we will prove that the converged Q-function $\boldsymbol{Q}^*$ is Lipschitz continuous in the parameters $\boldsymbol{\theta}$ similar to the proof by Neu and Szepesvári [74]. In order to do this, we need to make assumptions on the reward function and the model of the dynamics.

Theorem 4.4.12. *Assume that the reward features are uniformly bounded with $\langle s, a \rangle \in S \times A, \|\boldsymbol{f}(s, a)\| < +\infty$, that the parameters are bounded $\|\boldsymbol{\theta}\| < +\infty$, and that the transition model $P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a)$ is Lipschitz continuous as a function of $\boldsymbol{\theta}_{T_E}$ with some Lipschitz constant K. Then, the fixed-point $\boldsymbol{Q}_{\boldsymbol{\theta}}^*$ is Lipschitz continuous as a function of $\boldsymbol{\theta}$ such that for any state-action tuple $\langle s, a \rangle$, it holds that $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^{\Psi}$, $\|\boldsymbol{Q}_{\boldsymbol{\theta}}^* - \boldsymbol{Q}_{\boldsymbol{\theta}'}^*\| \leq L' \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$ with some $L' > 0$.*

Proof. Similar to Neu and Szepesvári [74], consider R to be a Lipschitz constant for $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^{\Psi}, \langle s, a \rangle \in S \times A, \|\boldsymbol{\theta}^{\intercal} \boldsymbol{f}(s, a) - \boldsymbol{\theta}'^{\intercal} \boldsymbol{f}(s, a)\| \leq R \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$ with some $R > 0$. Due to bounded reward features, this can be easily shown, as Lipschitz continuity holds for $R \geq \max_{\langle s,a \rangle \in S \times A} \|\boldsymbol{f}(s, a)\|$.

As parameters are uniformly bounded, it follows that $n \in \mathbb{N}^+$, $\|T^n_{\boldsymbol{\theta}} \boldsymbol{Q}_0\| \leq \boldsymbol{Q}_{max} < +\infty$, and thus both the converged fixed-point as well as intermediate iterates of the Q-iteration operator are bounded, too. In addition, the transition model $0 \leq P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) \leq 1$ is bounded as being a discrete probability distribution.

Analogous to Neu and Szepesvári [74], we assume that Q is L-Lipschitz in $\boldsymbol{\theta}$. Thus, the value function $V_{\boldsymbol{\theta}}(s') = \max_{a' \in A} Q_{\boldsymbol{\theta}}(s', a')$ is L-Lipschitz in $\boldsymbol{\theta}$, too. The latter part of the Q-iteration operator $T_{\boldsymbol{\theta}}(\boldsymbol{Q})$ is an expectation of the value function under the system dynamics $P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a)$. Consequently, both the value function and the system dynamics depend on the parameters $\boldsymbol{\theta}$. For showing Lipschitz continuity of the converged Q-function, it is required to derive Lipschitz bounds for this expectation. The previous assumptions on the dynamics and the Q-function allow to derive an upper bound on the subgradient of this expectation:

$$= \|\frac{\partial}{\partial \boldsymbol{\theta}} \sum_{s' \in S} \left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) V_{\boldsymbol{\theta}}(s') \right] \| \tag{4.69}$$

$$= \|\sum_{s' \in S} \left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) \frac{\partial}{\partial \boldsymbol{\theta}} V_{\boldsymbol{\theta}}(s') \right] + \sum_{s' \in S} \left[\left(\frac{\partial}{\partial \boldsymbol{\theta}} P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) \right) V_{\boldsymbol{\theta}}(s') \right] \| \tag{4.70}$$

$$\leq \|\sum_{s' \in S} \left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) \frac{\partial}{\partial \boldsymbol{\theta}} V_{\boldsymbol{\theta}}(s') \right] \| + \|\sum_{s' \in S} \left[\left(\frac{\partial}{\partial \boldsymbol{\theta}} P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a) \right) V_{\boldsymbol{\theta}}(s') \right] \| \tag{4.71}$$

$$\leq L + |S| K \boldsymbol{Q}_{max}, \tag{4.72}$$

by making use of product rule in Eq. (4.70) and triangle inequality in Eq. (4.71). Then, the first term of Eq. (4.72) can be upper bounded by L as it is an expectation of an L-Lipschitz function. Finally, we make use of Lipschitz-properties of linear combinations and products of bounded Lipschitz-continuous functions on bounded spaces for the second term of Eq. (4.72), which allow to derive global Lipschitz constants according to Eriksson *et al.* [27]. Consequently, this part is a sum of $|S|$ products of the transition model with Lipschitz constant K, which is scaled by the value function that is bounded by $\boldsymbol{Q}_{max}$. From this it follows that the Q-iteration operator $T_{\boldsymbol{\theta}} Q$ is $R + \gamma \left(L + |S| K \boldsymbol{Q}_{max} \right)$-Lipschitz.

As $Q_n = T^n_{\boldsymbol{\theta}} Q_0$ converges to $\boldsymbol{Q}^*$ for $n \to \infty$ [80], $\boldsymbol{Q}^*$ is $R + \gamma \left(|S| K \boldsymbol{Q}_{max} + R \right) + \gamma^2 \left(|S| K \boldsymbol{Q}_{max} + R \right) + \gamma^3 \left(|S| K \boldsymbol{Q}_{max} + R \right) + \ldots = \left(\frac{|S| K \boldsymbol{Q}_{max} + R}{1 - \gamma} - |S| K \boldsymbol{Q}_{max} \right) = \frac{\gamma |S| K \boldsymbol{Q}_{max} + R}{1 - \gamma}$-Lipschitz as being a geometric series if $0 \leq \gamma < 1$, which is satisfied due to the definition of the MDP. Consequently, the fixed-point $\boldsymbol{Q}^*_{\boldsymbol{\theta}}$ is Lipschitz continuous as a function of $\boldsymbol{\theta}$ for arbitrary discount values γ, if the transition model is Lipschitz continuous with some Lipschitz constant K and the reward function is bounded. □

Monotonicity of the Q-gradient iteration operator

In the following, we will prove that the Q-gradient iteration operator is a fixed-point equation and that it's solution is a sub-derivative of $\boldsymbol{Q}^*_{\boldsymbol{\theta}}$. Therefore, it is required to show that the Q-gradient iteration operator is monotone. We define the Q-gradient operator $U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}) \in \mathbb{R}^{|S|\times|A|\times|\Psi|}$ as:

$$U_{\boldsymbol{\theta}}(\boldsymbol{\Phi})(s,a,i) = \frac{\partial}{\partial\theta_i}\boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s,a) \tag{4.73}$$

$$+\gamma\sum_{s'\in S}\left[\left(\frac{\partial}{\partial\theta_i}P_{\boldsymbol{\theta}_{T_E}}\left(s' \mid s,a\right)\right)V_{\boldsymbol{\theta}}\left(s'\right)\right] \tag{4.74}$$

$$+\gamma\sum_{s'\in S}\left[P_{\boldsymbol{\theta}_{T_E}}\left(s' \mid s,a\right)\cdot\Phi(s',\pi^*_{\boldsymbol{\theta}}\left(s'\right),i)\right], \tag{4.75}$$

for all $s \in S, a \in A$, parameter dimensions $i \in \Psi$ with $\Psi = \{1,\ldots,\dim(\boldsymbol{\theta})\}$, the gradient $\Phi(s,a,i) = \frac{\partial}{\partial\theta_i}Q_\theta(s,a)$, and the optimal policy $\pi^*_{\boldsymbol{\theta}}\left(s'\right)$. The Definition 4.4.6 on the partial ordering allows to argue about the monotonicity of the Q-gradient iteration operator.

Lemma 4.4.13. *The Q-gradient iteration operator $U_{\boldsymbol{\theta}}(\boldsymbol{\Phi})(s,a,i)$ is monotone, satisfying $\forall\boldsymbol{\Phi}_m,\boldsymbol{\Phi}_n \in \mathbb{R}^{|S|\times|A|\times|\Psi|} : \boldsymbol{\Phi}_m \preceq \boldsymbol{\Phi}_n \to U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m) \preceq U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n)$.*

Proof. The partial derivative of $U_{\boldsymbol{\theta}}(\boldsymbol{\Phi})(s,a,i)$ with respect to a single value $\Phi(s_k,a_k,k)$ is

$$\frac{\partial}{\partial\Phi(s_k,a_k,k)}U_{\boldsymbol{\theta}}(\boldsymbol{\Phi})(s,a,i) \tag{4.76}$$

$$=\frac{\partial}{\partial\Phi(s_k,a_k,k)}\gamma\sum_{s'\in S}\left[P_{\boldsymbol{\theta}_{T_E}}\left(s' \mid s,a\right)\Phi(s',\pi^*_{\boldsymbol{\theta}}\left(s'\right),i)\right] \tag{4.77}$$

$$=\gamma P_{\boldsymbol{\theta}_{T_E}}\left(s_k \mid s,a\right)\delta\left(a' = \pi^*_{\boldsymbol{\theta}}\left(s'\right), k = i\right). \tag{4.78}$$

As the definition of the MDP ensures $\gamma \in [0,1)$, the probability distribution $P_{\boldsymbol{\theta}_{T_E}}\left(s_i \mid s,a\right) \in [0,1]$, and $\delta(\cdot) \in \{0,1\}$, all terms of the partial derivative $\frac{\partial}{\partial\Phi(s_k,a_k,k)}U_{\boldsymbol{\theta}}(\boldsymbol{\Phi})(s,a,i)$ are positive or zero. Hence, it follows that $\frac{\partial}{\partial\Phi(s_k,a_k,k)}U_{\boldsymbol{\theta}}(\boldsymbol{\Phi})\ (s,a,i) \geq 0$. □

Q-gradient operator is a fixed point equation

Theorem 4.4.14. *The Q-gradient iteration operator $U_{\boldsymbol{\theta}}(\boldsymbol{\Phi})(s,a,i)$ is a contraction mapping with only one fixed point. Therefore, it is Lipschitz continuous $||U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m) -$*

$U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n)||_\infty \leq L||\boldsymbol{\Phi}_m - \boldsymbol{\Phi}_n||_\infty$ for all $\boldsymbol{\Phi}_m, \boldsymbol{\Phi}_n \in \mathbb{R}^{|S|\times|A|\times|\Psi|}$ with a Lipschitz constant $L \in [0, 1)$.

Proof. Consider $\boldsymbol{\Phi}_m, \boldsymbol{\Phi}_n \in \mathbb{R}^{|S|\times|A|\times|\Psi|}$. There exists a distance d for which $\exists d \in \mathbb{R}_0^+$: $||\boldsymbol{\Phi}_m - \boldsymbol{\Phi}_n||_\infty = d$ holds and therefore $-d\mathbf{1} \preceq \boldsymbol{\Phi}_m - \boldsymbol{\Phi}_n \preceq d\mathbf{1}$ with $\mathbf{1} = (1)_{k,l,m}$, where $1 \leq k \leq |S|, 1 \leq l \leq |A|, 1 \leq m \leq |\Psi|$. If d is added to every element of $\boldsymbol{\Phi}_n$, it is guaranteed that $\boldsymbol{\Phi}_m \preceq \boldsymbol{\Phi}_n + d\mathbf{1}$. In addition, $U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m) \preceq U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n + d\mathbf{1})$ satisfies the monotonicity condition of Lemma 4.4.13. Then, it follows $\forall s \in S, a \in A, i \in \Psi$:

$$U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m)(s, a, i) \tag{4.79}$$

$$\leq U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n + d\mathbf{1})(s, a, i) \tag{4.80}$$

$$= \frac{\partial}{\partial \theta_i}\boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s, a) + \gamma \sum_{s' \in S}\left[\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a)\right) V_{\boldsymbol{\theta}}(s')\right] \tag{4.81}$$

$$+ \gamma \sum_{s' \in S}\left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a)\left[\Phi(s', \pi^*_{\boldsymbol{\theta}}(s'), i) + d\right]\right] \tag{4.82}$$

$$= \frac{\partial}{\partial \theta_i}\boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s, a) + \gamma \sum_{s' \in S}\left[\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a)\right) V_{\boldsymbol{\theta}}(s')\right] \tag{4.83}$$

$$+ \gamma \sum_{s' \in S}\left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a)\, \Phi(s', \pi^*_{\boldsymbol{\theta}}(s'), i)\right] + \gamma d \tag{4.84}$$

$$= U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n)(s, a, i) + \gamma d. \tag{4.85}$$

This yields $U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m) \preceq U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n) + \gamma d\mathbf{1}$. Since d is defined symmetrically, it equally holds that $\boldsymbol{\Phi}_n \preceq \boldsymbol{\Phi}_m + d$ and $U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n) \preceq U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m) + \gamma d\mathbf{1}$. We can show the Lipschitz continuity of the Q-gradient iteration by combining these inequations:

$$-\gamma d\mathbf{1} \preceq \quad U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m) - U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n) \quad \preceq \gamma d\mathbf{1}$$

$$||U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m) - U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n)||_\infty \leq \gamma d$$

$$||U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_m) - U_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_n)||_\infty \leq \gamma||\boldsymbol{\Phi}_m - \boldsymbol{\Phi}_n||_\infty.$$

This proves that the Q-gradient iteration operator $U_{\boldsymbol{\theta}}(\boldsymbol{\Phi})(s, a, i)$ is Lipschitz continuous with a Lipschitz constant $L = \gamma$ with $\gamma \in [0, 1)$, resulting in a contraction mapping. As this holds for the whole input space $\mathbb{R}^{|S|\times|A|\times|\Psi|}$, two points will always contract. As a consequence, there cannot exist two fixed points. □

Fixed-point of the Q-gradient iteration is a derivative of $\boldsymbol{Q}^*_{\boldsymbol{\theta}}$

Finally, we need to show that the resulting fixed-point of the Q-gradient iteration is a valid sub-derivative of the converged Q-function.

Theorem 4.4.15. *Except on a set of measure zero, the gradient $\nabla_{\boldsymbol{\theta}} Q^*_{\boldsymbol{\theta}}(s,a)$ of the Q-function, is given by the solution of the fixed-point equation $\Phi_n(s,a,i) = \frac{\partial}{\partial\theta_i}\boldsymbol{\theta}_R^\intercal \boldsymbol{f}(s,a)$ $+\gamma \sum_{s'\in S}\left[\left(\frac{\partial}{\partial\theta_i} P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a)\right) V_{\boldsymbol{\theta}}(s')\right]$ $+\gamma \sum_{s'\in S}\left[P_{\boldsymbol{\theta}_{T_E}}(s' \mid s,a)\, \Phi_{n-1}(s, \pi^*_{\boldsymbol{\theta}}(s'), i)\right]$, where the policy $\pi^*_{\boldsymbol{\theta}}$ is any greedy policy with respect to the converged Q-function.*

Proof. We derive our proof analogous to Neu and Szepesvári [74]. Bellman [13] has shown that the value iteration converges to a fixed point. By Theorem 4.4.14, $\Phi_{n+1}(s,a,i) = U_{\boldsymbol{\theta}}\Phi_{n+1}(s,a,i)$ converges to a fixed-point $\Phi(s,a,i)^*$ for large n. Proposition 4.4.10 says that the subdifferential $\Phi(s,a,i)^* \in \partial^-_{\theta_i}\boldsymbol{Q}^*_{\boldsymbol{\theta}}$. exists. Furthermore, by Proposition 4.4.11, the limit is a subdifferential of $\boldsymbol{Q}^*_{\boldsymbol{\theta}}$, as the Q-function is a contraction mapping, the subdifferential is convergent, and it is bounded. Since Theorem 4.4.12 proves that the converged Q-function $\boldsymbol{Q}^*_{\boldsymbol{\theta}}$ is Lipschitz continuous as a function of the parameters $\boldsymbol{\theta}$, Rademacher's theorem says that it is differentiable almost everywhere. If a function is differentiable, it's subderivative coincides with its derivative, which finishes the proof [53]. □

Hence, we have shown that the converged Q-function is subdifferentiable, that it is Lipschitz continuous, that the Q-gradient operator is a fixed point equation, and that the fixed-point is a derivative (or sub-derivative) of $\boldsymbol{Q}^*_{\boldsymbol{\theta}}$. Consequently, by optimizing parameters with the converged Q-gradient, both the environments dynamics and the reward function of the expert can be learned.

4.4.3 Discussion of policy models

Analyzing the Q-gradient of Eq. (4.66) and the soft Q-gradient of Eq. (4.21) indicates that the first two terms are constants, which specify the gradient of the reward function and the partial derivative of the transition model, which is weighted by the corresponding value. The last term adds the expected gradient and propagates it through the space of states S, actions A, and parameter dimensions $\boldsymbol{\theta}$. Fig. 4.5 visualizes this propagation and, thus, gives insights how the gradients are constructed. The main difference between the Q-gradients of the two approaches is that PM SERD propagates the gradients only along the optimum action, while MDCE SERD updates them along the stochastic policy. Since following a stochastic policy will result in a higher variance in the expected state

Table 4.2: Complexity of the Q-gradient computation

	Soft Q-gradient	Q-gradient
LU decomposition	$\mathcal{O}\left(N_{\boldsymbol{\theta}} \cdot (\lvert S\rvert \cdot \lvert A\rvert)^3\right)$	$\mathcal{O}\left(N_{\boldsymbol{\theta}} \cdot (\lvert S\rvert \cdot \lvert A\rvert)^3\right)$
1-step iteration	$\mathcal{O}\left(N_{\boldsymbol{\theta}} \cdot (\lvert S\rvert \cdot \lvert A\rvert)^2\right)$	$\mathcal{O}\left(N_{\boldsymbol{\theta}} \cdot \lvert A\rvert \cdot \lvert S\rvert^2\right)$

probability, the MDCE SERD solution considers a broader range of states and actions. Therefore, it can be expected that MDCE SERD is more stable and generalizes better than PM SERD.

Table 4.2 illustrates the computational complexity of solving the linear equation systems of the Q-gradients with $|S|$ states, $|A|$ actions, and $N_{\boldsymbol{\theta}}$ parameters. The computational cost of a direct computation via LU decomposition requires a number of computations that is proportional to $(|S| \cdot |A|)^3$. Especially if the state- and action-space is large, its computation can be very expensive. In such cases, an iterative approach can drastically reduce the computational cost. The complexity of PM SERD is lower than that of MDCE SERD, since it omits to compute the expected gradient. Furthermore, Alg. 1 requires to compute the gradient in an inner loop repeatedly with slightly changed parameters. Typically, this little change has only small effects on the gradient. Therefore, initializing the gradient with the solution of the last iteration offers the possibility of a reduction of the number of iterations.

4.5 Priors

The specification of the SERD problem introduces additional parameters compared to IRL, as it extends reward learning by a simultaneous estimation of the transition model.

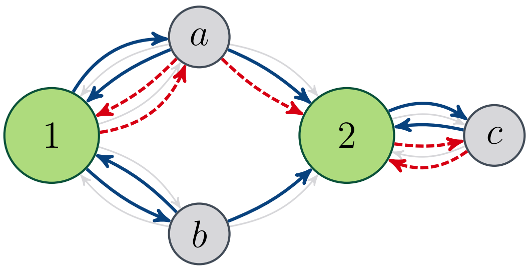


Figure 4.5: Visualization of the composition of the Q-gradients in the minimum example. PM SERD (red, dashed) propagates the gradients only along the optimum action. In contrast, MDCE SERD (blue) propagates them along the Boltzmann policy. Therefore, each partial derivative is influenced by all reachable states and actions.

However, the higher number of parameters increases the risk of overfitting the model to the data. Furthermore, (partially) unknown dynamics may introduce ambiguities. A possible solution to prevent overfitting and to resolve ambiguities is the use of priors $P(\boldsymbol{\theta})$. In the following, we discuss different priors. Some of them are applicable to all parameters and others only to transition model parameters.

4.5.1 Lp-norm

The Lp-norm jointly regularizes a set of parameters with different characteristics. For example, the L1-regularization enforces sparsity in the parameters. If it is applied to the feature weights $\boldsymbol{\theta}_R$, it will learn reward functions that strongly depend on a subset of features, while other features receive zero weights. The influence of L1-regularization on the transition model estimate depends on the model of the dynamics. However, if the parameters directly relate to the probability of a certain state, e.g. $\log P\left(s'|s,a\right) \propto \theta_k$, deterministic dynamics are learned. In contrast, L2-regularization is jointly shrinking the parameters and is not directly enforcing sparsity. However, since it decays the parameters, it prevents the reward function to become very large and Boltzmann policies not to be too sharp. The same applies to transition models, of which the parameters directly relate to the probability. The regularization term can be specified as:

$$\log P(\boldsymbol{\theta}) = -\lambda \|\boldsymbol{\theta}\|_p - Z \tag{4.86}$$

with the normalization constant $Z = \log\left[\int_{\boldsymbol{\rho}} \exp\left(-\lambda \|\boldsymbol{\rho}\|_p\right) d\boldsymbol{\rho}\right]$. Since it is constant for all $\boldsymbol{\theta}$, it can be ignored. The partial derivative of the Lp-norm is

$$\frac{\partial}{\partial \theta_i} \log P(\boldsymbol{\theta}) = -\frac{\theta_i |\theta_i|^{p-2}}{\|\boldsymbol{\rho}\|_p^{p-1}}. \tag{4.87}$$

4.5.2 Entropy regularization

Entropy regularization is applicable to parameters of probability distributions, such as the environment's dynamics. It enforces them to have a high entropy, which relates to flat probability distributions and thus avoids learning sharp, deterministic dynamics. The log likelihood of this prior is

$$\log P(\boldsymbol{\theta}) = \lambda \sum_{s \in S} \sum_{a \in A} \sum_{s' \in S} \left[-P_{\boldsymbol{\theta}_T}\left(s' \mid s, a\right) \cdot \log P_{\boldsymbol{\theta}_T}\left(s_{t+1}^{\tau} \mid s_t^{\tau}, a_t^{\tau}\right)\right] - Z, \tag{4.88}$$

with the following gradient:

$$\frac{\partial}{\partial \theta_i} \log P(\boldsymbol{\theta}) = -\lambda \sum_{s \in S} \sum_{a \in A} \sum_{s' \in S} \tag{4.89}$$

$$\left(\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_T} \left(s' \mid s, a \right) \right) \left(\log P_{\boldsymbol{\theta}_T} \left(s_{t+1}^{\tau} \mid s_t^{\tau}, a_t^{\tau} \right) + 1 \right). \tag{4.90}$$

4.5.3 KLD regularization

For some problems, it might be possible to specify a prior distribution Q that the learned probability distribution $P_{\boldsymbol{\theta}_T}$ should be close to. In these cases, it is possible to apply KLD regularization, which results in a large loss, if the relative entropy from Q to P is large. Hence, this regularization penalizes differences between the two distributions. The log likelihood of this prior is given by

$$\log P(\boldsymbol{\theta}) = -\lambda \sum_{s \in S} \sum_{a \in A} \sum_{s' \in S} \left[Q\left(s' \mid s, a \right) \cdot \log \frac{Q\left(s_{t+1}^{\tau} \mid s_t^{\tau}, a_t^{\tau} \right)}{P_{\boldsymbol{\theta}_T}\left(s_{t+1}^{\tau} \mid s_t^{\tau}, a_t^{\tau} \right)} \right] - Z, \tag{4.91}$$

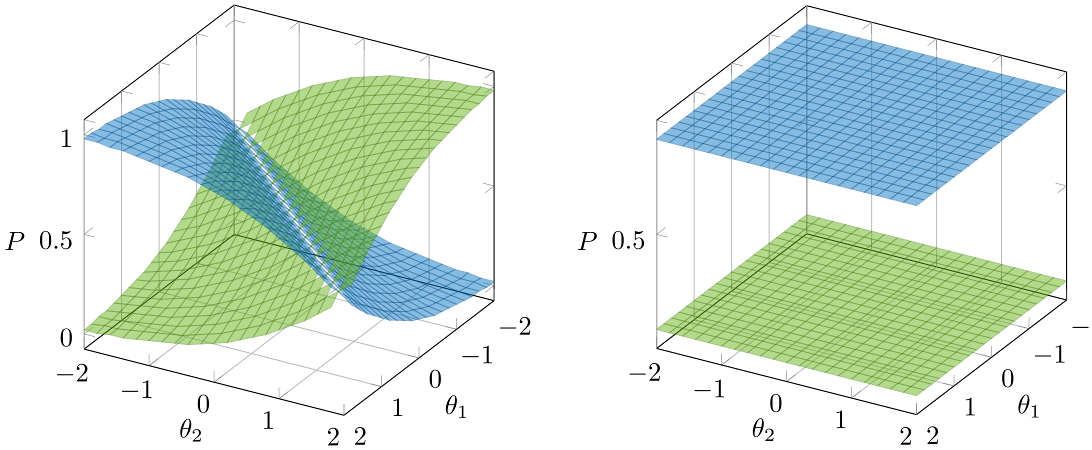
with the following gradient:

$$\frac{\partial}{\partial \theta_i} \log P(\boldsymbol{\theta}) = -\lambda \sum_{s \in S} \sum_{a \in A} \sum_{s' \in S} Q\left(s' \mid s, a \right) \frac{\frac{\partial}{\partial \theta_i} P_{\boldsymbol{\theta}_T}\left(s' \mid s, a \right)}{P_{\boldsymbol{\theta}_T}\left(s' \mid s, a \right)}. \tag{4.92}$$

4.5.4 Discussion

To clarify the characteristics and properties of different priors on resulting probability distributions, we present an example in which the parameters of a Boltzmann distribution $P_{\boldsymbol{\theta}}(x_i) = \frac{\exp(\theta_i)}{\sum_{j \in \{1,2\}} \exp(\theta_j)}$ are optimized. Fig. 4.6 (a) illustrates this distribution over parameters $\boldsymbol{\theta}$, which indicates that $P_{\boldsymbol{\theta}}(x_1)$ and $P_{\boldsymbol{\theta}}(x_2)$ become more different the larger the difference in the respective parameters θ_1 and θ_2 is. Fig. 4.6 (b) illustrates a prior distribution with $Q(x_1) = 0.9$ and $Q(x_2) = 0.1$, which is used for visualizing the KLD regularization. In typical MAP optimization problems, the objective consists of a likelihood and a prior term (see the SERD problem formulation Eq. (4.6)). In this example, the log likelihood of the prior is maximized under the parameter constraint $\|\boldsymbol{\theta}\|_2 = 1$, which is a substitute for the likelihood term. That way, it is possible to exemplify the influence of using various priors during optimization. Tab. 4.3 shows the surface of the log likelihood of the introduced priors together with the resulting optimal Boltzmann distributions $P_{\boldsymbol{\theta}}(x_i)$.

The surface of the *L1-norm prior* is not smooth. From all parameters $\boldsymbol{\theta}$ the ones that lie on the edge of the surface maximize the log prior. Hence, there exist four optimal parameter sets, while two of them result in the same distribution. In contrast, under the *L2-norm prior* all parameters that satisfy the constraint have equal log priors. Hence, there exist infinitely many equally good solutions for the probability distributions, while two samples of them are illustrated. Comparing the surface of the log *Entropy prior* over the parameters in Tab. 4.3 to the plot of the distribution in Fig. 4.6 (a), reveals that the log prior is high, when $P(x_1)$ and $P(x_2)$ are close. This is reasonable, since the Entropy prior penalizes sharp or deterministic probability distributions. Consequently, optimal parameters lie on the line with equal parameters $\theta^* = \theta_1 = \theta_2$ and thus result in a distribution $P(x)$ that is uniform. Finally, the ***KLD prior*** uses the prior distribution shown in Fig. 4.6. The negative KLD from Q to $P_{\boldsymbol{\theta}}$ in Tab. 4.3 is skewed to the left side, which is more similar to the prior (measured by KLD). Hence, optimal parameters result in a probability distribution $P^*(x)$ that is as close as possible to $Q(x)$ under the parameter constraint. In the example, this results in $P^*(x_1) \gg P^*(x_2)$, while it was not possible to exactly fit it due to the constraint $P^*(x_1) \neq Q(x_1)$.



(a) Parametric distribution $P(x_i) = \exp(\theta_i)/Z$ over varying parameters $\boldsymbol{\theta}$. $P_\theta(x_1)$ is indicated in blue and $P_{\boldsymbol{\theta}}(x_2)$ in green.

(b) Prior distribution $P(x_1) = 0.9$ (blue) and $P(x_2) = 0.1$ (green), which is constant over the parameters.

Figure 4.6: This figure illustrates an exemplary parametric distribution $P_\theta(x)$ that is used for visualizing and explaining various regularization methods for the parameters $\boldsymbol{\theta}$. (a) plots a Boltzmann distribution $P_{\boldsymbol{\theta}}(x_i) = \exp(\theta_i)/Z$ over the parameters $\boldsymbol{\theta}$, while (b) shows the prior distribution ($Q(x_1) = 0.9$ and $Q(x_2) = 0.1$).

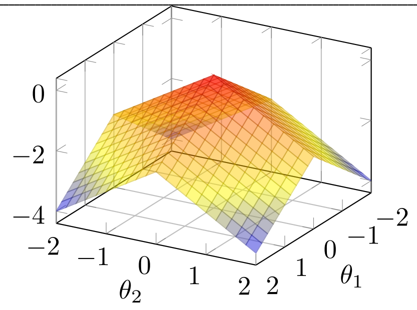
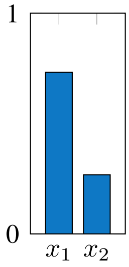
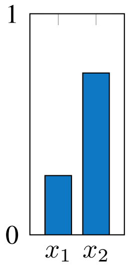
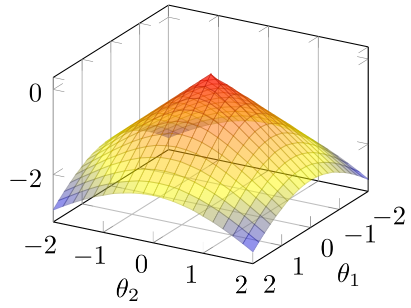
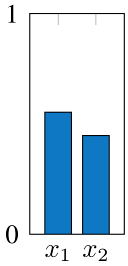
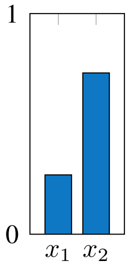
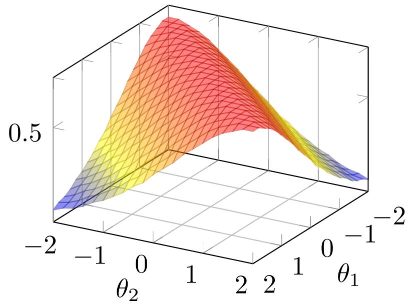
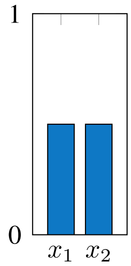
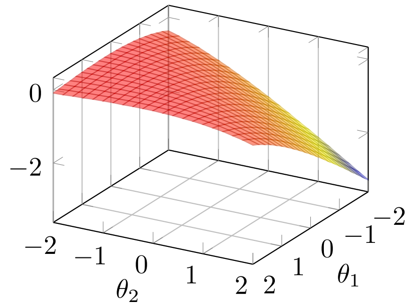
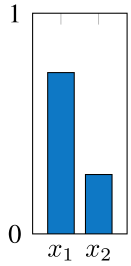
Reg.	**Log Prior Surface**	**Optimal distribution(s) $P(x)$**
L1		 
L2		 infinitely many solutions 
Entropy		
KLD$(Q\|P)$		

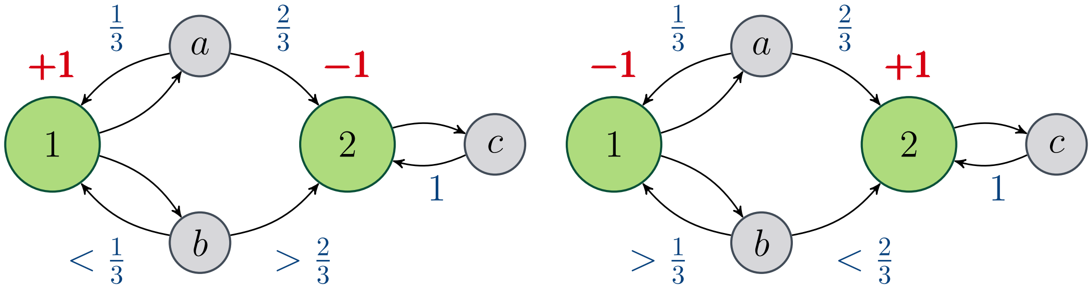
Table 4.3: The center column of this table illustrates the log prior surfaces of the parameters of a Boltzmann distribution with two outcomes for various regularization methods. The Boltzmann distribution corresponding to the different parameters is shown in Fig. 4.6. Furthermore, the right column illustrates the resulting optimal distributions under the constraint that the L2-norm of the parameters is $\|\boldsymbol{\theta}\|_2 = 1$.

4.6 Evaluation

In the previous section, we have introduced a toy example with only two states and three actions (see Fig. 4.2) to give an overview on IRL and common pitfalls. For example, when the environment's dynamics are partially or fully unknown, it might not be possible to deduce the expert's reward function due to additional ambiguities, which stem from the ignorance of the model of the environment. While many IRL approaches assume that the transition model is known or that it can be learned from the observed transitions prior to reward learning, this might result in estimates that are biased towards suboptimal solutions. However, rational and (close to optimal) experts typically consider both their individual reward as well as the transition model of the environment. Therefore, it is possible to estimate both from demonstrations of expert behavior.

To make use of this additional information and to overcome the IRL pitfalls, the previous section introduced SERD, which is an approach for a simultaneous estimation of rewards and dynamics solely from expert demonstrations. The theoretical derivations indicate that the combined approach can be beneficial, since it allows making use of both the observed transitions and the observed actions of the expert for learning a model of the dynamics. In addition, a more accurate model of the dynamics could also improve the reward estimates. However, it is not clear whether the approach really improves the estimates, since learning a larger number of parameters might also result in overfitting, when learning from a small, fixed dataset of observations. Furthermore, the SERD approach requires to make assumptions about the policy model of the expert. In Sec. 4.4, we have introduced MDCE and PM SERD motivated by popular policy models used in IRL. Nevertheless, wrong assumptions about the policy model might tamper with the estimate of the environment's dynamics. Since this transition model also influences the estimate of the expert's reward function, it could also be falsified.

Therefore, the following section evaluates SERD against classical IRL methods on several varying examples. First, we revisit the toy example from Fig. 4.3 to check the claims about pitfalls from Chap. 4 and whether SERD can solve them. Afterwards, we use a grid world navigation task based on satellite images to examine the capabilities of SERD in a more complex scenario that allows determining differences in the estimates exactly by specifying ground truth models.



(a) MDP A

(b) MDP B

Figure 4.7: Toy example based on the previously introduced MDP. It is assumed that a set of expert demonstrations is available, in which the expert always starts in state 1 and chooses action a. Therefore, action b is never observed. The Figures (a) and (b) illustrate two possible MDPs that could have caused these observations: In (a) the expert chooses action a to stay in state 1 since it has a higher reward than state 2 and action a has a higher probability to stay in state 1. In contrast, (b) depicts an MDP where the optimal expert also chooses action a, but in this case with the ambition to transition to state 2 due to a higher reward. Hence, the probability to end up in state 2 is higher when applying action a.

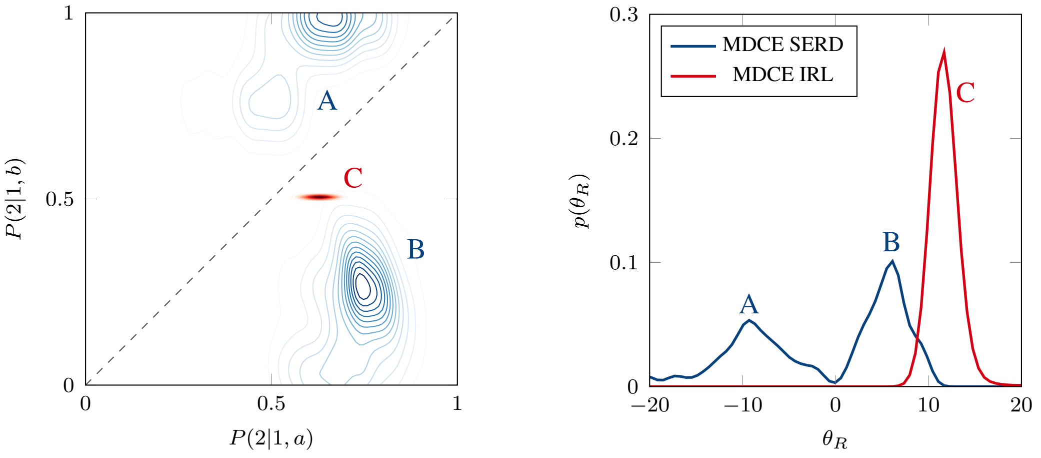
4.6.1 Minimum Example

In Sec. 4.1, we have introduced several limitations of IRL by means of a toy example environment that is depicted in Fig. 4.2. An agent may choose from action a or b in state 1, while in state 2 only action c is available. As soon as the agent reaches state 2, he is not able to transit back to state 1, since only action c is available with the single successor state 2. As illustrated in Fig. 4.7, the ground truth dynamics of the environment are set to $P(2 \mid 1, a) = \frac{2}{3}$, $P(1 \mid 1, a) = \frac{1}{3}$, and $P(2 \mid 2, c) = 1$. In this experiment, the only available information is that the expert always chooses action a. Consequently, the agent never observes samples from the transition model under action b. According to the analysis in Sec. 4.1, the behavior could have been caused by the two different sets of MDPs that are visualized in Fig. 4.7, while in MDP A the agent chooses action a with the goal to stay in state 1 and in MDP B the agent chooses action a to transition to state 2.

For running IRL and SERD, a parametric model of the environments dynamics is required that should be learned from demonstrations. Therefore, four transition model parameters are introduced for $P(1 \mid 1, a)$, $P(2 \mid 1, a)$, $P(1 \mid 1, b)$, and $P(2 \mid 1, b)$, which are parameterized by energies of Boltzmann distributions.

According to the problem statement, the dynamics are unknown in advance. However, MDCE IRL requires a transition model for learning and for MDCE SERD a good initial guess is probably beneficial. Therefore, we used an M-estimator with a uniform prior to estimate a transition model directly from expert demonstrations. Since action b is never observed as the expert always chooses action a, the initial guess of the dynamics for action b will be the uniform prior $P_{\boldsymbol{\theta}}(1 \mid 1, b) = 0.5$ and $P_{\boldsymbol{\theta}}(2 \mid 1, b) = 0.5$.

To reduce the complexity of the learning task, we simplify the reward function $R(s) = \theta_R f(s)$ to a single weighted state-dependent feature, which is $f(s = 1) = -1$ and $f(s = 2) = 1$. Therefore, by flipping the sign of the feature weight, the reward function changes between Fig. 4.7 (a) and (b). By increasing $|\theta_R|$, the difference between the rewards of the two states increases, which also increases differences in Q-values. Consequently, Boltzmann distribution based policy models such as the MDCE (Eq. (3.30)) or PM (Eq. (3.11)) will be less random and even become deterministic for $\theta_R \to \pm\infty$.



(a) PDF of learned transition models

(b) PDF of feature weights

Figure 4.8: Results of the toy example. (a) Estimated probability density function of the transition models that have been learned by MDCE SERD (blue, A and B) or used by MDCE IRL (red, C). (b) Estimated probability density function of the learned feature weights θ_R.

Finally, we sample $5,000$ expert demonstration sets of varying sizes and a horizon of $H = 3$ from the deterministic, optimal policy, which always chooses action a in state 1. We use MDCE SERD and MDCE IRL for learning with randomly initialized feature weights $\theta_R \in [-10, 10]$. Matching Boltzmann distributions to deterministic behavior is not possible unless the energies become $\pm\infty$. To ensure nondegenerate distributions, an L2-regularization with a small weight of 10^{-6} is applied to the parameters of the transition model $\boldsymbol{\theta_T}$ and the reward function θ_R.

Fig. 4.8 (a) illustrates the density estimate of the resulting transition models with $P(2 \mid 1, a)$ and $P(2 \mid 1, b)$ on the x- and y-axis. The dashed, diagonal line separates the two options that we have identified in Sec. 4.1 and which mean that either action a or b has a higher probability to cause a transition to the second state (See Fig. 4.7). This difference of the dynamics changes the interpretation of the expert's behavior, by making either the reward of state 1 or of state 2 higher.

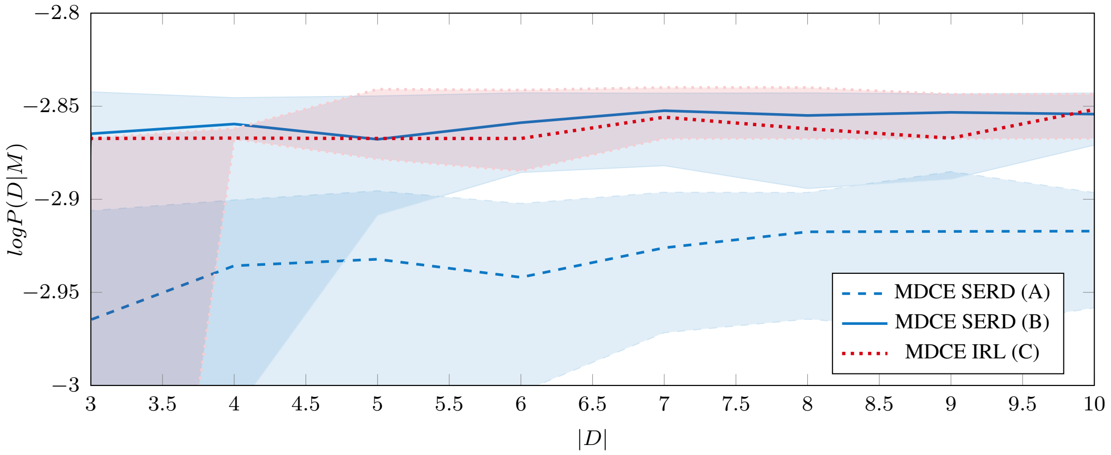


Figure 4.9: Median and quartiles of the log likelihood of ground truth expert demonstrations under the learned models of the reward function and the transition model over varying demonstration set sizes.

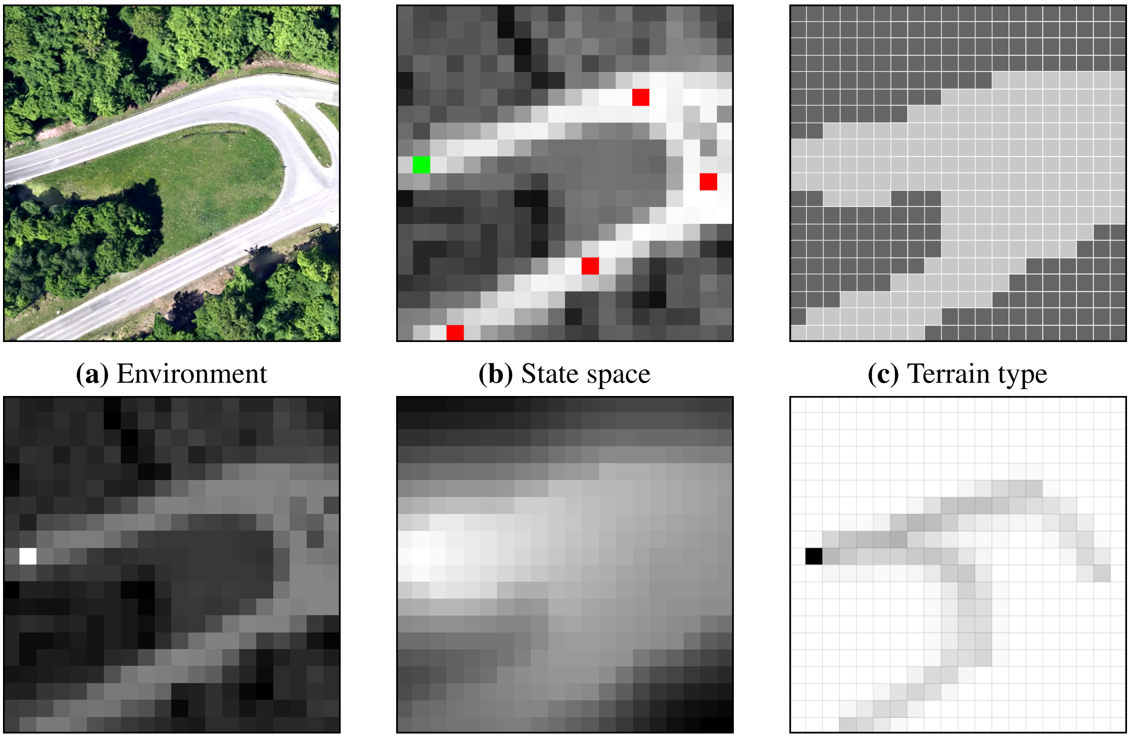
It can be seen that there exists one red mode (C), which belongs to MDCE IRL. Since we have specified a uniform prior for the initial estimation of the transition model, this mode lies at $P(2 \mid 1, b) = P(1 \mid 1, b) = 0.5$, while the probability for a transition to state 2 roughly matches the ground truth model $P(2 \mid 1, a) = \frac{2}{3}$. However, MDCE IRL was only able to find one of the two possible solutions, since the initialization of the transition model introduces a bias. In contrast, different runs of MDCE SERD end up in both modes (A and B), that lie in different regions and thus correspond to the two different explanations of the observed behavior.

Fig. 4.8 (b) also indicates that MDCE SERD finds both explanations for the expert's behavior. Furthermore, it even allows arguing that (A) and (B) directly correspond to Fig. 4.7 (a) and (b). At the same time, MDCE IRL gets stuck in one mode. Indeed, by using another prior for the M-estimator, MDCE IRL could have found the other mode, too. However, without training with different transition model initializations, it is not possible to find different modes. Without further knowledge it is not possible to say, which of the two modes of MDCE SERD models correspond to the real environment. While MDCE SERD allows an expert to choose the right one of them, MDCE IRL outputs only one estimate. Thus, an expert would typically interpret this mode as the correct one. We believe that this property of MDCE SERD is an advantage, since it allows experts to find and to resolve ambiguities. Finally, Fig. 4.9 illustrates the median log likelihood of ground truth demonstrations under the stochastic policy of the learned model. When splitting the MDCE SERD estimates according to the mode that they belong to, it can be seen that the estimates of MDCE IRL and the corresponding mode of MDCE SERD perform similarly. The second mode of MDCE SERD has a lower log likelihood. Hence, this measure might indicate, which of the modes the more likely one is.

Altogether, this evaluation has demonstrated some of the claims about the limitations of IRL. For example, we have shown that naive estimates of the dynamics can be inaccurate due to the goal-bias of demonstrations. Furthermore, we have illustrated that unknown transition models can introduce ambiguities, which can often only be found if the problem is solved simultaneously. Otherwise, IRL ends up with one of the solutions, which could be the wrong one.

4.6.2 Gridworld Navigation

To evaluate the performance of the proposed approaches on a more complex problem, we have created an artificial satellite navigation task, which is illustrated in Fig. 4.10 and 4.11. The evaluation consists of two parts: First, we train a model from samples in the training environment (see Fig. 4.10) and then evaluate it in the transfer environment (see


(a) Environment **(b)** State space **(c)** Terrain type

(d) Reward **(e)** Value **(f)** State frequency

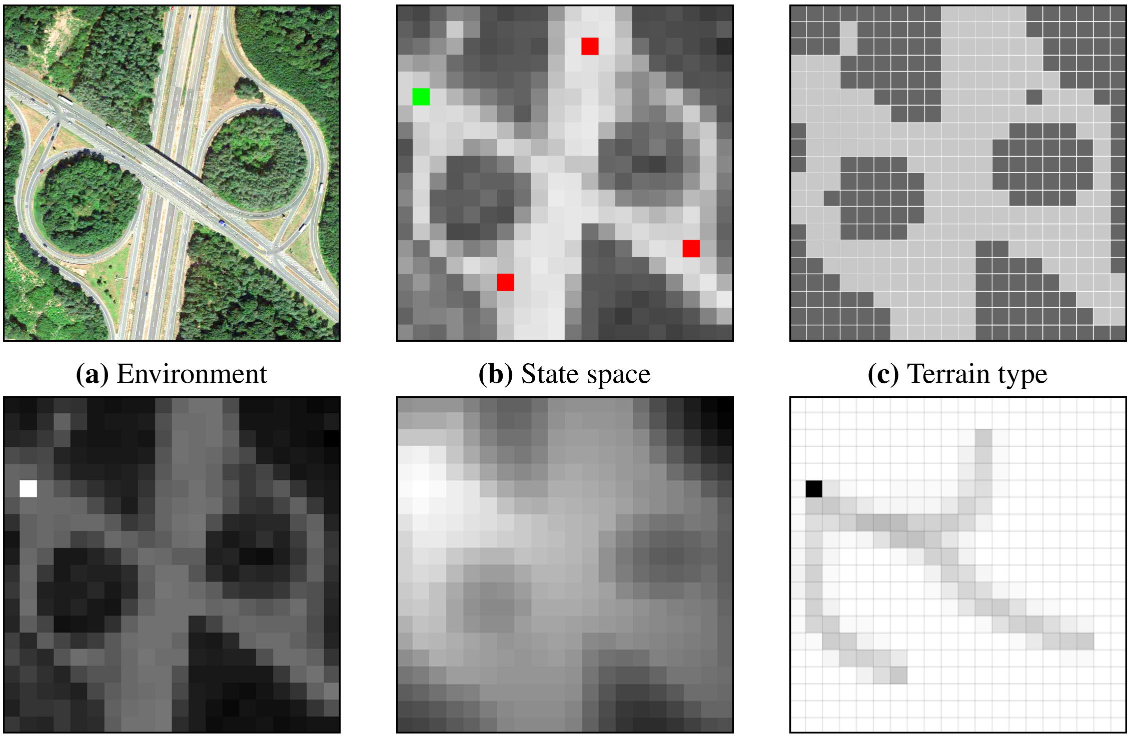
Figure 4.10: Illustration of properties of the training environment. (a) Environment, Map data: Google. (b) Discretized state space. The goal state is indicated in green and start states in red. (c) Forest states are indicated in a dark-gray color and open terrain in light gray. Furthermore, plot (d) shows the reward, (e) the resulting value function, and (f) the expected state frequency.

Fig. 4.11) for checking its generalization performance.

This experiment aims for recovering the environment's dynamics as well as the ground truth reward function that an expert uses for reaching desired states via navigating on satellite images. Fig. 4.10 (b) and Fig. 4.11 (b) indicate the goal states in green and initial states in red. The satellite image is discretized into 20×20 states and the action space consists of five different actions, which are moving in one of four directions (north, east, south, or west) or staying in the state. However, the dynamics restrict potential successor states such that transitions can only occur to the four neighboring states or the current one. If the agent is in an open terrain state (Fig. 4.10 (c) and Fig. 4.11 (c): depicted in light gray) a successful move into the desired direction occurs with probability 0.8. Otherwise, he moves to the left or to the right of the motion direction with probability 0.1. In contrast, the dynamics in the forest are more noisy with a probability of 0.3 of successful transitions or otherwise 0.175 for moving into one of the remaining successor states. Staying in the current state is always successful. Altogether, the agent moves faster on open terrain than through the forest. Accordingly, he needs to trade off between going straight through the forest or taking longer paths on open terrain, which are more likely to be successful.

As defined in Sec. 4.4, the reward function is a linear combination of state-dependent features. The first feature is the average gray scale value of the image in a certain state and is normalized to $[0, 1]$. The second feature indicates the goal state, by being 1 in the goal area or 0 otherwise. To enable a more substantial evaluation, we do not ask humans to provide expert demonstrations, but rather sampled them from ground truth models, which allow for evaluating the resulting estimates. Therefore, we specify the discount as 0.99 and set the ground truth feature weights to $\boldsymbol{\theta}_R = (6, 6)^\intercal$. Additionally, the policy parameter of PM SERD, which scales the stochasticity, is set to $\theta_P = 2$. Fig. 4.10 (d) and Fig. 4.11 (d) illustrate the resulting reward function. They visualize that the roads and the goal state have a higher reward than the lawn or the forest. Consequently, an agent needs to weigh between low rewards and slow progress in the forest, medium rewards and fast progress on lawn, or higher rewards and fast progress on roads.

Based on this definition, we have computed stochastic policies of MDCE and PM according to Eq. (4.13) and Eq. (4.62) from the corresponding value and soft-value functions. Fig. 4.10 (e), (f) and Fig. 4.11 (e), (f) illustrate the soft value function of the MDCE policy together with the state probability distribution that arises when an agent applies the MDCE policy in the environment. The optimal value function, the resulting PM policy, and the corresponding state probability distribution differ only slightly from



(d) Reward **(e)** Value **(f)** State frequency

Figure 4.11: Illustration of properties of the transfer environment. (a) Environment, Map data: Google. (b) Discretized state space. The goal state is indicated in green and start states in red. (c) Forest states are indicated in a dark-gray color and open terrain in light gray. Furthermore, plot (d) shows the reward, (e) the resulting value function, and (f) the expected state frequency.

the MDCE one. Therefore, we do not present the corresponding plots.

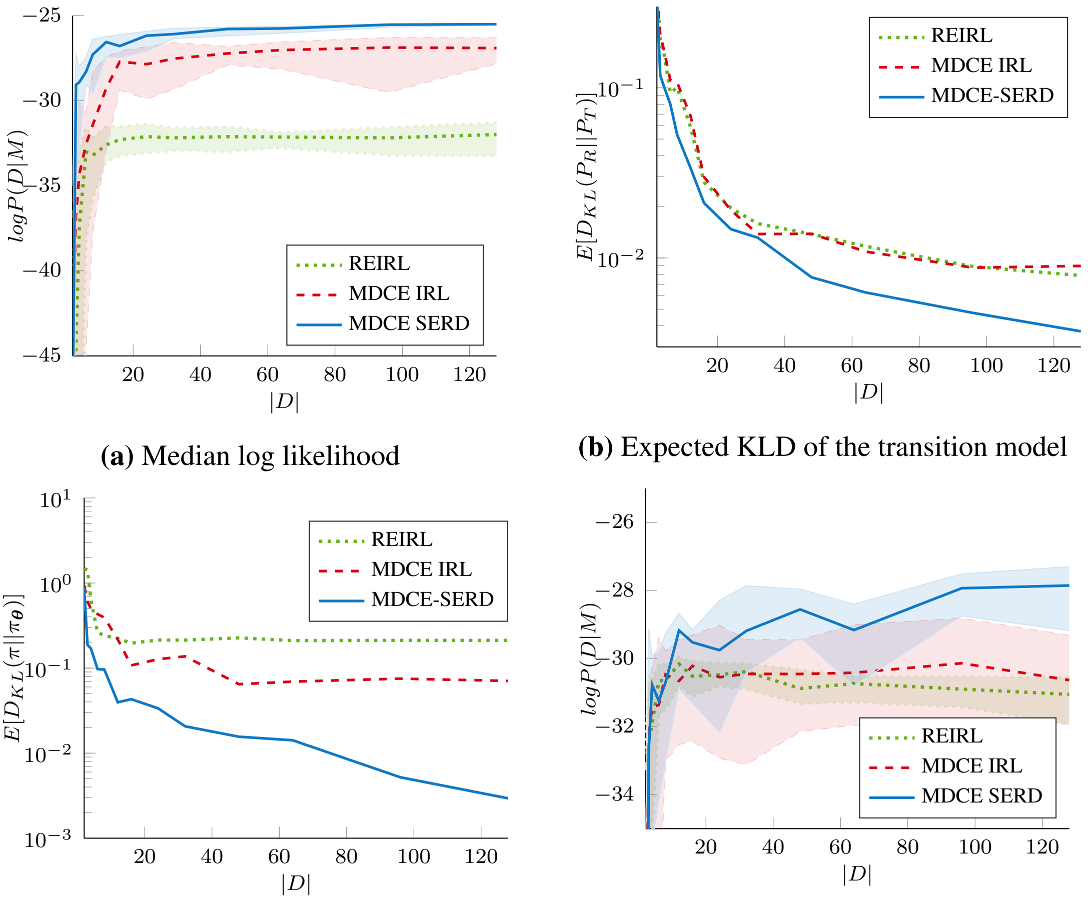
These stochastic policies as well as the previously specified transition model are the ground truth models for sampling expert demonstrations. Sec. 4.6.2 uses samples from the MDCE policy, while Sec. 4.6.2 evaluates learning from samples of the PM policy. The definition of the unified SERD problem distinguishes between the true environment's dynamics $P_{\boldsymbol{\theta}_T}$ and the expert's belief about them $P_{\boldsymbol{\theta}_{T_E}}$. In this experiment, we assume that the expert has full knowledge about the true dynamics of the environment. Therefore, the parameters $\boldsymbol{\theta}_{T_E}$ and $\boldsymbol{\theta}_T$ are equal and the optimization of the transition model parameters uses both terms of Eq. (4.9). In general, it is problem dependent whether they can be assumed to be identical. If the expert has a lot of knowledge about the environment and is indeed able to provide optimal demonstrations, it is fine to make this assumption. However, violating this assumption introduces a bias. The usage of differing models would require introducing an additional regularization term, which relates the two models to each other and keeps them similar. This is a future field of research, which we do not consider in this thesis.

Due to the problem statement, we assume that the dynamics of the environment are unknown and only a fix set of expert demonstrations is available. Therefore, we estimate the dynamics by an M-estimator with a uniform prior from expert demonstrations. For each action and terrain type an independent transition model is trained, except for the staying action which is identical in both terrains. We have chosen independent models to increase the complexity of the learning task (larger number of learnable parameters), even though that the transition models are rotational invariant. Altogether, we have trained 9 transition models with each having 5 possible successor states, resulting in 45 transition model parameters, which are modeled by energies of Boltzmann distributions.

Finally, the evaluation compares MDCE IRL, REIRL, and either MDCE SERD or PM SERD in terms of reward function learning from the sampled ground truth expert demonstrations. For learning, we initialize the feature weights by randomly sampling them from a uniform distribution $\forall i : \theta_i \in [-10, 10]$. All approaches are trained on various demonstrations set sizes and use the M-estimated dynamics for learning. However, the proposed SERD approaches further optimize the estimate of the dynamics, while the other approaches assume the given transition model to be the real one. REIRL requires an additional set of samples from an arbitrary known policy. Since the problem assumption is that only expert demonstrations are available, we sample arbitrary demonstrations from the M-estimated dynamics and an M-estimated policy trained via imitation learning from expert demonstrations. In the following, we discuss the results of the two SERD approaches.

MDCE SERD

Fig. 4.12 summarizes the results of MDCE SERD and alternative approaches based on IRL from expert demonstrations, which stem from applying a ground truth MDCE policy in the environment. Since no prior knowledge is available, none of the possible solutions to the problem is more likely than the others. Therefore, we compute MDCE SERD estimates with a uniform prior for both the reward function parameters as well as the transition model parameters. The first figure shows a comparison of the median log likelihood of ground truth demonstrations under the learned models. It indicates that MDCE SERD outperforms the other approaches, is very sample efficient, and has a low variance already for small numbers of demonstrations. MDCE IRL has a lower performance than MDCE SERD. One reason might be that it conditions on a potentially inaccurate model of the dynamics and cannot correct it. REIRL has the lowest performance in this task among the



(a) Median log likelihood

(b) Expected KLD of the transition model

(c) Expected KLD of the policy

(d) Median log likelihood (transfer task)

Figure 4.12: Evaluation of MDCE SERD. (a) Median log likelihood with quartiles of demonstrations drawn from the true model under the estimated model. (b) Average Kullback-Leibler divergence between the true dynamics of the environment and the estimated one. (c) Average Kullback-Leibler divergence between the true ground truth policy and the learned one. (d) Median log likelihood with quartiles of demonstrations drawn from the true model under the estimated model in the transfer task.

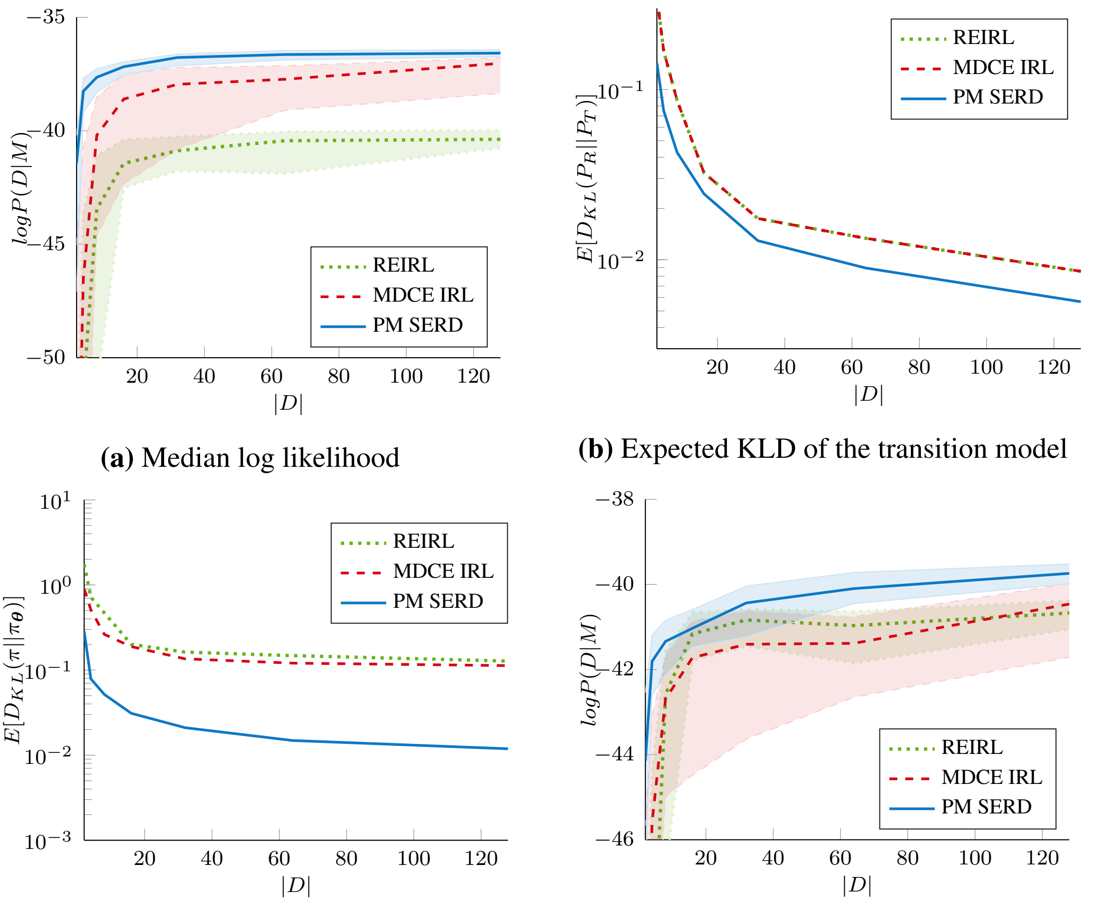
evaluated methods. This is probably caused by the model assumptions of REIRL. Since it is based on MaxEnt IRL, it approximates the probability distribution over trajectories by inappropriately modeling the influence of the stochastic transition model. To verify that the differences of the mean log likelihood of the demonstrations under the trained models is significant, Welch's t-test [111] has been used. In the training task, the performance of MDCE SERD against the other approaches is statistically significant for sample set sizes that are larger than 3, while in the transfer task statistical significance is given at least for demonstration set sizes larger than 12. Fig. 4.12 (b) illustrates the expected Kullback-Leibler divergence from the real transition model P_R to the estimated one P_T under the joint probability distribution of states and actions from the ground truth model. Both REIRL and MDCE IRL use the M-estimated dynamics. Therefore, their transition models perform similarly. Since MDCE SERD optimizes the transition model simultaneously with the reward function, it is able to correct transition model errors. In contrast to the initial hypothesis about overfitting with small dataset, it seems that SERD is even beneficial, when only a small number of expert demonstrations is available. In addition, Fig. 4.12 (c) illustrates the expected Kullback-Leibler divergence from the real ground truth policy to the learned one, which indicates that SERD had the highest accuracy in learning a model that matches the real expert's behavior.

The second part of the experiment evaluates the generalizability of the estimates by transferring them to the environment in Fig. 4.11. Figure 4.12 (d) shows the median of the log likelihood of ground truth demonstrations under the stochastic policy, which is based on the estimated reward functions and transition models. Even in the new environment, MDCE SERD performs better than MDCE IRL or REIRL. In summary, MDCE SERD is able to learn generalizable rewards and dynamics from an expert executing a stochastic policy in a noisy environment without using an inductive bias via a prior. SERD has shown to result in models that explain the observed behavior better, while being sample efficient and yielding estimates with small variance.

PM SERD

When having knowledge about the model of the expert's stochastic policy, PM SERD can be beneficial for learning, since it allows applying SERD to custom policy models as long as they are differentiable. In this experiment, we use a Boltzmann distribution as a stochastic policy model, which is introduced in Sec. 4.4.2. Since MDCE SERD performed well without regularization, we first tried to learn with a uniform prior, which is an uninformed distribution that does not favor any particular solution. Even though PM SERD did outperform the other approaches in terms of the log likelihood of the ground truth demonstrations, its estimate of the dynamics was worse and the transfer task performance was poor, which indicates overfitting to the training task. By analyzing the estimates, it was obvious that the transition model in the forest did not match the real dynamics. Hence, PM SERD uses the parts of the transition model that are rarely or never observed to better match the policy to the demonstrations of the expert. This result confirms our belief that MDCE SERD generalizes better than PM SERD due to the influence of the stochastic policy on the gradient.

Therefore, we rerun experiments with regularization to prevent overfitting. The policy parameters and the feature weights were regularized by Gaussian priors, while the parameters of the dynamics used an entropy prior, which favors a high entropy of the transition model and thus causes the conditional distribution $P_{\boldsymbol{\theta}_{T_E}}(s' \mid s, a)$ to be flat. Fig. 4.13 summarizes the results. Similarly to the previous experiment of MDCE SERD, PM SERD outperforms the other approaches. The median log likelihood of demonstrations from the ground truth model exceeds the results of the other approaches. Furthermore, the proposed approach is very sample efficient and results in low variance estimates. Fig. 4.13 (b) shows that PM SERD learns models of the dynamics that are more accurate by further improving the initial transition model estimate. Entropy regularization was sufficient to achieve these results, even though this regularization type only prevents models of the environment's dynamics that become too close to deterministic. In addition, Fig. 4.13 (c) shows the expected Kullback-Leibler divergence of the policy, which indicates that



(a) Median log likelihood

(b) Expected KLD of the transition model

(c) Expected KLD of the policy

(d) Median log likelihood (transfer task)

Figure 4.13: Evaluation of PM SERD. (a) Median log likelihood with quartiles of demonstrations drawn from the true model under the estimated model. (b) Average Kullback-Leibler divergence between the true dynamics of the environment and the estimated one. (d) Median log likelihood with quartiles of demonstrations drawn from the true model under the estimated model in the transfer task.

PM SERD matches the experts policy more accurately. This is reasonable since REIRL and MDCE IRL make wrong model assumptions. The ground truth model computes an optimal Q-function and plugs a stochastic policy model on top of it. This corresponds to an agent that is able to predict optimal actions and Q-values, but with a noisy execution of the policy. In contrast, REIRL and MDCE IRL assume stochastic behavior in all hierarchies of the model via relaxations. For example, MDCE IRL uses the softened Q-function instead of the optimal one. Finally, Fig. 4.13 (d) illustrates the median log likelihood of demonstrations from the ground truth model in the transfer task under the learned model from the training task. As before, PM SERD has a higher performance than the other approaches, which shows that with appropriate priors PM SERD is able to generalize well.

Discussion

Both MDCE SERD and PM SERD are able to model the demonstrated behavior, by learning appropriate estimates of the environment's dynamics as well as accurate reward function that explain the observed behavior. Furthermore, the SERD-based approaches are more sample efficient and provide estimates with lower variances. This is probably caused by inaccurate initial estimates of the environment's dynamics, which are likely to be wrong if the number of samples is small. In contrast to IRL approaches, SERD is able to correct these inaccuracies during training.

A comparison of the two SERD approaches shows that PM SERD is more prone to overfitting. A possible explanation is the difference of the Q-gradient iteration. While the soft Q-gradient of MDCE SERD propagates expected gradients along the stochastic policy, PM SERD propagates them only along the optimum policy. Therefore, the converged soft Q-gradient of MDCE SERD considers a larger area of the state- and action-space. This fact serves as a regularization and makes the results more stable. However, optimizing the hyperparameters of the prior can help to overcome the burden of PM SERD. Then, it has the advantages of more flexibility in the policy model as well as a more efficient computation of the Q-gradient.

Chapter 5

Learning to Navigate in a Populated Hallway

With an increasing number of intelligent systems acting in populated environments, there is an emerging necessity for programming techniques that allow for efficient adjustment of the robot's behavior to new environments or tasks. Programming by demonstration can be a way to allow non computer scientists for adjusting the behavior of a system such as intelligent robots. A major burden in learning from demonstrations is that often both the underlying goal and the environment's dynamics are unknown. The proposed approaches for a Simultaneous Estimation of Rewards and Dynamics provide a theoretical advantage over the former IRL-based approaches with naive estimation of transition models and can therefore be beneficial in these cases. In the following, classical IRL algorithms, such as Maximum Discounted Causal Entropy IRL and Relative Entropy IRL, are compared with SERD for learning high-level navigation strategies in a realistic hallway navigation scenario solely from human expert demonstrations. We show that the theoretical advantage of SERD also pays off in practice by estimating better models of the environment's dynamics and explaining the expert's demonstrations more accurately.

In the last couple of years, intelligent systems such as mobile robots have been used in various populated environments. Well-known examples are the museum tour-guide robots RHINO [20] and MINERVA [100], the conference-attending robot GRACE [94],

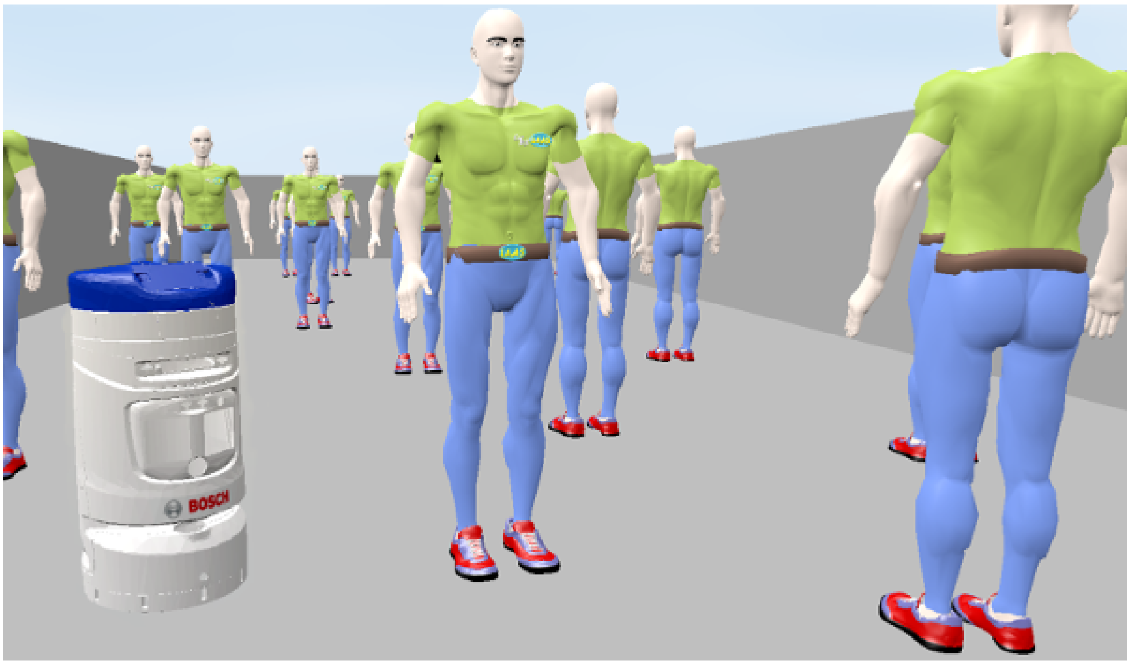


Figure 5.1: Navigating in populated environments requires a robot to weigh between several contradictory properties, such as being goal directed, complying with social rules, as well as keeping comfort and safety distances. This picture shows the implemented environment based on the MORSE simulator [26].

and the robot Obelix [60], which navigates through urban city-centers. As being accepted by humans is desirable for autonomous systems, the importance of their social-awareness increases. The term “social” typically refers to some type of interaction between individuals. However, different approaches and opinions exist in current research, as the term allows a scope for discussion.

Kirby [47] points out the need of robots to behave according to social norms and introduces a path planning approach based on a cost function that depends on social and task-related conventions. Morales *et al.*[72] model the side-by-side walking behavior with a utility function based on social and environment’s parameters. Lichtenthäler *et al.*[68] define the legibility as a prerequisite for user acceptance and measure its influence on the perceived safety or comfort. Since paying attention to the social acceptance is only necessary if humans are affected, Tipaldi *et al.*[101, 102] try to minimize the interference probability with people. This strategy is suitable in environments with a small number of humans, but navigating in crowdy areas requires different approaches. Therefore, Müller *et al.*[73] propose to follow pedestrians that walk into the direction of the goal, while Lerner *et al.*[61] suggest a data-driven approach, where trajectories from similar situations are chosen from a database. In contrast, Luber *et al.*[69] learn a set of dynamic motion prototypes from observations to compute cost maps using an any-angle A* algorithm.

A popular approach for modeling pedestrian motion is the social force model of Helbing and Molnar [36], which proposes that a human is a particle under the influence of attracting and repulsive forces. While this approach is widely used for simulating crowds, it sometimes causes unnatural trajectories when predicting trajectories for single individuals. Ferrer *et al.*[28] extend the social force approach by considering interactions between the robot and pedestrians.

Various approaches originate from the field of crowd simulation as well. Treuille *et al.*[105] propose a crowd model based on continuum dynamics. In contrast, Guy *et al.*[34] formulate an optimization algorithm to compute paths based on the principle of least effort. As situations affect human behaviors Kim *et al.*[46] introduce a method for simulating dynamic crowd behavior based on the General Adaption Syndrome Theory. Since uncertainty increases in crowdy environments, many approaches suffer from robots that are unable to move. To overcome this problem, Trautmann and Krause [104] propose a cooperative method, which models joint trajectories as dependent Gaussian processes.

Many of these approaches have in common that they require specification of fundamental parts of the resulting behavior, such as the controller, utility functions or features, when designing the system. However, these systems could also be deployed in different environments with varying tasks, which would require a computer scientist to redesign behavior, e.g. by specifying features or parameters. Imagine a robot that should solve transportation tasks in buildings. In Europe, the robot should drive on the right side and keep proper distances to people, due to cultural characteristics as well as social rules. In contrast, left-hand traffic could be more appropriate in other countries and social distances to pedestrians could also differ. To ensure a fast and easy deployment of robots in various environments, it is necessary to provide simple programming approaches for non-experts to parameterize new behaviors, goals or tasks. Several approaches have been proposed that allow teaching new behaviors to systems by demonstrating the task instead of programming it directly.

For example, Maximum Entropy IRL has been successfully applied to different tasks such as inferring decisions of taxi drivers by Ziebart *et al.*[117], learning a priority-adaptive navigation [38], or learning to navigate through crowded environments by Henry *et al.*[37], which shares similarities with the experiment in this work. Vasquez *et al.*[106] compare IRL algorithms and features for robot navigation in crowds. Kuderer *et al.*[54] predict human trajectories by learning their motion behavior in high-dimensional, continuous state spaces. This cooperative model is also useful for teaching mobile robots to navigate in populated environments jointly with arbitrary many pedestrians [55], by

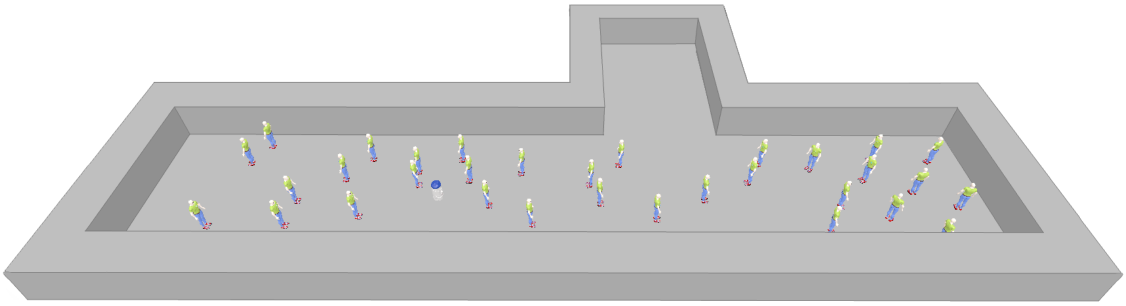
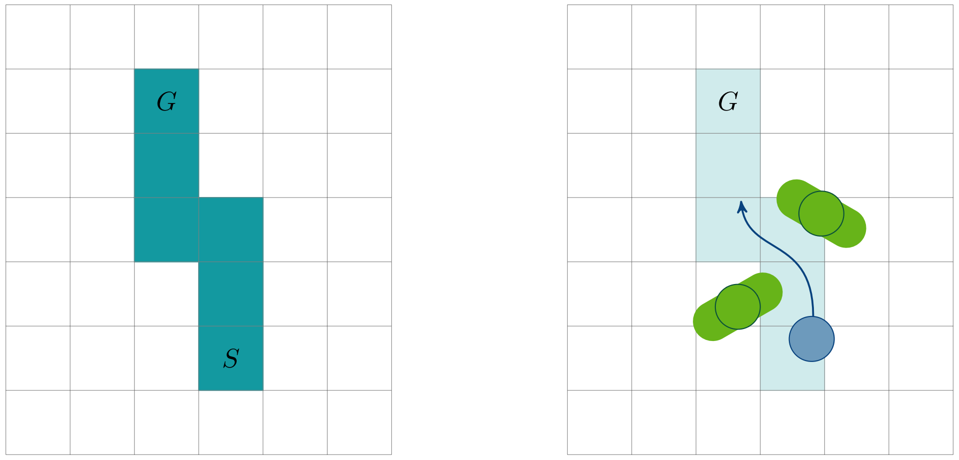


Figure 5.2: A simulated hallway scenario with humans following a social convention to walk on the right side implemented based on MORSE [26]. Humans are controlled by the Social Force model [36] and the robot is either controlled by a human to demonstrate desired behavior or by a controller following the optimal policy under the estimated reward function and transition model.

learning a probability distribution over the trajectories of all agents. Levine *et al.*[62] extend the MaxEnt IRL with a local approximation of the reward function to allow for locally optimal samples in continuous, high-dimensional state- and action-spaces.

Many of the outlined approaches repeatedly apply a Reinforcement Learning algorithm to find optimal policies as part of the IRL algorithm. As summarized in Chap. 4, this has several drawbacks such as the approaches often require the dynamics of the environment to be known, a simulator of the environment needs to be available for querying demonstrations, or appropriate heuristics need to be found that do not tamper with the results. In contrast, we have introduced approaches for a Simultaneous Estimation of Rewards and Dynamics (SERD) in Chap. 4, which to some degree overcome the need of an existing model of the environment's dynamics or the availability of a simulator. This is done by exploiting the fact that the expert's policy has been influenced by the reward function as well as the environment's dynamics, which allows to train both solely from expert demonstrations. The evaluation in Sec. 4.6 has shown that it is possible to estimate both reward functions and transitions models solely from expert demonstrations in toy examples. However, instead of learning from human demonstrations, we have used artificial expert demonstrations that stemmed from a stochastic policy of the same type of distribution that was modeled by the proposed approach. In contrast, real human behavior may differ from this type of distribution. Therefore, it is unclear whether learning rewards and dynamics from real human expert demonstrations is possible or whether traditional IRL approaches generalize better in such cases.

In this paper, we investigate the performance of IRL approaches for robot programming tasks from human demonstrations. We compare SERD Chap. 4, Maximum Discounted Causal Entropy IRL [16] and Relative Entropy IRL [18] for learning high-level navigation strategies in a densely populated hallway scenario. Further, we assume that only human demonstrations are available and the dynamics as well as the reward function are unknown beforehand. In general, it would be possible to evaluate all of the approaches based on a real world navigation task. However, the pedestrians are part of the environment and by changing the number of pedestrians, the individual people or parts of the experimental configuration, the environment's dynamics might also change. For example, the preferred walking speed or comfort distance varies across people. In order to enable reproducibility and thereby achieve comparable results, we have created a simulation of a populated hallway. Fig. 5.1 and Fig. 5.2 illustrate the simulation environment, which consists of a hallway populated by several pedestrians that are walking in circles according to right-hand traffic. In the following evaluation, we present that SERD outperforms the evaluated IRL approaches in terms of the modeled behavior as well as the learned dynamics of the environment in both the training and a transfer task. Furthermore, we show that more accurate dynamics can itself be beneficial, since they allow making more accurate predictions of the duration of task execution.



(a) High-Level navigation strategy

(b) Local motion plan

Figure 5.3: Due to the high complexity of planning problems in large, stochastic environments, motion planning is often decomposed into high-level strategies and local motion planning. This decomposition is based on the assumption that it is not necessary to plan in trajectory space until the goal as the environment is too stochastic. Hence, the high-level strategies omits certain available local aspects (e,g, dynamic objects) and outputs a less complex high-level plan to the goal, while the local motion planner takes into account the local aspects and the high-level path for creating the motion. In (a), a discrete high-level strategy is depicted as a blue path from a robots current position S towards a goal G, while omitting aspects such as dynamics objects or the real continuous position of the robot in the space. In contrast, (b) illustrates a local motion plan, which takes into account the two humans (indicated in green) and produces a driveable trajectory that guarantees to not collide with humans, while moving roughly in the direction of the global path from the high-level strategy.

5.1 Learning High-Level Navigation Strategies

Robot planning problems in large, populated environments are often highly complex, since robots suffer from partial observability (e.g. they cannot perceive humans that are occluded) and the environment's dynamics are highly stochastic. For example, models of joint human motion are not available and human behavior by itself is highly stochastic, which render it difficult to make accurate long-term predictions. Consequently, motion planning is often decomposed into high-level strategies and local motion planning as illustrated in Fig. 5.3. This is feasible, since it is not necessary to plan full exact trajectories that end up in the goal location, since they are going to be outdated after some timesteps due to stochastic and inaccurate transition models, as well as the unobservability of occluded dynamic objects. High-level strategies typically provide global directions without taking into account all aspects of the environment such as dynamic objects. In contrast, local motion planners consider dynamic objects and their movement and output low-level control commands.

The following section provides a discrete grid-based model for learning high-level navigation strategies that provide guidance for the local motion planner, which outputs directly executable actions. The resulting policy serves as a global strategy, which incorporates the estimated reward function as well as the trained dynamics of the environment including the influence of the underlying local motion planner. In the evaluation, we use this model for learning navigation strategies in densely populated hallways. Therefore, the estimated transition model incorporates the influence of walking humans on the motion planning of the robot. In the following, we describe the modeling of the environment with detailed explanations of the state- and action-space, the dynamics, and the reward features.

Table 5.1: Explanation of transitions

Transition	Meaning
0	Not moving
D	Moving into the preferred direction
R	Moving right wrt. the preferred direction
L	Moving left wrt. the preferred direction
B	Moving backwards

5.1.1 State- and action-space

The high-level navigation strategy guides the local planner through the environment. Therefore, we use a coarse discretization of a 2D map as a discrete state space on which the robot is navigating. Accordingly, the robots position in the map fully defines the state. In each state, the robot can pick from five different actions, which are moving into one of the four directions or waiting.

5.1.2 Modeling the dynamics

We assume that the robot is only able to move to adjacent cells. Therefore, the transitions $\Psi = \{0, D, R, B, L\}$, as specified in Tab. 5.1, are restricted to end up in one of the neighbouring states or the current one. Fig. 5.4 illustrates the available successor states if the robot is located in state 0. We assume that the dynamics are invariant of the preferred movement direction and modeled by parameters $\Theta = \{\theta_0, \theta_D, \theta_R, \theta_B, \theta_L\}$ of a Gibbs distribution:

$$P(\Psi = i) = \frac{\exp\left(\beta\theta_i\right)}{\sum_{j\in\Psi}\exp\left(\beta\theta_j\right)} \tag{5.1}$$

with β being a scaling parameter. The gradient of this transition model with respect to its parameters is:

$$\frac{\partial}{\partial\theta_j}P(\Psi = i) = \begin{cases} \beta\frac{\exp(\beta\theta_i)\sum_{k\in\Psi\setminus i}\exp(\beta\theta_k)}{\left(\sum_{k\in\Psi}\exp(\beta\theta_k)\right)^2} & \text{if } i = j \\ -\beta\frac{\exp(\beta(\theta_i+\theta_j))}{\left(\sum_{k\in\Psi}\exp(\beta\theta_k)\right)^2} & \text{if } i \neq j \end{cases} \tag{5.2}$$

The local dynamics may differ depending on the state or action. Therefore, it may be necessary to introduce several local transition models, such as one for moving in the same direction as the humans, for moving into the opposite direction, for moving

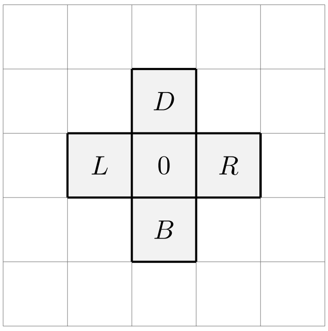


Figure 5.4: Visualization of the transition model.

in the orthogonal direction, or for standing still. Our assumption is that only expert demonstrations are available and that the transition model is unknown. However, many IRL approaches require the transition model to learn the reward function. Therefore, we initially estimate a model of the dynamics from expert demonstrations by using an M-estimator. However, this model may be inaccurate, since expert demonstrations omit undesired actions and hence large numbers of states and actions are rarely or never observed. Consequently, these inaccuracies might tamper with the reward estimate, if traditional IRL approaches are used.

5.1.3 Features

Previous work, e.g. Vasquez *et al.*[106], has studied features that are suitable for teaching robots to navigate. We use a subset of those features to reduce the number of possible ambiguities, while allowing all IRL algorithms to learn suitable rewards for explaining the expert demonstrations. The goal identifier $f_G \in \{0, 1\}$ is 1 at the goal and 0 otherwise. The social norm feature $f_S \in \{-1, 0, 1\}$ is 1 if the agent waits or if he satisfies the social norm of walking on the right side of the hallway. If the agent walks into the opposite direction, the feature is -1 and 0 otherwise. The last feature is the normalized inverse distance to walls $f_D \in [0, 1]$, since preferring one side of a hallway can also be explained by favoring small distances to walls. Fig. 5.5 visualizes these state and action dependent features in the hallway environment.

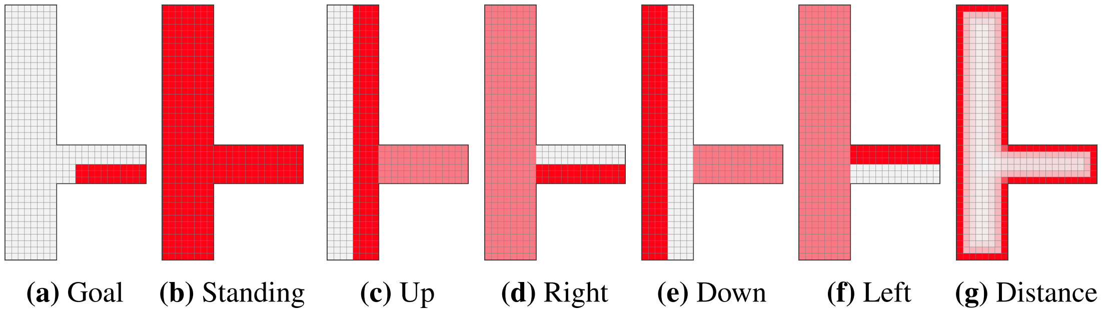

(a) Goal **(b)** Standing **(c)** Up **(d)** Right **(e)** Down **(f)** Left **(g)** Distance

Figure 5.5: Visualization of the reward function features that are used in the hallway navigation experiment. A darker shade of red indicates a higher feature value. The goal feature (a) is 1 at the goal and 0 otherwise. The social norm features (b), (c), (d), (e), and (f) are 1 if the agent waits or if he satisfies the social norm of walking on the right side of the hallway. If the agent walks into the opposite direction, the feature is -1 and 0 otherwise. As the feature values are action dependent, a feature plot has been created for each of the actions (Standing, Up, Right, Down, Left). The last feature is the normalized inverse distance to walls (g), which can be used to explain an expert's preference of walking on the side of a hallway or close to walls.

5.2 Experimental evaluation

In order to evaluate the IRL algorithms in a realistic scenario, we use a simulation of a densely populated hallway, which is illustrated in Fig. 5.6. We chose this scenario, as the transition model of the motion planner is stochastic and unknown, the resulting policy is non-trivial, and the discretized state is fully observable. In the simulation, humans permanently walk through the hallway, satisfying a social norm to walk on the right side. To synthesize human motion behavior, the Social Force model of Helbing and Monar [36] controls the simulated pedestrians. The goal of the evaluation is to analyze the estimates of the IRL approaches with respect to the resulting policy, the accuracy of the dynamics, as well as their generalization capabilities by inferring expert trajectories in new starting states. Consequently, the evaluation consists of two parts. In the training task, the robot starts randomly in one of the blue grid cells in the lower left of the hallway (Fig. 5.6) and its goal is to navigate into the green goal area. In the transfer task, the robot starts in one of the red cells, while the goal area stays the same. We implemented the scenario in ROS, using the movebase[1] as a local motion planner, and the MORSE simulator [26]. Therefore, the learned high-level strategy sends intermediate waypoints to the movebase. The environment dynamics consist of four different transition models: a transition model for moving with the human flow, a model for moving against the human flow, one for an orthogonal movement, and a transition model for choosing to stand still.

[1] http://wiki.ros.org/move_base

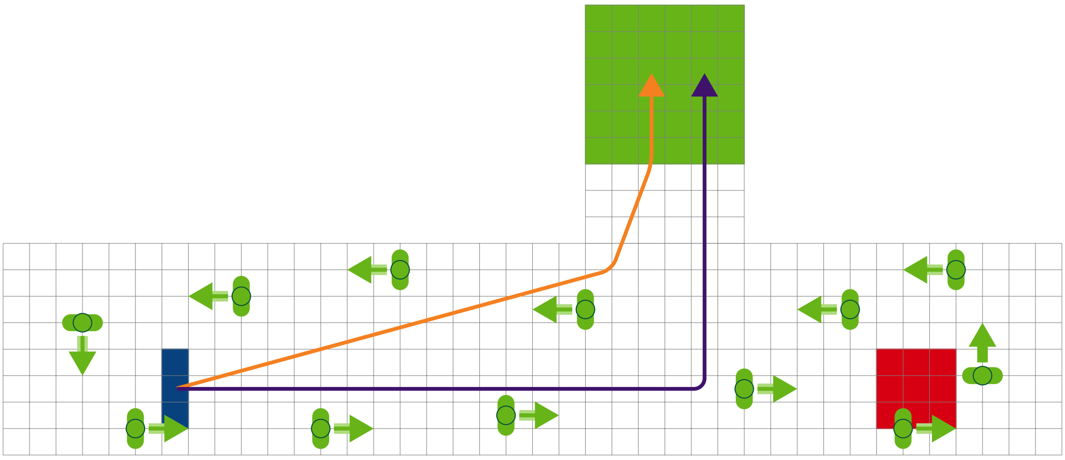
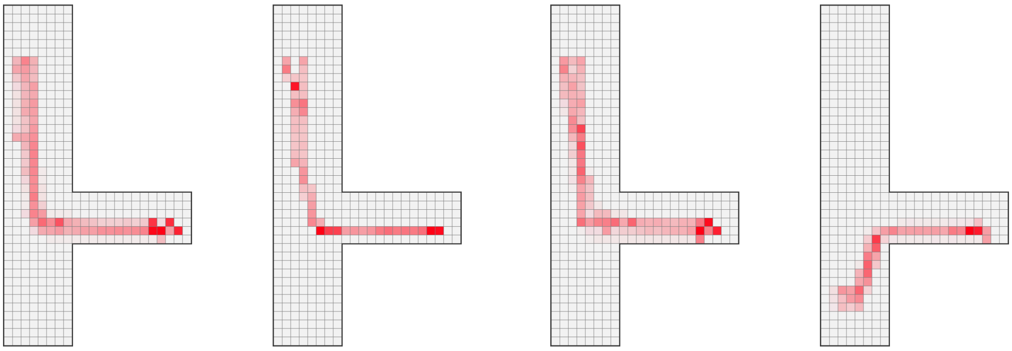


Figure 5.6: State space of the populated hallway scenario. The blue states indicate the robot's start positions in the training task, the red states depict the robot's initial states in the transfer task, and the green area is the goal zone. Simulated humans are colored in green with arrows indicating their movement direction. While the orange arrow indicates the shortest path to the goal, the human experts typically guided the robot along the violet path.



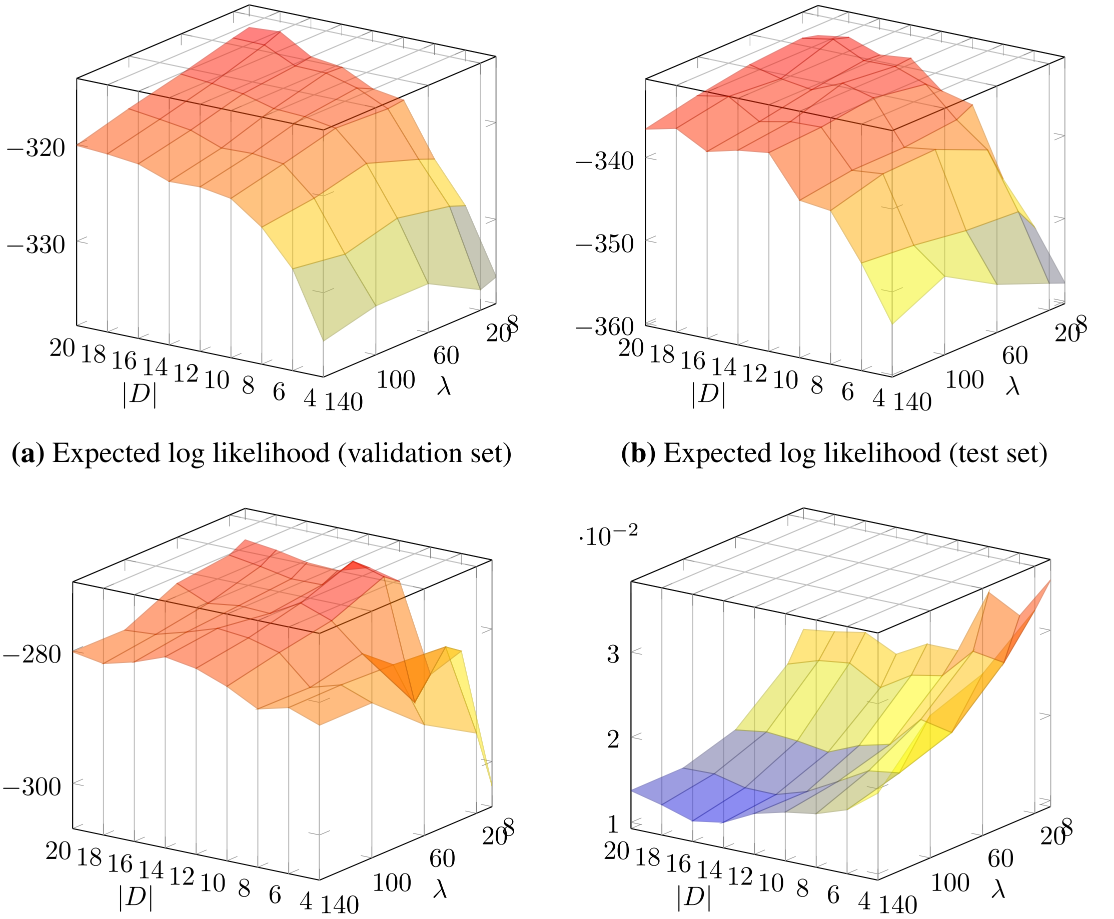
(a) Training Set **(b)** Validation Set **(c)** Test Set **(d)** Transfer Set

Figure 5.7: Visualization of the state visitation frequency of the different expert demonstration datasets for training (a), validation (b), test (c), and transfer (d). A darker shade of red indicates a higher state visitation frequency.

Human experts that were advised to guide the robot to the goal provided the demonstrations for training, validation, test and transfer. To achieve this, the robot had to move through the hallway, cross an intersection and enter the corridor. Even though the shortest way leads diagonally through the hallway, all human experts operated the robot to move on the right side of the hallway, staying in the human flow until they reached the intersection. Then, they guided the robot through the intersection to the goal area. Altogether, they have provided twenty training demonstrations from which random subsets are used for training to evaluate the influence of the dataset size. From these reduced training sets, $10\% - 20\%$ of the demonstrations are used for validation purposes such as early stopping or tuning hyperparameters. Furthermore, we have recorded ten additional test demonstrations for each the training as well as the transfer task to evaluate the estimates of the IRL approaches. Fig. 5.7 visualizes the state visitation frequency of the different datasets. While the state visitation frequencies of training, validation, and test are very similar, the behavior of the transfer task looks somehow different, since the robot started in a different state.

We compare MDCE IRL, REIRL, and MDCE SERD, considering the quality of the estimates in terms of the expected log likelihood of test demonstrations and the expected Kullback Leibler divergence. Since MDCE IRL can be trained by maximizing the causal entropy or by maximizing the log likelihood under the distribution specified in Eq. (3.30), both training methods are applied and compared to the other approaches. For initialization, we sample feature weights uniformly between $[-10, 10]$ before applying the IRL algorithms. MDCE IRL and REIRL need a transition model for learning the rewards. For SERD meaningful initial dynamics are beneficial. Therefore, we estimate dynamics from expert demonstrations while assuming a uniform prior. The SERD-based approach is further optimizing the transition model from this initial guess and might overfit to the data without proper regularization. Therefore, the application of λ-weighted KLD regularization (Sec. 4.5.3) should keep the trained model of the dynamics close to the initial guess. The additional hyper parameter λ allows to vary the penalty for transition model adjustments. For $\lambda \to 0$, the regularization term vanishes completely, while for $\lambda \to \infty$ MDCE SERD turns to MDCE IRL, since the regularizer does not allow any difference between the initial guess of the dynamics and the transition model estimate.

Fig. 5.8 (a) – (c) illustrate the performance of MDCE SERD measured by the expected



(a) Expected log likelihood (validation set)

(b) Expected log likelihood (test set)

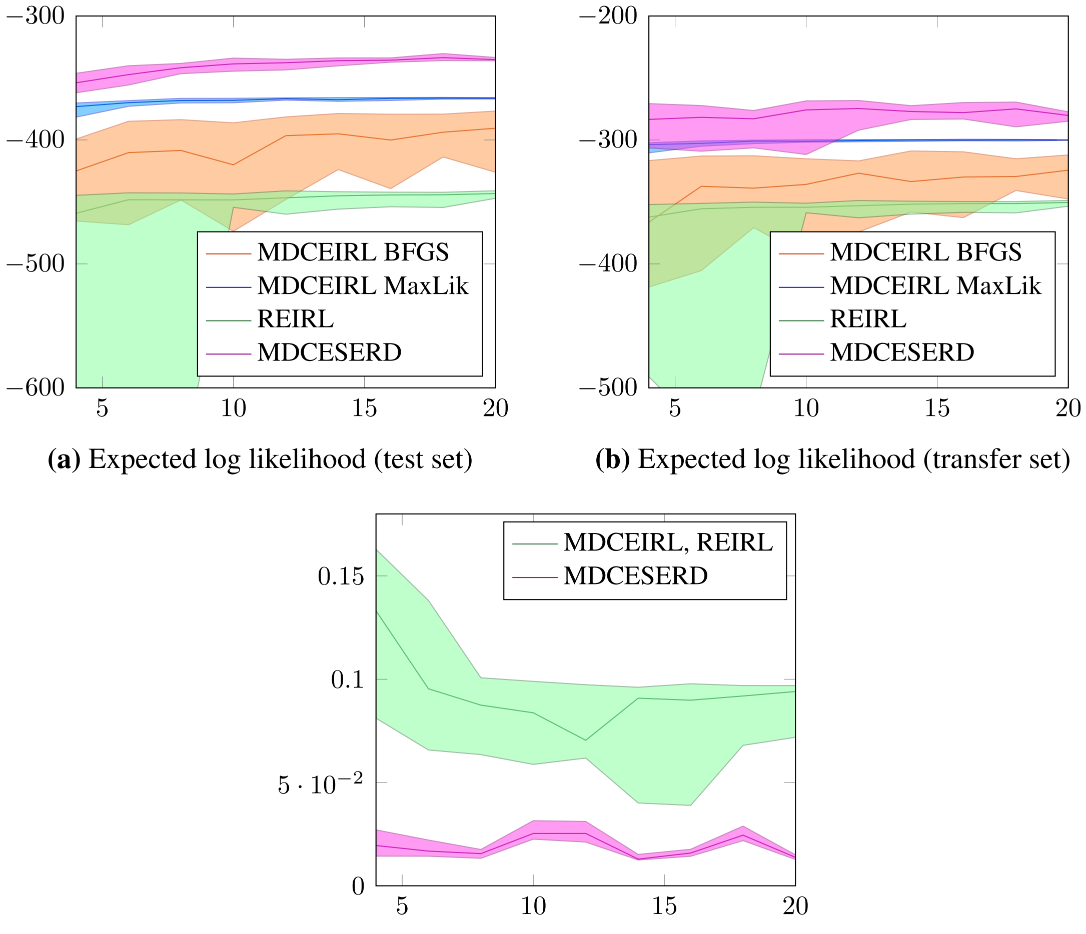
(c) Expected log likelihood (transfer set)

(d) KLD from estimated to real transition model

Figure 5.8: Results of hyperparameter optimization over different KLD regularization weights λ. The figures in (a), (b), and (c) illustrate the expected log likelihood of samples from different sets (validation, test, and transfer) over the sample set size as well as for different KLD regularization weights. (d) depicts the expected Kullback Leibler divergence $KLD(P||P_{\boldsymbol{\theta}})$ from the estimated model of the dynamics $P_{\boldsymbol{\theta}}$ to the more accurately estimated one from more samples P over the sample set size and the KLD regularization weight.

log likelihood of expert actions under the learned models on the validation, test, and transfer environment for different training set sizes $|D|$ and regularization weights λ. Fig. 5.8 (a) indicates that larger datasets increase performance, while larger regularization weights are only beneficial for training sets with a small number of demonstrations. With a smaller effect, this is also observable in the test set performance in Fig. 5.8 (b). The transfer set results in Fig. 5.8 (c) show a performance drop for small numbers of demonstrations and a small regularization weight. However, with more than 10 training samples, larger regularization was not beneficial anymore. Interestingly, datasets with 12 training samples achieve the best transfer set performance, which might be caused by early stopping interrupting the training process earlier than with 20 training samples and thus further stopping the model to overfit to the training task. Finally, Fig. 5.8 (d) illustrates the expected Kullback Leibler divergence between the true model (estimated from a larger sample set) and the one estimated by MDCE SERD. As in the log likelihood evaluation, a larger training set was increasing transition model performance (lower KLD). However, in contrast to the log likelihood evaluation, increasing the regularization weight is always improving transition model quality, which was not the case for the expected log likelihood under larger dataset sizes. This mismatch might be caused by real humans not fully acting as suboptimal decision makers according to an MDCE policy. Hence, during hyperparameter optimization it is a design choice whether to favor a more accurate transition model or policy.

Since, the goal of this experiment is teaching a robot to navigate in populated hallways, the optimal regularization weight is chosen based on the expected log likelihood on the validation set in Fig. 5.8 (a) for every dataset size. Even though that this might decrease the accuracy of the estimated transition model. Fig. 5.9 (a) illustrates the expected log likelihood of samples from the test set on multiple models trained with MDCE SERD, REIRL, and MDCE IRL with both maximum causal likelihood and maximum causal entropy optimization over varying training dataset sizes. MDCE SERD outperforms all other approaches followed by MDCE IRL with log likelihood optimization, MDCE IRL with maximum causal entropy optimization, and REIRL. MDCE IRL with log likelihood optimization and MDCE SERD optimize the log likelihood of the training demonstrations directly. Thus, they estimate proper reward functions under limited demonstrations by finding appropriate feature weights that cause the stochastic policy to match the observed behavior. The improvement of MDCE SERD over MDCE IRL with log likelihood optimization is caused by jointly optimizing the transition model and the reward function, which allows to further optimize the transition model by considering its influence on the experts behavior. In our experiment, the maximum causal likelihood optimization of MDCE IRL yields better results than the maximum causal entropy counterpart, even though that the traditional paper motivates the maximum causal entropy optimization [16] and it has been shown that both optimization procedures should be equivalent [119]. However, equivalence only holds if the correct transition model is known and the experts feature expectation can be estimated accurately, which is often not true in the limited data setting. This causes probabilistically bounded approximation errors as derived by Ziebart *et al.*[121]. We believe that the differing model of the probability distribution over trajectories as well as errors in approximating the feature expectation via importance



(a) Expected log likelihood (test set)

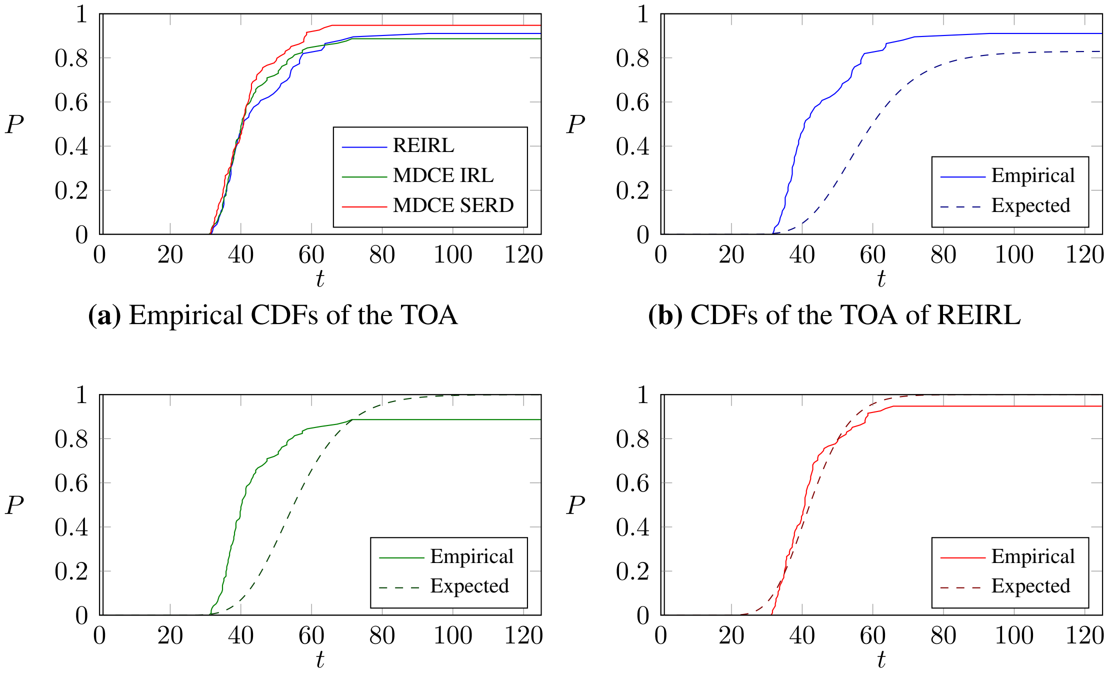
(b) Expected log likelihood (transfer set)

(c) Expected Kullback Leibler divergence $KLD(P||P_{\boldsymbol{\theta}})$

Figure 5.9: Results of MDCE SERD, MDCE IRL, and REIRL on training and test task over varying dataset sizes. The figures in (a) and (b) illustrate the median expected log likelihood with quartiles of samples from the test and transfer set, while (c) depicts the median expected Kullback Leibler divergence $KLD(P||P_{\boldsymbol{\theta}})$ with quartiles from the estimated model of the dynamics $P_{\boldsymbol{\theta}}$ to the more accurately estimated one from more samples P over the sample set size.

sampling causes the results of REIRL. For the transfer set evaluation in Fig. 5.9 (b), we have recorded ten expert demonstrations from the transfer task, in which the robot starts in the bottom right of the hallway. We sample the start state uniformly from one of the red cells in Fig. 5.6. Similar to the training task, MDCE SERD outperforms the other approaches followed by MDCE IRL, while REIRL yields worse results in explaining the expert demonstrations. This implies that even though MDCE SERD has more degrees of freedom during optimization, accurate, generalizable models can be trained without overfitting to the training task.

Fig. 5.9 (c) shows a measure for the accuracy of the estimated transition models. This measure is the expected Kullback-Leibler divergence $KLD(P||P_{\boldsymbol{\theta}})$ from the estimated model of the dynamics $P_{\boldsymbol{\theta}}$ to the more accurate transition model, which has been estimated from all existing demonstrations from the whole environment including additional ones from other tasks (e.g. moving against the human flow). We compute the expectation with respect of the joint probability of states and actions of the demonstrations from the training task. As a consequence, this metric specifies how accurate the transition



(a) Empirical CDFs of the TOA

(b) CDFs of the TOA of REIRL

(c) CDFs of the TOA of MDCE IRL

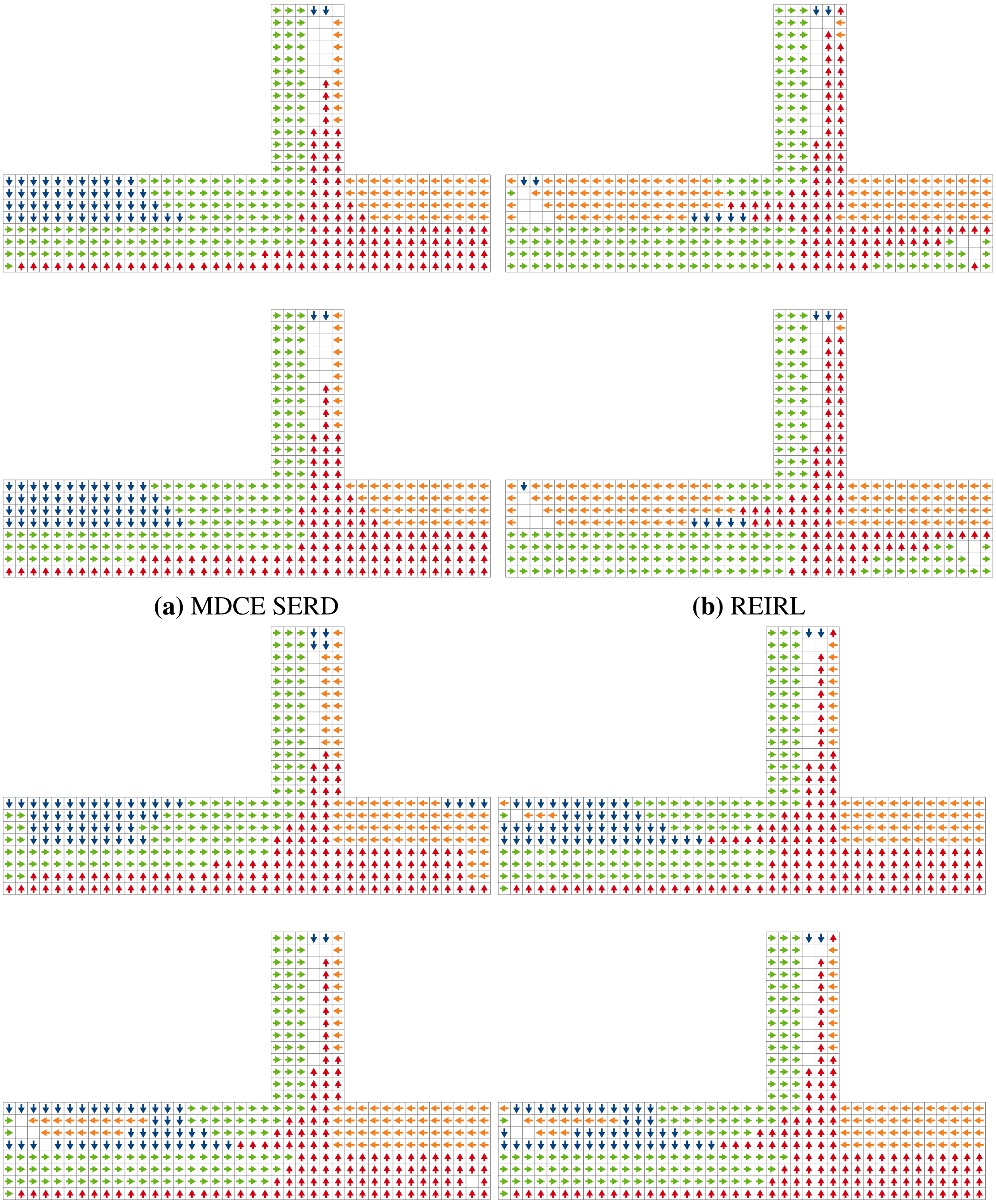
(d) CDFs of the TOA of MDCE SERD

Figure 5.10: Comparison between expected and estimated cumulative distribution functions (CDF) of the time of arrival (TOA). The expected CDF is directly computed based on the estimated transition model and the optimal policy of the learned models. The empirical CDF is approximated by applying the optimal policies in the simulator. All CDF's that do not increase up to 1 indicate that the robot sometimes is not able to reach the goal.

model is, while the KLD of more likely transitions is more strongly weighted. Both MDCE IRL and REIRL use the same transition model estimates and therefore share a plot. Their estimate of the environment's dynamics improves with an increasing number of available samples. However, for further decreasing the Kullback Leibler divergence, a lot more samples would be required as experts have goal-directed behavior and bias their demonstrations. Consequently, certain transitions are rarely observed. In contrast, MDCE SERD further optimizes the estimates of the dynamics and improves over the initialized one. Interestingly, it behaves almost constant, while the hyperparameter optimization in Fig. 5.8 (d) indicated possible improvements with increasing sample set sizes. This is caused by two contradicting objectives during hyperparameter optimization, which favored models with high log likelihoods over more accurate transition models. By changing this objective, one could further reduce the transition model mismatch.

As the human experts were guiding the robot to the goal, the following part of the evaluation focusses on the quality of the learned task. Therefore, we use all the resulting models from the different approaches based on the largest training dataset as high-level controllers in the environment. In every run of the experiment, the simulated robot is randomly spawning in one of the blue starting states of Fig. 5.6. Then, we apply the learned controllers to the robot until it either reaches the goal or exceeds seven minutes. Fig. 5.10 (a) illustrates the cumulative distribution function (CDF) of the times of arrival (TOA) in the goal area of the resulting optimal policies. The resulting times of arrival statistics of the different approaches are comparable, indicating that, in general, the SERD and IRL approaches were able to find meaningful estimates. However, the empirical CDFs show that MDCE SERD learns estimates, which guide the robot a little bit faster to the goal. It is important to note that the CDFs of the TOA are not increasing up to 1, which indicates that some estimates of the compared approaches do not model the desired behavior accurately enough to guide the robot to the goal area in seven minutes.

As indicated in Fig. 5.9 (c), the transition model estimates of MDCE SERD seem to be more accurate than the naive initial estimates. Hence, we can assume that more accurate dynamics result in more accurate estimates of the time of arrival. Fig. 5.10 (b) - (d) illustrate the CDFs of the expected time of arrival and the empirical one, which is generated by running simulations of the resulting optimal policies. While the empirical and the expected TOA CDFs differ strongly for MDCE IRL and REIRL, the expected TOA CDF of MDCE SERD is much more accurate and very similar to the empirical one. This gain in accuracy can be beneficial for TOA-based applications, e.g. tasks with strict time constraints.



(a) MDCE SERD

(b) REIRL

(c) MDCE IRL (Entropy optimization)

(d) MDCE IRL (Likelihood optimization)

Figure 5.11: Each of the sub-figures consists of two exemplary final policies for one of the evaluated approaches. All of the approaches learn reward functions and transition models that explain the experts trajectories well. While the MDCE SERD policies generalize well to states that have never or rarely been observed in the expert demonstrations, the baseline approaches REIRL and MDCE IRL (maximum entropy and maximum likelihood optimization) yield some deficiencies. For example, they might guide the robot into trapping states and thus will never reach the goal.

Finally, Fig. 5.11 (a) - (d) illustrate two of the learned optimal policies for each of the different methods. All of the approaches learned correct policies in the observed states from the training data and thus correctly operate the robot to drive on the right side of the hallway, staying in the human flow, and crossing at the intersection for moving to the goal area. However, differences between the policies can be seen, when inspecting the generalization to states that have never or rarely been observed in the expert demonstrations. For example, the upper left part of Fig. 5.11 (a) indicates that policies of MDCE SERD control the robot against the human flow, if only a few steps are required. For longer routes, it rather guides the robot into states, in which it can move with the human flow even though that this yields longer path lengths. In contrast, the policies based on reward functions trained with REIRL in Fig. 5.11 (b) wrongly generalize to unobserved states in the upper left and the lower right, e.g. they control the robot to move into the corner and to stay there. Hence, the robot can get stuck, if it accidentally ends up in one of these states. This might happen due to noise in the transition model or when adjusting the initial state during deployment. A similar behavior partially exists in the policies of MDCE IRL in Fig. 5.11 (c) and (d). Especially, in the upper left there exists a state, in which a robot might get stuck. However, the number of states, which will guide the robot into this trap, is smaller than in REIRL. Hence, these policies seem to be worse than the ones from MDCE SERD, but probably generalize better than the ones from REIRL. The upper policy in Fig. 5.11 (c) is similar to the ones of MDCE SERD. Consequently, it can be concluded that with an appropriate initialization (reward function and transition model) the MDCE SERD can also converge to suitable solutions.

5.3 Conclusions

In this section, we investigated the performance of IRL approaches for robot programming tasks from human demonstrations. Previous evaluations of SERD only focused on learning from non-human policies. Consequently, it was not clear whether the results would transfer to learning from human demonstrations, which might stem from stochastic policies that differ to the proposed ones. For this purpose, we have created a simulator of a densely populated hallway to allow for learning high-level navigation strategies. The evaluation shows improved performance of MDCE SERD, which incorporates the learning of the transition model into IRL. The computed models explain the demonstrations more accurately, which further results in better estimates of the time of arrival. However, MDCE IRL was able to estimate the reward function accurately, too, while REIRL sometimes failed to learn meaningful rewards. Therefore, the usage of SERD compared to MDCE IRL is especially beneficial if better transition models are useful. Such scenarios are, for example, high-level task planning problems, which need to infer the time of arrival of robots accurately. Additionally, it can be of interest to estimate the probability that a robot is not achieving the goal in a specific time. Future work could investigate the generalization capabilities of the estimated reward functions and transition models when transferring them to new environments. Furthermore, since SERD estimates the environment's dynamics based on expert demonstrations, research could focus on feature learning for transition models, which could even allow for training shared features between both the reward function and the dynamics.

Chapter 6

Conclusion

One of the key enablers for autonomous systems that should act in the real world is their ability for coping with varying tasks in changing environments. Consequently, there is a necessity for simple programming methods to adjust the system to new tasks, goals, and strategies. For this reason, the field of Imitation Learning offers approaches for learning policies from expert demonstrations. However, most existing approaches have several drawbacks, as they require a model of the environment's dynamics to be known or a simulator to be available. In many cases, this is not possible as models are often not available or it is not known in which environment the system is going to be deployed. Therefore, the goal of this work was to provide an approach for estimating both an expert's motivation and goal as well as the environment's dynamics solely from expert demonstrations to enable task adjustment of autonomous systems in changing environment.

In this thesis, we have uncovered several problems of recent IRL approaches that encounter when the environment is unknown. We have created a minimum example, which can illustrate these deficiencies, and specified a new problem class, called "Simultaneous Estimation of Rewards and Dynamics" (SERD). It formalizes problems, in which both the reward function as well as a transition model are unknown. Our proposed approaches aim for learning them simultaneously from expert demonstrations, since both have an influence on the expert's policy. If the expert misconceived the environment, he might act suboptimal. To account for these cases, SERD allows decoupling the real environment's dynamics and the expert's estimate of it. We have derived two solutions to the upper problem, which make different assumptions on the policy of the expert. The solutions consider that both dynamics and rewards have long-term influences on the policy and estimates them by learning reward functions and transition models that explain the observed demonstrations. This way, it even allows drawing conclusions on the transition model in unobserved states. In a first study, we have evaluated the two proposed approaches on idealized experiments and shown that they are in general able to solve the SERD problem in an idealized experiment, in which expert demonstrations stem from a ground truth model. We have shown that they improve both the transition model estimate as well as the accuracy, when predicting expert behavior given the learned models. As real humans might follow a different unknown policy type, a second study focused on evaluating whether SERD also yields accurate models, when learning from suboptimal, noisy demonstrations of human experts. Indeed, the SERD approach improved both the reward and transition model estimates as well as the generalization of the policy to states that were rarely or never observed. Furthermore, it shows that the proposed approach is able to learn high-level navigation strategies of robots in populated environments, in which pedestrians obey social rules or etiquette.

While this work provides first approaches for solving reward and transition model estimation from expert demonstrations, several other open points could not be addressed. As indicated in the minimum example, the SERD problem can be ambiguous, yielding no clear solution. In addition, the two proposed approaches are gradient-based methods of a non-convex objective and thus optimization can get stuck in local optima. Consequently, potential improvements could be better methods for global optimization, such as using multiple random initializations. When finding all equally accurate global optima, a human expert can choose the most accurate one by resolving the ambiguity using prior knowledge, e.g. certain transition models might not be meaningful. In this work, we cast SERD as a maximum a posteriori problem under a specified policy model. However, many tasks require learned transition models and policies to be robust for ensuring a safe deployment of autonomous systems in new environments. Given knowledge about a potential prior distribution over transition models, a fully Bayesian SERD approach could extract the posterior distribution over transition models. The resulting uncertainty is beneficial for improving safety, e.g. learning not to drive into states with high uncertainty by constrained reinforcement learning. Often prior knowledge is not available or hard to formalize. In these cases, the SERD problem could also be formulated as a full Maximum Entropy problem, which tries to find the least committed transition and policy model under the given information. In various domains and settings, these problems have shown to yield very robust models.

The proposed approaches and proofs in this thesis assume discrete, finite state- and action-spaces. However, many real world problems are continuous. Therefore, the proposed solutions for the SERD problem are not directly applicable. A promising future research direction would be extending the approaches to variants with efficient solutions for continuous problems. This will probably require relaxing certain assumptions on the optimality of the iterative gradient estimation.

A property of the problem class that has not been exploited, is that the expert's estimate of the environment's dynamics might differ from the real dynamics. In such cases, learning separate models allows to identify errors in the expert's assumptions about the world. This can enable studies of wrong human decision making under risk and might even allow analyzing the reasons for sub-optimality. Furthermore, several publications have shown that humans are not optimal decision makers, that they make errors, and that they are even biased. For really learning human objectives of their decision making, it is required to adjust the general IRL and SERD approaches to the types of sub-optimality and the errors that humans make.

Furthermore, SERD can be extended by several recent advances in IRL, such as learning in multi-agent environments, learning from trajectories that are scored by humans instead of generating optimal demonstrations directly, or meta-learning for easily and efficiently changing environments and tasks.

Abbreviations

BC	Behavioral Cloning
EM	Expectation maximization
Feature Expectation	Expected, cumulated, discounted feature counts
GAIL	Generative Adversarial Imitation Learning
GAN	Generative Adversarial Network
IOC	Inverse Optimal Control
IRL	Inverse Reinforcement Learning
LfD	Learning from Demonstration
MAP	Maximum a posteriori
MDP	Markov Decision Process
ME	Maximum Entropy
MCE	Maximum Causal Entropy
MDCE	Maximum Discounted Causal Entropy
PM	Policy Matching
POMDP	Partially Observable MDP
RE	Relative Entropy
SERD	Simultaneous Estimation of Rewards and Dynamics
SL	Supervised Learning

List of Figures

List of Tables

Bibliography

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML)*, New York, NY, USA, 2004. ACM.

[2] P. Abbeel, D. Dolgov, A. Y. Ng, and S. Thrun. Apprenticeship learning for motion planning with application to parking lot navigation. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1083–1090, Sept 2008.

[3] N. Abdo, H. Kretzschmar, L. Spinello, and C. Stachniss. Learning manipulation actions from a few demonstrations. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1268–1275, May 2013.

[4] 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, May 2009.

[5] R. Ash. *Basic Probability Theory*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 2012.

[6] M. Babes-Vroman, V. N. Marivate, K. Subramanian, and M. L. Littman. Apprenticeship learning about multiple intentions. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 897–904. Omnipress, 2011.

[7] M. Bain and C. Sammut. A framework for behavioural cloning. In *Machine Intelligence*, volume 15, pages 103–129, Oxford, UK, 1999. Oxford University.

[8] C. L. Baker, J. B. Tenenbaum, and R. Saxe. Bayesian models of human action understanding. *Advances in neural information processing systems (NIPS)*, 18:99, 2006.

[9] N. Baram, O. Anschel, and S. Mannor. Model-based adversarial imitation learning. In *NIPS Workshop on Deep Reinforcement Learning*, 2016.

[10] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, New York, NY, USA, 2012.

[11] U. Baumann, C. Gläser, M. Herman, and J. M. Zöllner. Predicting ego-vehicle paths from environmental observations with a deep neural network. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, 2018.

[12] U. Baumann, Y.-Y. Huang, C. Gläser, M. Herman, H. Banzhaf, and J. M. Zöllner. Classifying road intersections using transfer-learning on a deep neural network. In *Proc. of the IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[13] R. Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957.

[14] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability, 2nd Edition*. Athena Scientific, 2nd edition, July 2008.

[15] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[16] M. Bloem and N. Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. In *53rd IEEE Conference on Decision and Control (CDC)*, pages 4911–4916, 2014.

[17] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[18] A. Boularias, J. Kober, and J. Peters. Relative entropy inverse reinforcement learning. In *Proceedings of Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[19] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[20] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial intelligence*, 114(1-2):3–55, 1999.

[21] J. Choi and K. eung Kim. Map inference for bayesian inverse reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1989–1997, 2011.

[22] J. Choi and K.-E. Kim. Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, 12:691–730, July 2011.

[23] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[24] A. Cypher, D. C. Halbert, D. Kurlander, H. Lieberman, D. Maulsby, B. A. Myers, and A. Turransky. *Watch What I Do: Programming by Demonstration*. MIT Press, Cambridge, MA, USA, 1993.

[25] M. Dudík and R. E. Schapire. *Maximum Entropy Distribution Estimation with Generalized Regularization*, pages 123–138. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[26] G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, and S. Stinckwich. Simulating complex robotic scenarios with morse. In *SIMPAR*, pages 197–208, 2012.

[27] K. Eriksson, D. Estep, and C. Johnson. Lipschitz continuity. In *Applied Mathematics: Body and Soul: Volume 1: Derivatives and Geometry in IR3*, pages 149–164. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[28] G. Ferrer, A. Garrell, and A. Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, pages 1688–1694, 2013.

[29] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. In *NIPS Workshop on Adversarial Training*, 2016.

[30] C. Finn, S. Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 49–58, 2016.

[31] C. Fox. *An introduction to the calculus of variations*. Dover Publications, New York, 1987.

[32] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[33] M. Golub, S. Chase, and B. Yu. Learning an internal dynamics model from control demonstration. In *International Conference on Machine Learning (ICML)*, volume 28 of *JMLR Proceedings*, pages 606–614, 2013.

[34] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. C. Lin, and D. Manocha. Pledestrians: A least-effort approach to crowd simulation. In Z. Popovic and M. A. Otaduy, editors, *Symposium on Computer Animation*, pages 119–128. Eurographics Association, 2010.

[35] D. C. Halbert. *Programming by example*. PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1984.

[36] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[37] P. Henry, C. Vollmer, B. Ferris, and D. Fox. Learning to navigate through crowded environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 981–986, 2010.

[38] M. Herman, V. Fischer, T. Gindele, and W. Burgard. Inverse reinforcement learning of behavioral models for online-adapting navigation strategies. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3215–3222. IEEE, 2015.

[39] M. Herman, T. Gindele, J. Wagner, F. Schmitt, and W. Burgard. Simultaneous estimation of rewards and dynamics from noisy expert demonstrations. In *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 677–682, April 2016.

[40] M. Herman, T. Gindele, J. Wagner, F. Schmitt, and W. Burgard. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Artificial Intelligence and Statistics (AISTATS)*, 2016.

[41] M. Herman, T. Gindele, J. Wagner, F. Schmitt, C. Quignon, and W. Burgard. Learning high-level navigation strategies via inverse reinforcement learning: A comparative analysis. In *Australasian Joint Conference on Artificial Intelligence*, pages 525–534. Springer, 2016.

[42] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4565–4573, 2016.

[43] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[44] D. Huang, A. M. Farahmand, K. M. Kitani, and J. A. Bagnell. Approximate maxent inverse optimal control and its application for mental simulation of human interactions. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2673–2679, 2015.

[45] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106 (4):620–630, May 1957.

[46] S. Kim, S. J. Guy, D. Manocha, and M. C. Lin. Interactive simulation of dynamic crowd behaviors using general adaptation syndrome theory. In *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*, pages 55–62, 2012.

[47] R. Kirby. *Social Robot Navigation*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2010.

[48] K. Kitani, B. D. Ziebart, J. A. D. Bagnell, and M. Hebert. Activity forecasting. In *European Conference on Computer Vision*. Springer, October 2012.

[49] E. Klein, M. Geist, B. Piot, and O. Pietquin. Inverse reinforcement learning through structured classification. In *Advances in Neural Information Processing Systems (NIPS)*, Lake Tahoe (NV, USA), December 2012.

[50] S. Krantz and H. Parks. *The Implicit Function Theorem: History, Theory, and Applications*. The Implicit Function Theorem: History, Theory, and Applications. Birkhäuser, 2002.

[51] H. Kretzschmar, M. Kuderer, and W. Burgard. Learning to predict trajectories of cooperatively navigating agents. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[52] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research (IJRR)*, 2016.

[53] A. Y. Kruger. On fréchet subdifferentials. *Journal of Mathematical Sciences*, 116 (3):3325–3358, Jul 2003.

[54] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *Proc. of Robotics: Science and Systems (RSS)*, Sydney, Australia, 2012.

[55] M. Kuderer, H. Kretzschmar, and W. Burgard. Teaching mobile robots to cooperatively navigate in populated environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.

[56] M. Kuderer, S. Gulati, and W. Burgard. Learning driving styles for autonomous vehicles from demonstration. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2641–2646, Seattle, USA, 2015.

[57] A. Kuefler, J. Morton, T. A. Wheeler, and M. Kochenderfer. Imitating driver behavior with generative adversarial networks. In *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017.

[58] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

[59] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[60] R. Kummerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. A navigation system for robots operating in crowded urban environments. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.

[61] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[62] S. Levine and V. Koltun. Continuous inverse optimal control with locally optimal examples. In *Proc. of the 29th International Conference on Machine Learning (ICML)*, 2012.

[63] S. Levine, Z. Popovic, and V. Koltun. Feature construction for inverse reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1342–1350. Curran Associates, Inc., 2010.

[64] S. Levine, Z. Popovic, and V. Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 19–27. Curran Associates, Inc., 2011.

[65] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1):1334–1373, Jan. 2016.

[66] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[67] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 163–168, June 2011.

[68] C. Lichtenthäler, T. Lorenz, M. Karg, and A. Kirsch. Increasing perceived value between human and robots-measuring legibility in human aware navigation. In *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), 2012*, pages 89–94. IEEE, 2012.

[69] M. Luber, L. Spinello, J. Silva, and K. O. Arras. Socially acceptable robot navigation: A learning approach. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[70] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.

[71] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2308–2315. IEEE, 2010.

[72] L. Y. Morales Saiki, S. Satake, R. Huq, D. Glas, T. Kanda, and N. Hagita. How do people walk side-by-side?: Using a computational model of human behavior for a social robot. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308, New York, NY, USA, 2012. ACM.

[73] J. Müller, C. Stachniss, K. Arras, and W. Burgard. Socially inspired motion planning for mobile robots in populated environments. In *Proc. of the International Conference on Cognitive Systems (CogSys)*, pages 85–90, Karlsruhe, Germany, April 2008.

[74] G. Neu and C. Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proc. of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 295–302, 2007.

[75] G. Neu and C. Szepesvári. Training parsers by inverse reinforcement learning. *Machine Learning*, 77(2):303–337, 2009.

[76] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proc. of the Seventeenth International Conference on Machine Learning (ICML)*, pages 663–670, San Francisco, CA, USA, 2000.

[77] J. Penot. On the interchange of subdifferentiation and epi-convergence. *Journal of Mathematical Analysis and Applications*, 196(2):676 – 698, 1995.

[78] A. Phatak, H. Weinert, I. Segall, and C. N. Day. Identification of a modified optimal control model for the human operator. *Automatica*, 12(1):31 – 41, 1976.

[79] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems(NIPS)*, pages 305–313. Morgan-Kaufmann, 1989.

[80] M. L. Puterman. *Markov Decision Processes*. John Wiley & Sons, Inc., 2008.

[81] D. Ramachandran and E. Amir. Bayesian Inverse Reinforcement Learning. *Proc. of the 20th International Joint Conference on Artifical Intelligence (IJCAI)*, 51:2586–2591, 2007.

[82] N. Ratliff, D. Bradley, J. A. D. Bagnell, and J. Chestnutt. Boosting structured prediction for imitation learning. In *Advances in Neural Information Processing Systems (NIPS)*, Cambridge, MA, 2007. MIT Press.

[83] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. Maximum margin planning. In *Proc. of the 23rd International Conference on Machine Learning (ICML)*, pages 729–736, New York, NY, USA, 2006. ACM.

[84] C. A. Rothkopf and C. Dimitrakakis. Preference elicitation and inverse reinforcement learning. In *Joint European European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, Lecture Notes in Computer Science, pages 34–48. Springer, 2011.

[85] S. Russell. Learning agents for uncertain environments. In *Proc. of the eleventh annual conference on Computational learning theory*, pages 101–103. ACM, 1998.

[86] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, December 2002.

[87] C. Sammut, S. Hurst, D. Kedzier, and D. Michie. Learning to fly. In *Proc. of the Ninth International Conference on Machine Learning (ICML)*, pages 385–393. Morgan Kaufmann, 1992.

[88] F. Schmitt, H.-J. Bieg, D. Manstetten, M. Herman, and R. Stiefelhagen. Predicting lane keeping behavior of visually distracted drivers using inverse suboptimal control. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, June 2016.

[89] F. Schmitt, H.-J. Bieg, D. Manstetten, M. Herman, and R. Stiefelhagen. Exact maximum entropy inverse optimal control for modelling human attention switching and control. In *Proc. of the IEEE Conference on Systems, Man and Cybernetics (SMC)*, October 2016.

[90] F. Schmitt, H.-J. Bieg, M. Herman, and C. Rothkopf. I see what you see: Inferring sensor and policy models of human real-world motor behavior. In *AAAI Conference on Artificial Intelligence*, 2017.

[91] J. Schulman, J. Ho, C. Lee, and P. Abbeel. Learning from demonstrations through the use of non-rigid registration. In *Robotics Research*, pages 339–354, 2013.

[92] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[93] Z. Shao and M. J. Er. A review of inverse reinforcement learning theory and recent advances. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2012.

[94] R. G. Simmons, D. Goldberg, A. Goode, M. Montemerlo, N. Roy, B. Sellner, C. Urmson, A. C. Schultz, M. Abramson, W. Adams, A. Atrash, M. D. Bugajska, M. Coblenz, M. MacMahon, D. Perzanowski, I. Horswill, R. Zubek, D. Kortenkamp, B. Wolfe, T. Milam, and B. A. Maxwell. Grace: An autonomous robot for the aaai robot challenge. *AI Magazine*, 24(2):51–72, 2003.

[95] B. C. Stadie, P. Abbeel, and I. Sutskever. Third-person imitation learning. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017.

[96] G. Strang. *Linear algebra and its applications*. Thomson, Brooks/Cole, Belmont, CA, 2006.

[97] R. Sutton and A. Barto. *Reinforcement learning: An introduction*, volume 116. Cambridge Univ Press, 1998.

[98] U. Syed and R. E. Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems (NIPS)*, pages 1449–1456. Curran Associates, Inc., 2007.

[99] U. Syed, M. Bowling, and R. E. Schapire. Apprenticeship learning using linear programming. In *Proc. of the 25th International Conference on Machine Learning (ICML)*, pages 1032–1039, New York, NY, USA, 2008. ACM.

[100] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: a second-generation museum tour-guide robot. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 1999–2005 vol.3, 1999.

[101] G. D. Tipaldi and K. O. Arras. Planning problems for social robots. In *Proc. of the 21st International Conference on Automated Planning and Scheduling (ICAPS)*, 2011.

[102] G. D. Tipaldi and K. O. Arras. Please do not disturb! minimum interference coverage for social robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1968–1973, 2011.

[103] A. C. Y. Tossou and C. Dimitrakakis. Probabilistic inverse reinforcement learning in unknown environments. In *Proc. of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

[104] P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 797–803, 2010.

[105] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. *ACM Transactions on Graphics*, 25(3):1160–1168, July 2006.

[106] D. Vasquez, B. Okal, and K. O. Arras. Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, USA, 2014.

[107] M. C. Vroman. *Maximum likelihood inverse reinforcement learning*. PhD thesis, Rutgers, The State University of New Jersey, 2014.

[108] J. Wagner, V. Fischer, M. Herman, and S. Behnke. Multispectral pedestrian detection using deep fusion convolutional neural networks. In *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 509–514, April 2016.

[109] J. Wagner, V. Fischer, M. Herman, and S. Behnke. Learning semantic prediction using pretrained deep feedforward networks. In *25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, April 2017.

[110] J. Wagner, V. Fischer, M. Herman, and S. Behnke. Functionally modular and interpretable temporal filtering for robust segmentation. In *29th British Machine Vision Conference (BMVC)*, September 2018.

[111] B. L. Welch. The Generalization of 'Student's' Problem when Several Different Population Variances are Involved. *Biometrika*, 34(1/2):28–35, 1947.

[112] M. Wulfmeier, P. Ondruska, and I. Posner. Maximum entropy deep inverse reinforcement learning. In *Neural Information Processing Systems Conference (NIPS), Deep Reinforcement Learning Workshop*, volume abs/1507.04888, Montreal, Canada, 2015.

[113] A. Yamaguchi and C. G. Atkeson. Neural networks and differential dynamic programming for reinforcement learning problems. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[114] K. Yosida. *Functional Analysis*. Classics in Mathematics. Springer Berlin Heidelberg, 1995.

[115] Z. Zhou, M. Bloem, and N. Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. *IEEE Transactions on Automatic Control*, 63(9): 2787–2802, 2018.

[116] B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA, Dec 2010. AAI3438449.

[117] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proc. of the 10th international conference on Ubiquitous computing*, pages 322–331, 2008.

[118] B. D. Ziebart, A. Maas, J. A. D. Bagnell, and A. Dey. Maximum entropy inverse reinforcement learning. In *Proc. of the 23rd AAAI Conference on Artificial Intelligence*, July 2008.

[119] B. D. Ziebart, J. A. Bagnell, and A. K. Dey. Modeling interaction via the principle of maximum causal entropy. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 1255–1262, 2010.

[120] B. D. Ziebart, A. Dey, and J. A. D. Bagnell. Probabilistic pointing target prediction via inverse optimal control. In *International Conference on Intelligent User Interfaces (IUI 2012)*, February 2012.

[121] B. D. Ziebart, J. A. D. Bagnell, and A. Dey. The principle of maximum causal entropy for estimating interacting processes. *IEEE Transactions on Information Theory*, 59(4):1966–1980, February 2013.