Efficient Performance Bounds for Primal-Dual Reinforcement Learning from Demonstrations

Angeliki Kamoutsi [1] **Goran Banjac** [1] **John Lygeros** [1]

Abstract

We consider large-scale Markov decision processes with an unknown cost function and address the problem of learning a policy from a finite set of expert demonstrations. We assume that the learner is not allowed to interact with the expert and has no access to reinforcement signal of any kind. Existing inverse reinforcement learning methods come with strong theoretical guarantees, but are computationally expensive, while state-of-the-art policy optimization algorithms achieve significant empirical success, but are hampered by limited theoretical understanding. To bridge the gap between theory and practice, we introduce a novel bilinear saddle-point framework using Lagrangian duality. The proposed primal-dual viewpoint allows us to develop a model-free provably efficient algorithm through the lens of stochastic convex optimization. The method enjoys the advantages of simplicity of implementation, low memory requirements, and computational and sample complexities independent of the number of states. We further present an equivalent no-regret online-learning interpretation.

1. Introduction

Reinforcement learning (RL) is an area in machine learning with connections to control and optimization that has shown tremendous success in large-scale real-world applications, such as robotics, aritificial intelligence, cognitive autonomy, operations research, and healthcare (Tesauro & Kephart, 2002; Mnih et al., 2015; Chen et al., 2017; Vamvoudakis & Kokolakis, 2020). It studies the problem of learning to act optimally in a sequential decision-making problem, while interacting with an unknown environment (Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998).

In the standard RL setting a cost signal is given to instruct agents how to complete the desired task. However, oftentimes encoding preferences using demonstrations provided by an expert, is easier than designing a cost function (Pomerleau, 1991; Russell, 1998; Bagnell, 2015). In such cases, the goal of *learning from demonstrations* (LfD) or *imitation learning* (IL) is to learn a policy that achieves or even surpasses the performance of the expert policy.

A lot of methods have been proposed to solve the LfD problem. The most straightforward approach is *behavior cloning*, which casts the problem as a supervised learning problem, in which the goal is to learn a map from states to optimal actions (Pomerleau, 1991). Although behavior cloning is simple and easy to implement, the crucial i.i.d. assumption made in supervised learning is violated. As a result, the approach suffers from the problem of *cascading errors*, which is related to *covariate shift* (Bagnell, 2015; Ho et al., 2016). Later works (Ross et al., 2011) eliminate this distribution mismatch by formulating the problem as a no-regret online learning problem, but require interaction with the expert. On the contrary, in this paper, we consider the *batch* LfD scenario, i.e., the learner can observe only a finite set of expert demonstrations, is not allowed to query the expert for more data while training, and is not provided any reinforcement signal.

Inverse reinforcement learning (IRL) (Abbeel & Ng, 2004) is a prevalent approach to LfD. In this paradigm, the learner first infers the unknown cost function that the expert tries to minimize and then uses it to reproduce the optimal behavior. IRL algorithms do not suffer from the problem of cascading errors because the training takes place over entire expert trajectories, rather than individual actions. In addition, since the recovered cost function “explains” the expert behavior, they can easily generalize to unseen states or even new environments. Note however, that most existing IRL algorithms (Abbeel & Ng, 2004; Ratliff et al., 2006; Syed & Schapire, 2007; Neu & Szepesvári, 2007; Ziebart et al., 2008; Abbeel et al., 2008; Levine et al., 2010; 2011) are computationally expensive because they use RL as a subroutine.

On the other hand, under the assumption of a linearly parameterized cost class, one can frame the problem as a single convex program (Syed et al., 2008), bypassing the inter-

[1] Automatic Control Laboratory, ETH Zurich, Switzerland. Correspondence to: Angeliki Kamoutsi <kamoutsa@ethz.ch>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

mediate step of learning the cost function. Although the associated program can be solved exactly for small-sized MDPs, the approach suffers from the *curse of dimensionality*, making it intractable for large-scale problems.

The formulations and reasoning in (Syed et al., 2008) formed the ground and inspired later state-of-the-art policy optimization algorithms (Ho et al., 2016; Ho & Ermon, 2016). In particular, the authors in (Ho et al., 2016) consider the case of linearly paremeterized cost classes and present policy gradient algorithms, which are parallel to those proposed in (Williams, 1992; Schulman et al., 2015) for RL. Moreover, (Ho & Ermon, 2016) propose a generative adversarial imitation learning (GAIL) method for the case of nonlinear costs. They do so, by formulating the problem as minimax optimization and drawing a connection to generative adversarial networks (Goodfellow et al., 2014). In particular, GAIL solves IL with alternating updates of both policy and cost functions. These approaches are model-free and achieve significant empirical success in challenging benchmark tasks. However, in general the associated minimax problem is highly nonconvex-nonconcave and as a result remains hampered by limited theoretical understanding.

Indeed, the global convergence properties of alternating policy gradient schemes for the minimax formulation of GAIL have been studied only for linear (although infinite-dimensional) Markov decision processes (MDPs) and linear or linearizable costs (Cai et al., 2019; Zhang et al., 2020). It however remains unclear whether such neural policy gradient methods converge to the optimal policy or if they converge at all, for the case of nonlinear dynamics. As a result, provably efficient policy optimization schemes for the LfD problem beyond the linear setting remain largely unexplored.

Contributions. In an attempt to tackle this longstanding question, in this work we present a convex-analytic viewpoint of the LfD problem. To this end, we adopt the apprenticeship learning (AL) formalism which carries the assumption that the unknown true cost function can be represented as a weighted combination of some known basis functions, where the true unknown weights specify how different desiderata should be traded-off (Abbeel & Ng, 2004; Syed & Schapire, 2007; Syed et al., 2008; Ho & Ermon, 2016; Brown et al., 2020a). In particular, we make no restrictive assumptions on the MDP model (linearity or ergodicity).

Following the recent line of works (Chen et al., 2018; Lee & He, 2019; Wang, 2020; Bas-Serrano & Neu, 2020; Cheng et al., 2020; Jin & Sidford, 2020; Shariff & Szepesvári, 2020) on approximate linear programming (ALP) for large-scale MDPs, we formulate the LfD problem as a bilinear saddle-point problem in light of Lagrangian duality. We study primal-dual optimality conditions and prove relations between saddle points of the Lagrangian function and optimal solutions to the LfD problem. In particular, we show that under the expert optimality assumption, the set of solutions to the dual linear program (LP) characterizes the set of solutions to the inverse problem, i.e., the set of cost functions for which the expert is optimal. Moreover, in this case, we show that the expert policy, the true cost function, and the true optimal value function form a saddle point of the proposed Lagrangian.

Analogous to ALP, we obtain a linearly-relaxed saddle-point formulation by limiting our search to a linear subspace defined by a small number of features. We exhibit a formal link between approximate saddle points of the reduced Lagrangian and the optimality gap of our problem.

The aforementioned analysis lays theoretical foundations for a provably efficient stochastic primal-dual algorithm. By using linear function approximators, we propose a mirror-descent-based algorithm with a generative model and derive explicit probabilistic performance bounds on the quality of the extracted policy. A salient feature of the algorithm is that its sample complexity does not depend on the size of the state space but instead on the number of approximation features. We note however that our algorithm degrades with the approximation error due to the linear approximation architecture. We consider both the case of weak and strong linear features (Lakshminarayanan et al., 2018; Shariff & Szepesvári, 2020). Finally, we present an equivalent no-regret online-learning interpretation of our primal-dual algorithm. This kind of reduction has been studied up to now only for the online IL setting (Ross et al., 2011), where interaction with the expert is allowed.

Related works Our work is related to the LP approach to AL (Syed et al., 2008) and the theoretical analysis of GAIL made in (Cai et al., 2019; Chen et al., 2020a; Zhang et al., 2020). Unlike (Syed et al., 2008), who consider the case of tabular MDPs and known dynamics, we consider large-scale problems and only access to a simulator for the MDP model. We revisit the LP approach in (Syed et al., 2008) and focus instead on its saddle-point formulation due to its potential for scalable algorithms with theoretical guarantees (Cheng et al., 2020; Nachum & Dai, 2020). The authors in (Cai et al., 2019) study the global convergence properties of GAIL for the linear quadratic regulator problem by extending the results in (Fazel et al., 2018), while the authors in (Zhang et al., 2020) consider the case of infinite-dimensional linear MDPs and linearizable cost functions. Similar to (Wang et al., 2020) the policies and cost functions are approximated by overparameterized ReLU neural networks. In our work, we consider general nonlinear MDPs without any restrictive assumption such as linearity or ergodicity, and employ linear function approximators. The authors in (Chen et al., 2020a) study the convergence and generalization of GAIL for general MDPs. However they only prove convergence to a stationary point. On the contrary, our algorithm produces a nearly-optimal policy.

Our work builds upon a vast line of works on ALP for forward RL (De Farias & Van Roy, 2003; Abbasi-Yadkori et al., 2014; Chen et al., 2018; Lakshminarayanan et al., 2018; Mohajerin Esfahani et al., 2018; Wang, 2020; Lee & He, 2019; Bas-Serrano et al., 2020; Cheng et al., 2020; Jin & Sidford, 2020; Shariff & Szepesvári, 2020; Bas-Serrano & Neu, 2020). The LP approach to MDPs, which dates back to (Manne, 1960; Hernández-Lerma & Lasserre, 1996; Borkar, 1988), has recently gained traction as an alternative to dynamic programming techniques, for its advantage to lead to problem formulations that are directly amenable to modern large-scale stochastic optimization methods. Moreover, this optimization-based approach can tackle unconventional problems involving additional safety constraints or secondary costs, where traditional dynamic programming techniques are not applicable (Hernández-Lerma et al., 2003; Dufour & Prieto-Rumeau, 2013; Shafieepoorfard et al., 2013). In this paper, we develop an ALP framework for the LfD problem. While the forward policy optimization problem tries to minimize the total expected cost, the LfD problem is a minimax problem that tries to match the expert across a given cost class. In our saddle-point formulation the cost function is not fixed but is itself a decision variable. Thus, the variation of the cost function during the algorithm continuously changes the Lagrangian making the analysis of the problem more challenging.

2. Preliminaries and Problem Setup

2.1. Basic Definitions and Notations

We denote by $\mathbf{R}^n_+$ and $\mathbf{R}^n_{++}$ the sets of n-dimensional vectors with nonnegative and positive real elements, respectively. For a matrix $\mathbf{A} \in \mathbf{R}^{m\times n}$, its p-norm is defined by $\|\mathbf{A}\|_p \triangleq \sup\{\|\mathbf{A}\mathbf{x}\|_p \mid \|\mathbf{x}\|_p = 1\}$. For vectors $\mathbf{x}$ and $\mathbf{y}$, we denote by $\langle \mathbf{x}, \mathbf{y}\rangle$ the usual inner product. Moreover, $\mathbf{x} \leq \mathbf{y}$ denotes elementwise inequality, i.e., $x_i \leq y_i$ for all i. We use $\mathbf{1}$ and $\mathbf{0}$ to denote vectors with all elements equal to one and zero, respectively. The set of probability distributions on a finite set $\mathcal{S}$ is denoted by $\Delta_{\mathcal{S}}$, i.e., $\Delta_{\mathcal{S}} \triangleq \{\mathbf{p} \in \mathbf{R}^{|\mathcal{S}|}_+ \mid \sum_{s\in\mathcal{S}} p(s) = 1\}$, where $|\mathcal{S}|$ is the cardinality of $\mathcal{S}$. Sums spanning over the spaces $\mathcal{X}$ and $\mathcal{A}$ will be simply denoted by $\sum_x$ and $\sum_a$, respectively. For $\boldsymbol{\mu}, \boldsymbol{\mu}' \in \Delta_{\mathcal{X}}$ their *Kullback–Leibler divergence* is given by $\mathrm{KL}(\boldsymbol{\mu}\|\boldsymbol{\mu}') \triangleq \sum_x \mu(x) \log \frac{\mu(x)}{\mu'(x)}$. For a nonempty closed convex set Θ, the Euclidean projection of $\mathbf{x}$ onto Θ is given by $\Pi_\Theta(\mathbf{x}) \triangleq \arg\min_{\mathbf{y}\in\Theta} \|\mathbf{x} - \mathbf{y}\|_2$.

2.2. Reinforcement Learning

A finite MDP is given by a tuple $(\mathcal{X}, \mathcal{A}, P, \boldsymbol{\nu}_0, \mathbf{c}, \gamma)$, where $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{X}\times\mathcal{A} \to \Delta_{\mathcal{X}}$ is the transition law, $\boldsymbol{\nu}_0 \in \Delta_{\mathcal{X}}$ is the initial state distribution, $\mathbf{c} \in [-1, 1]^{|\mathcal{X}||\mathcal{A}|}$ is the one-stage cost, and $\gamma \in (0, 1)$ is the discount factor. We focus on problems where $\mathcal{X}$ and $\mathcal{A}$ are too large to be enumerated.

The MDP models a controlled discrete-time stochastic system with initial state $x_0 \sim \boldsymbol{\nu}_0$. At each round t, if the system is in state $x_t = x \in \mathcal{X}$ and the action $a_t = a \in \mathcal{A}$ is taken, then a cost $c(x, a)$ is incurred, and the system transitions to the next state $x_{t+1} \sim P(\cdot|x, a)$.

A *stationary Markov policy* is a map $\pi\colon \mathcal{X} \to \Delta_{\mathcal{A}}$, and $\pi(a|x)$ denotes the probability of choosing action a, while being in state x. We denote the space of stationary Markov policies by Π_0.

The *value function* $\mathbf{V}^\pi_{\mathbf{c}} \in \mathbf{R}^{|\mathcal{X}|}$ of π, given a cost $\mathbf{c}$, is defined by $V^\pi_{\mathbf{c}}(x) \triangleq (1-\gamma)\, \mathbf{E}^\pi_x \left[\sum_{t=0}^\infty \gamma^t c(x_t, a_t)\right]$, where $\mathbf{E}^\pi_x$ denotes the expectation with respect to the trajectories generated by π starting from $x_0 = x$.

The goal of RL is to solve the following optimal control problem

$$\rho^\star_{\mathbf{c}} \triangleq \min_{\pi\in\Pi_0} \rho_{\mathbf{c}}(\pi), \qquad (\mathbf{RL}_{\mathbf{c}})$$

where $\rho_{\mathbf{c}}(\pi) = \langle \boldsymbol{\nu}_0, \mathbf{V}^\pi_{\mathbf{c}}\rangle$ is the *total expected cost* of π.

For every policy π, we define the *normalized state-action occupancy measure* $\boldsymbol{\mu}_\pi \in \Delta_{\mathcal{X}\times\mathcal{A}}$, by $\boldsymbol{\mu}_\pi(x, a) \triangleq (1 - \gamma) \sum_{t=0}^\infty \gamma^t\, \mathbf{P}^\pi_{\boldsymbol{\nu}_0}[x_t = x, a_t = a]$, where $\mathbf{P}^\pi_{\boldsymbol{\nu}_0}[\cdot]$ denotes the probability of an event when following π starting from $x_0 \sim \boldsymbol{\nu}_0$. The occupancy measure can be interpreted as the discounted visitation frequency of state-action pairs. This allows us to write $\rho_{\mathbf{c}}(\pi) = \langle \boldsymbol{\mu}_\pi, \mathbf{c}\rangle$.

The *optimal value function* $\mathbf{V}^\star_{\mathbf{c}} \in \mathbf{R}^{|\mathcal{X}|}$ is defined by $V^\star_{\mathbf{c}}(x) \triangleq \min_{\pi\in\Pi_0} V^\pi_{\mathbf{c}}(x)$. For clarity, when the cost $\mathbf{c}$ is a parameterized function, written as $\mathbf{c}_{\mathbf{w}}$ for a parameter vector $\mathbf{w}$, we will replace $\mathbf{c}$ by $\mathbf{w}$ in the notation of the aforementioned quantities, e.g., $\rho_{\mathbf{w}}(\pi)$, $\mathbf{V}^\pi_{\mathbf{w}}$, etc.

2.3. Learning from Demonstrations

The goal of LfD is to learn a policy that outperforms the expert policy π_{E} for an unknown true cost function $\mathbf{c}_{\text{true}}$. We assume that the learner is given only a finite set of truncated expert sample trajectories and is not allowed to interact or query the expert for more data while training.

Although the MDP model is not known, we assume access to a *generative-model oracle* which, given a state-action pair (x, a), outputs the next state $x' \sim P(\cdot|x, a)$. Moreover we can sample $x_0 \sim \boldsymbol{\nu}_0$. This is also known as the simulator-defined MDP (Szörényi et al., 2014; Taleghan et al., 2015).

To address the LfD problem, we adopt the AL formalism (Abbeel & Ng, 2004; Syed et al., 2008; Ho et al., 2016; Ho & Ermon, 2016), which carries the assumption that $\mathbf{c}_{\text{true}}$ belongs to a class of cost functions $\mathcal{C}$. We then seek a policy that performs better than the expert across $\mathcal{C}$ by solving the following minimax optimization problem

$$\alpha^\star \triangleq \min_{\pi\in\Pi_0} \max_{\mathbf{c}\in\mathcal{C}} \rho_{\mathbf{c}}(\pi) - \rho_{\mathbf{c}}(\pi_{\text{E}}). \qquad (\mathbf{LfD}_{\pi_{\text{E}}})$$

Equivalently, we can write $\alpha^\star = \min_\pi \delta_{\mathcal{C}}(\pi, \pi_{\text{E}})$, where $\delta_{\mathcal{C}}(\pi, \pi_{\text{E}}) \triangleq \max_{\mathbf{c}\in\mathcal{C}} \big(\rho_{\mathbf{c}}(\pi) - \rho_{\mathbf{c}}(\pi_{\text{E}})\big)$ denotes the $\mathcal{C}$-distance between π and π_{E} (Ho et al., 2016; Chen et al., 2020a; Zhang et al., 2020). An optimal solution π_{A} to $(\mathbf{LfD}_{\pi_{\text{E}}})$ is called an *apprentice policy* and satisfies $\rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{A}}) \le \rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{E}}) + \alpha^\star$ with $\alpha^\star$ being always nonpositive.

Intuitively, the cost class $\mathcal{C}$ distinguishes the expert from other policies. The maximization in $(\mathbf{LfD}_{\pi_{\text{E}}})$ assigns high total cost to non-expert policies and low total cost to π_{E} (Ho et al., 2016), while the minimization aims to find the policy that matches the expert as close as possible with respect to the $\mathcal{C}$-distance.

In this work, we assume that the true cost function can be represented as a convex combination of some known features, where the true unknown weights specify how different desiderata should be traded-off. In particular, we consider the following cost function class (Syed & Schapire, 2007; Syed et al., 2008; Ho et al., 2016)

$$\mathcal{C} = \mathcal{C}_{\text{conv}} \triangleq \{\mathbf{c}_{\mathbf{w}} \triangleq \sum_{i=1}^{n_c} w_i \mathbf{c}_i \mid w_i \ge 0,\ \sum_{i=1}^{n_{\mathbf{c}}} w_i = 1\},$$

where $\{\mathbf{c}_i\}_{i=1}^{n_{\mathbf{c}}} \subset \mathbf{R}^{|\mathcal{X}||\mathcal{A}|}$ are fixed cost vectors, such that $\|\mathbf{c}_i\|_\infty \le 1$ for all $i = 1, \dots, n_{\mathbf{c}}$. In this case, $\delta_{\mathcal{C}}(\pi, \pi_{\pi_{\text{E}}}) = \max_{i\in[n_c]} \big(\rho_{\mathbf{c}_i}(\pi) - \rho_{\mathbf{c}_i}(\pi_{\text{E}})\big)$.

Note that this assumption is not necessarily restrictive as usually in practice the true cost function depends on just a few key properties, but the desirable weighting is unknown (Abbeel & Ng, 2004). Moreover, these features can be arbitrarily complex nonlinear functions and can be obtained via unsupervised learning from raw state observations (Brown et al., 2020b; Chen et al., 2020b).

We highlight that one can consider other linearly parameterized cost classes, e.g., $\mathcal{C}_{\text{lin}} = \{\sum_{i=1}^{n_{\mathbf{c}}} w_i \psi_i \mid \|w\|_2 \le 1\}$ (Abbeel & Ng, 2004), leading to similar reasoning and analysis.

2.4. Technique Overview

Before diving into technical details, we provide a technique overview by formalizing the main building block of our analysis, as described informally in the introduction. Figure 1 illustrates the big picture behind our reduction.

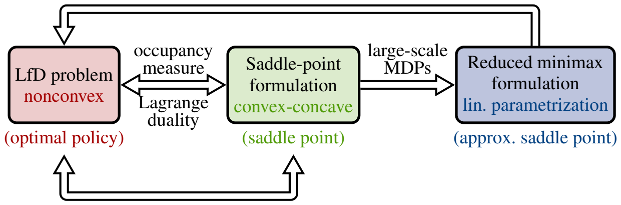

Figure 1. Main building blocks of our analysis.

The main difficulty in $(\mathbf{LfD}_{\pi_{\text{E}}})$ comes from its nonconvex-nonconcave structure. In Section 3, by using Lagrangian duality, we handle this difficulty by transforming the original nonconvex-nonconcave optimization program $(\mathbf{LfD}_{\pi_{\text{E}}})$ into the bilinear saddle-point problem (1). The new decision variables are occupancy measures $\boldsymbol{\mu}$, value functions $\mathbf{u}$ and cost weights $\mathbf{w}$. The two formulations are equivalent, since the primal optimizer $\boldsymbol{\mu}_{\text{A}}$ directly maps to that of the original problem π_{A} and vice versa. Moreover, under the assumption of expert optimality the triplet $(\boldsymbol{\mu}_{\pi_{\text{E}}}, \mathbf{V}^\star_{\mathbf{w}_{\text{true}}}, \mathbf{w}_{\text{true}})$ is a saddle-point of (1).

For large-scale MDPs the saddle-point formulation (1) is intractable. To mitigate this dificulty, in Section 4 we consider the reduced saddle-point problem (2) by introducing parameterized families $\{\boldsymbol{\mu}_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta\}$ and $\{\mathbf{u}_{\boldsymbol{\lambda}} \mid \boldsymbol{\lambda} \in \Lambda\}$, for appropriately chosen parameter sets Θ and Λ. We then relate the saddle-point residual $\epsilon_{\text{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w})$ (3) of an approximate saddle-point $(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w})$ to the suboptimality gap of the extracted policy π for the original $(\mathbf{LfD}_{\pi_{\text{E}}})$. In particular we show that $\delta_{\mathcal{C}}(\pi_{\boldsymbol{\theta}}, \pi_{\text{E}}) - \delta_{\mathcal{C}}(\pi_{\text{A}}, \pi_{\text{E}}) \le k\epsilon_{\text{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) + \varepsilon_{\text{approx}}$, where $\varepsilon_{\text{approx}}$ is a measure of expressivity of the function approximators and $k \in \{1, 3\}$. In this way, it is apparent that $(\mathbf{LfD}_{\pi_{\text{E}}})$ is decoupled in two parts: minimization of $\epsilon_{\text{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w})$ and function approximation.

In Section 5, we choose linear function approximators (LFA) to preserve the convexity-concavity of the Lagrangian and design a scalable algorithm with sublinear convergence rate to a neighborhood of π_{A}, where the representation power of LFA determines the convergence error. In particular, we present a variance reduced stochastic mirror descent algorithm (Nemirovski et al., 2009) under local norms and coordinate-wise gradient estimators (Carmon et al., 2019; Jin & Sidford, 2020) to solve the $(\mathbf{LfD}_{\pi_{\text{E}}})$ problem with a generative-model oracle.

3. The Convex Optimization View

In this section, we revisit the LP approach to the LfD problem (Syed et al., 2008) and introduce a novel linear duality framework, which will be later exploited to lay theoretical foundations for an efficient stochastic primal-dual algorithm. The details on the formulations and the proofs of this section can be found in Appendix A.

3.1. Primal-Dual Linear Programming Formulations

Assuming that $\mathcal{C} = \mathcal{C}_{\mathrm{conv}}$, one can frame the $(\mathbf{LfD}_{\pi_{\mathrm{E}}})$ problem as an LP over occupancy measures (Syed et al., 2008). We first review briefly the rationale behind the formulation.

The set of occupancy measures can be characterized in terms of linear constraint satisfaction. To this aim, let $\mathfrak{F} \triangleq \left\{\boldsymbol{\mu} \in \mathbf{R}^{|\mathcal{X}||\mathcal{A}|} \mid \mathbf{T}_\gamma \boldsymbol{\mu} = \boldsymbol{\nu}_0,\ \boldsymbol{\mu} \geq \mathbf{0}\right\}$, where $\mathbf{T}_\gamma \colon \mathbf{R}^{|\mathcal{X}||\mathcal{A}|} \to \mathbf{R}^{|\mathcal{X}|}$ is a linear operator given by $\mathbf{T}_\gamma \boldsymbol{\mu} = \frac{1}{1-\gamma}(\mathbf{B} - \gamma \mathbf{P})^\intercal \boldsymbol{\mu}$. Here, $\mathbf{P}$ is the vector form of P, i.e., $P_{(x,a),x'} \triangleq P(x'|x,a)$, and $\mathbf{B}$ is a binary matrix defined by $B_{(x,a),x'} \triangleq 1$ if $x = x'$, and $B_{(x,a),x'} \triangleq 0$ otherwise.

Proposition 1. *(Puterman, 1994) It holds that, $\mathfrak{F} = \left\{\boldsymbol{\mu}_\pi \mid \pi \in \Pi_0\right\}$. Indeed, for every $\pi \in \Pi_0$, we have that $\boldsymbol{\mu}_\pi \in \mathfrak{F}$. Moreover, for every feasible solution $\boldsymbol{\mu} \in \mathfrak{F}$, we can obtain a stationary Markov policy $\pi_{\boldsymbol{\mu}} \in \Pi_0$ by $\pi_{\boldsymbol{\mu}}(a|x) \triangleq \frac{\boldsymbol{\mu}(x,a)}{\sum_{a' \in \mathcal{A}} \boldsymbol{\mu}(x,a')}$. Then, the induced occupancy measure is exactly $\boldsymbol{\mu}$.*

Using the epigraph reformulation and Proposition 1, it follows that $(\mathbf{LfD}_{\pi_{\mathrm{E}}})$ is equivalent to the following primal LP:

$$\begin{aligned} \min_{(\boldsymbol{\mu},\varepsilon)} \quad & \varepsilon \\ \text{s.t.} \quad & \left\langle \boldsymbol{\mu} - \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{c}_i \right\rangle \leq \varepsilon,\ i \in [n_c], \\ & \boldsymbol{\mu} \in \mathfrak{F}. \end{aligned} \tag{$\mathbf{P}_{\pi_{\mathrm{E}}}$}$$

The linear program $(\mathbf{P}_{\pi_{\mathrm{E}}})$ resembles the dual LP for solving forward MDPs (Puterman, 1994). Its optimization variables are occupancy measures and a maximum per-feature cost component.

The constraints that define the set $\mathfrak{F}$ are also known as *Bellman flow constraints* and ensure that $\boldsymbol{\mu}$ is an occupancy measure generated by a stationary Markov policy. In particular, $(\boldsymbol{\mu}_{\pi_{\mathrm{A}}}, \alpha^\star)$ is a primal optimizer. Conversely, by an optimal occupancy measure $\boldsymbol{\mu}_{\mathrm{A}}$, an apprentice policy can be extracted as $\pi_{\boldsymbol{\mu}_{\mathrm{A}}}$.

The main objective of this subsection is to shed light to the dual of $(\mathbf{P}_{\pi_{\mathrm{E}}})$ and interpret the dual optimizers.

It holds that $\langle \boldsymbol{\nu}_0, \mathbf{u} \rangle = \left\langle \mathbf{T}_\gamma \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{u} \right\rangle = \left\langle \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{T}_\gamma^* \mathbf{u} \right\rangle$, where $\mathbf{T}_\gamma^* \colon \mathbf{R}^{|\mathcal{X}|} \to \mathbf{R}^{|\mathcal{X}||\mathcal{A}|}$, given by $\mathbf{T}_\gamma^* \mathbf{u} = \frac{1}{1-\gamma}(\mathbf{B} - \gamma \mathbf{P})\mathbf{u}$, is the adjoint operator of $\mathbf{T}_\gamma$. We then get by standard linear duality that the dual LP is given by

$$\begin{aligned} \max_{(\mathbf{u},\mathbf{w})} \quad & \left\langle \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{T}_\gamma^* \mathbf{u} - \mathbf{c}_{\mathbf{w}} \right\rangle \\ \text{s.t.} \quad & \mathbf{c}_{\mathbf{w}} - \mathbf{T}_\gamma^* \mathbf{u} \geq \mathbf{0}, \\ & \mathbf{u} \in \mathbf{R}^{|\mathcal{X}|}, \quad \mathbf{w} \in \Delta_{[n_c]}. \end{aligned} \tag{$\mathbf{D}_{\pi_{\mathrm{E}}}$}$$

The inequality constraint in $(\mathbf{D}_{\pi_{\mathrm{E}}})$ is a relaxation of the Bellman optimality conditions. Therefore, $(\mathbf{D}_{\pi_{\mathrm{E}}})$ resembles the primal LP for solving forward MDPs but its optimization variables are both value functions and cost weights.

Next, we derive optimality conditions for our LP formulations. We first assume that the expert is optimal for the true cost. In this case, the expert policy π_{E} is an optimal solution to $(\mathbf{LfD}_{\pi_{\mathrm{E}}})$.

Lemma 1. *Assume that π_{E} is optimal for $(\mathbf{RL}_{\mathbf{c}_{\mathrm{true}}})$. Then π_{E} is optimal for $(\mathbf{LfD}_{\pi_{\mathrm{E}}})$ with optimal value $\alpha^\star = 0$. Equivalently, $(\boldsymbol{\mu}_{\pi_{\mathrm{E}}}, 0)$ is optimal for $(\mathbf{P}_{\pi_{\mathrm{E}}})$.*

The following Proposition gives necessary and sufficient conditions for dual optimality. The proof is based on Lemma 1 (if π_{E} is optimal, then $\alpha^\star = 0$) and complementary slackness conditions for the LP approach to forward MDPs.

Proposition 2 (Optimal expert)**.** *Assume that π_{E} is optimal for $(\mathbf{RL}_{\mathbf{c}_{\mathrm{true}}})$. A pair $(\mathbf{u}_{\mathrm{A}}, \mathbf{w}_{\mathrm{A}})$ is optimal for $(\mathbf{D}_{\pi_{\mathrm{E}}})$ if and only if $\mathbf{w}_{\mathrm{A}} \in \Delta_{[n_c]}$, π_{E} is optimal for $(\mathbf{RL}_{\mathbf{c}_{\mathbf{w}_{\mathrm{A}}}})$ and $\mathbf{u}_{\mathrm{A}} = \mathbf{V}^\star_{\mathbf{w}_{\mathrm{A}}}$. In particular, $(\mathbf{V}^\star_{\mathbf{w}_{\mathrm{true}}}, \mathbf{w}_{\mathrm{true}})$ is an optimal solution to $(\mathbf{D}_{\pi_{\mathrm{E}}})$.*

Proposition 2 states that, under the expert optimality assumption, the set of dual opimizers characterizes the set of solutions to the IRL problem, i.e, the set of costs in $\mathcal{C}_{\mathrm{conv}}$ for which the expert is optimal. Indeed, a weight vector $\mathbf{w}_{\mathrm{A}} \in \Delta_{[n_u]}$ is dual optimal if and only if the expert policy π_{E} is optimal for the forward RL problem with cost $\mathbf{c}_{\mathbf{w}_{\mathrm{A}}}$. In this case, $\mathbf{u}_{\mathrm{A}}$ coincides with the corresponding optimal value function[1]. In particular, the true weights $\mathbf{w}_{\mathrm{true}}$ and the true optimal value function $\mathbf{V}^\star_{\mathbf{w}_{\mathrm{true}}}$ are dual optimizers.

Is is worth noting that the linear program $(\mathbf{D}_{\pi_{\mathrm{E}}})$ is different from the LP formulation of IRL proposed in (Ng & Russell, 2000; Komanduru & Honorio, 2019), which is based on dynamic programming principles. In contrast, our characterization of solutions to the inverse problem is based on duality arguments in the same line as (Ahuja & Orlin, 2001; Iyengar & Kang, 2005; Pauwels et al., 2016). Moreover, the dual LP formulation $(\mathbf{D}_{\pi_{\mathrm{E}}})$ is independent of the complexity of the expert policy π_{E}, which can be even history-dependent.

In the general case of a suboptimal expert, we have the following necessary conditions for dual optimality.

Proposition 3 (Suboptimal expert)**.** *If $\boldsymbol{\mu}_{\mathrm{A}}$ is optimal for $(\mathbf{P}_{\pi_{\mathrm{E}}})$ and $(\mathbf{u}_{\mathrm{A}}, \mathbf{w}_{\mathrm{A}})$ is optimal for $(\mathbf{D}_{\pi_{\mathrm{E}}})$, then any apprentice policy π_A is optimal for $(\mathbf{RL}_{\mathbf{c}_{\mathbf{w}_{\mathrm{A}}}})$ and $\mathbf{u}_{\mathrm{A}} = \mathbf{V}^\star_{\mathbf{w}_{\mathrm{A}}}$.*

Proposition 3 states that, in general, the apprentice policy π_{A} is an optimal solution to the forward RL problem with cost $\mathbf{c}_{\mathbf{w}_{\mathrm{A}}}$, where $\mathbf{w}_{\mathrm{A}}$ is an optimal dual variable. In addition, $\mathbf{u}_{\mathrm{A}}$ coincides with the corresponding optimal value function.

A similar result to Proposition 3 has been obtained in (Ho

[1] To be precise, this is the case if $\boldsymbol{\nu}_0 \in \mathbf{R}^{|\mathcal{X}|}_{++}$, otherwise they coincide $\boldsymbol{\nu}_0$-almost surely.

& Ermon, 2016) for the convex-concave formulation of maximum causal entropy IRL, by using the Sion minimax theorem (Sion, 1958).

3.2. Saddle-Point Formulation

A serious limitation of the primal and dual LP formulations in Section 3.1 is that they have an intractable number of constraints, which in addition may be not satisfied when working with function approximators. To mitigate this difficulty, we propose a more tractable unconstrained formulation by using the Lagrangian

$$\mathcal{L}(\boldsymbol{\mu}, \mathbf{u}, \mathbf{w}) = \left\langle \boldsymbol{\mu} - \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{c}_{\mathbf{w}} - \mathbf{T}_{\gamma}^{*}\mathbf{u} \right\rangle .$$

We can then frame the problem as a bilinear saddle-point problem

$$\alpha^{\star} = \min_{\boldsymbol{\mu} \in \Delta_{\mathcal{X}\times\mathcal{A}}} \ \max_{\substack{\mathbf{u} \in \mathcal{U} \\ \mathbf{w} \in \Delta_{[n_c]}}} \ \mathcal{L}(\boldsymbol{\mu}, \mathbf{u}, \mathbf{w}), \tag{1}$$

where $\mathcal{U} \triangleq \{\mathbf{u} \in \mathbf{R}^{|\mathcal{X}|} \mid \|\mathbf{u}\|_{\infty} \leq 1\}$. Note that we have added extra constraints $\boldsymbol{\mu} \in \Delta_{\mathcal{X}\times\mathcal{A}}$ and $\mathbf{u} \in \mathcal{U}$ for the sake of analysis. These constraints do not change the problem optimality, but will considerably accelerate the convergence of the algorithm by considering smaller domains.

Indeed, note that the constraint $\boldsymbol{\mu} \in \Delta_{\mathcal{X}\times\mathcal{A}}$ is redundant since it is satisfied for all primal feasible solutions. Moreover, by Proposition 3, for all dual optimizers $(\mathbf{u}_{\mathrm{A}}, \mathbf{w}_{\mathrm{A}})$ it holds that $\mathbf{u}_{\mathrm{A}} = \mathbf{V}^{\star}_{\mathbf{w}_{\mathrm{A}}}$. Thus, we obtain the bound $\|\mathbf{u}_{\mathrm{A}}\|_{\infty} \leq 1$.

The next Corollary follows from Propositions 2–3.

Corollary 1. *Suppose that the expert is optimal for $\mathbf{c}_{\mathrm{true}}$. Then, $(\boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{V}^{\star}_{\mathbf{c}_{\mathrm{true}}}, \mathbf{w}_{\mathrm{true}})$ is a saddle-point to the minimax problem (1). In the general case, for a saddle-point $(\boldsymbol{\mu}_{\mathrm{A}}, \mathbf{u}_{\mathrm{A}}, \mathbf{w}_{\mathrm{A}})$ it holds that: (i) $\pi_{\boldsymbol{\mu}_{\mathrm{A}}}$ is optimal for $(\mathbf{LfD}_{\pi_{\mathrm{E}}})$, and (ii) $\pi_{\boldsymbol{\mu}_{\mathrm{A}}}$ is optimal for $\mathbf{c}_{\mathbf{w}_{\mathrm{A}}}$ and $\mathbf{u}_{\mathrm{A}} = \mathbf{V}^{\star}_{\mathbf{w}_{\mathrm{A}}}$.*

4. A Linearly-Relaxed Saddle-Point Problem

Optimizing directly over $\boldsymbol{\mu}$ and $\mathbf{u}$ is impractical since their dimensions scale linearly with the size of the state space. We reduce the order of complexity by limiting our search to linear subspaces defined by a small number of features. In particular, we consider the feature matrices $\boldsymbol{\Phi} \in \mathbf{R}^{|\mathcal{X}||\mathcal{A}|\times n_{\mu}}$ and $\boldsymbol{\Psi} \in \mathbf{R}^{|\mathcal{X}|\times n_{u}}$. We then assume that the decision variables are parameterized in the form $\boldsymbol{\mu}_{\boldsymbol{\theta}} = \boldsymbol{\Phi}\boldsymbol{\theta}$ and $\mathbf{u}_{\boldsymbol{\lambda}} = \boldsymbol{\Psi}\boldsymbol{\lambda}$, where $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{\lambda} \in \Lambda$ are the parameters to learn. Moreover, Θ and Λ are appropriately chosen parameter sets. We denote $\mathcal{W} \triangleq \Delta_{[n_c]}$ for brevity.

The corresponding *linearly-relaxed* saddle-point formulation is

$$\min_{\boldsymbol{\theta}\in\Theta} \ \max_{\substack{\boldsymbol{\lambda}\in\Lambda \\ \mathbf{w}\in\mathcal{W}}} \mathcal{L}_r(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}), \tag{2}$$

where $\mathcal{L}_r \colon \Theta \times \Lambda \times \mathcal{W} \to \mathbf{R}$ it the *reduced* bilinear Lagrangian $\mathcal{L}_r(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \triangleq \left\langle \boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{c}_{\mathbf{w}} - \mathbf{T}_{\gamma}^{*}\mathbf{u}_{\boldsymbol{\lambda}} \right\rangle$.

We measure the quality of an approximate saddle-point $(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w})$ by its *saddle-point residual (SPR)* defined as

$$\epsilon_{\mathrm{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) = \max_{\substack{\boldsymbol{\lambda}'\in\Lambda \\ \mathbf{w}'\in\mathcal{W}}} \mathcal{L}_r(\boldsymbol{\theta}, \boldsymbol{\lambda}', \mathbf{w}') - \min_{\boldsymbol{\theta}'\in\Theta} \mathcal{L}_r(\boldsymbol{\theta}', \boldsymbol{\lambda}, \mathbf{w}). \tag{3}$$

We assume that every column of $\boldsymbol{\Phi}$ belongs to $\Delta_{\mathcal{X}\times\mathcal{A}}$, and every column of $\boldsymbol{\Psi}$ belongs to $\mathcal{U}$. These conditions ensure that if we choose

$$\Theta \triangleq \Delta_{[n_{\mu}]}, \ \Lambda \triangleq \{\boldsymbol{\lambda} \in \mathbf{R}^{n_u} \mid \|\boldsymbol{\lambda}\|_2 \leq \frac{\beta}{\sqrt{n_u}}\}, \ \beta \geq 2,$$

then $\boldsymbol{\mu}_{\boldsymbol{\theta}} \in \Delta_{\mathcal{X}\times\mathcal{A}}$ and $\|\mathbf{u}_{\boldsymbol{\lambda}}\|_{\infty} \leq \beta$, for all $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{\lambda} \in \Lambda$. Note that we have enlarged the original optimization domain for the $\mathbf{u}$ variable. In addition, by the definition of $\mathcal{C}_{\mathrm{conv}}$, we get $\|\mathbf{c}_{\mathbf{w}}\|_{\infty} \leq 1$, for all $\mathbf{w} \in \mathcal{W}$. These bounds will be used in the sequel in the algorithm design and its theoretical analysis.

4.1. From Approximate Saddle Points to Optimal Policies

A crucial part in our setting is to connect the saddle-point residual $\epsilon_{\mathrm{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w})$ of a feasible solution $(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w})$ for the linearly-relaxed saddle-point problem (2) to the suboptimality gap $\delta_{\mathcal{C}}(\pi_{\boldsymbol{\mu}_{\boldsymbol{\theta}}}, \pi_{\mathrm{E}}) - \delta_{\mathcal{C}}(\pi_{\mathrm{A}}, \pi_{\mathrm{E}})$ of the induced policy $\pi_{\boldsymbol{\mu}_{\boldsymbol{\theta}}}$. If small SPR implies small suboptimality gap, then we can first run any stochastic primal-dual algorithm with sublinear convergence rate to the objective in (2), and then convert the obtained approximate saddle-point to a nearly optimal policy for the original $(\mathbf{LfD}_{\pi_{\mathrm{E}}})$ problem.

For the following results, let π_{A} be an apprentice policy, i.e., π_{A} is optimal for $(\mathbf{LfD}_{\pi_{\mathrm{E}}})$. The proofs can be found in Appendix B. Our first result makes no assumptions on the choice of features.

Proposition 4. *Let $(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \in \Theta \times \Lambda \times \mathcal{W}$ be a feasible solution to (2). Set $\pi_{\boldsymbol{\theta}} = \pi_{\boldsymbol{\mu}_{\boldsymbol{\theta}}}$. It then holds that*

$$\delta_{\mathcal{C}}(\pi_{\boldsymbol{\theta}}, \pi_{\mathrm{E}}) - \delta_{\mathcal{C}}(\pi_{\mathrm{A}}, \pi_{\mathrm{E}}) \leq \epsilon_{\mathrm{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) + \varepsilon_{\mathrm{approx},\boldsymbol{\theta}},$$

where the approximation error $\varepsilon_{\mathrm{approx},\boldsymbol{\theta}}$ is no larger than $\frac{2}{1-\gamma}\big(\beta \min_{\boldsymbol{\theta}'\in\Theta}\|\boldsymbol{\mu}_{\boldsymbol{\theta}'} - \boldsymbol{\mu}_{\pi_{\mathrm{A}}}\|_1 + \min_{\boldsymbol{\lambda}'\in\Lambda}\|\mathbf{u}_{\boldsymbol{\lambda}'} - \mathbf{V}^{\pi_{\boldsymbol{\theta}}}_{\mathbf{c}_{i_{\boldsymbol{\theta}}}}\|_{\infty}\big)$, and $i_{\boldsymbol{\theta}} \triangleq \arg\max_{i\in[n_c]} \big(\rho_{\mathbf{c}_i}(\pi_{\boldsymbol{\theta}}) - \rho_{\mathbf{c}_i}(\pi_{\mathrm{E}})\big)$. In particular, $\rho_{\mathbf{c}_{\mathrm{true}}}(\pi_{\boldsymbol{\theta}}) - \rho_{\mathbf{c}_{\mathrm{true}}}(\pi_{\mathrm{E}}) \leq \epsilon_{\mathrm{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) + \varepsilon_{\mathrm{approx},\boldsymbol{\theta}} + \alpha^{\star}$.

By Proposition 4, if $(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w})$ is an (ε, δ)-optimal saddle point to (2), i.e., $\epsilon_{\mathrm{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \leq \varepsilon$ with probability at least $1-\delta$, then the associated induced policy $\pi_{\boldsymbol{\theta}}$ is $(\varepsilon + \varepsilon_{\mathrm{approx},\boldsymbol{\theta}} + \alpha^{\star})$-optimal for the original $(\mathbf{LfD}_{\pi_{\mathrm{E}}})$ problem with high probability. Recall, that $\alpha^{\star}$ is always zero or negative. The term $\varepsilon_{\mathrm{approx},\boldsymbol{\theta}}$ is a measure of expressiveness of the linear function approximators. The approximation error $\varepsilon_{\text{approx},\boldsymbol{\theta}}$ depends on how well $\{\boldsymbol{\mu}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ approximates the apprentice occupany measure $\boldsymbol{\mu}_{\pi_{\text{A}}}$ and how well $\{\mathbf{u}_{\boldsymbol{\lambda}} : \boldsymbol{\lambda} \in \Lambda\}$ approximates the value function $\mathbf{V}^{\pi_{\boldsymbol{\theta}}}_{\mathbf{c}_{i_{\boldsymbol{\theta}}}}$ of the extracted policy $\pi_{\boldsymbol{\theta}}$ under the cost $\mathbf{c}_{i_{\boldsymbol{\theta}}}$. Note, however, that $\mathbf{V}^{\pi_{\boldsymbol{\theta}}}_{\mathbf{c}_{i_{\boldsymbol{\theta}}}}$ is not fixed before the learning process. Therefore, in order to guarantee a priori low approximation error, we need the optimal occupancy measure to be accurately representable by $\{\boldsymbol{\mu}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$, while $\{\mathbf{u}_{\boldsymbol{\lambda}} : \boldsymbol{\lambda} \in \Lambda\}$ is required to accurately represent the value functions of non-optimal policies as well, under costs in $\{\mathbf{c}_i : i \in [n_c]\}$. Borrowing the terminology from (Shariff & Szepesvári, 2020), we require the occupancy measure features to be *weak*, while the value function features to be *strong*.

An open question is whether we can relax the previously described notion of *good features* for the value function approximation (Shariff & Szepesvári, 2020). We provide the following result in this direction. However, the analysis requires stronger conditions on the choice of features. In particular, we introduce the following assumption first studied in (Bas-Serrano & Neu, 2020).

Assumption 1 (Coherence Assumption)**.** *The columns of $\boldsymbol{\Psi}$ are well-conditioned in the following sense: for every $\boldsymbol{\theta} \in \Theta$ and $\mathbf{u} \in \mathbf{R}^{|\mathcal{X}|}$ with $\|\mathbf{u}\|_\infty \leq 2$, there exists $\boldsymbol{\lambda} \in \Lambda$ such that $\langle \boldsymbol{\nu}_0 - \mathbf{T}_\gamma \boldsymbol{\mu}_{\boldsymbol{\theta}}, \mathbf{u} - \mathbf{u}_{\boldsymbol{\lambda}} \rangle = 0$.*

Proposition 5. *Let Assumption 1 hold. Let $(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \in \Theta \times \Lambda \times \mathcal{W}$ be a feasible solution to (2). Set $\pi_{\boldsymbol{\theta}} = \pi_{\boldsymbol{\mu}_{\boldsymbol{\theta}}}$. It then holds that*

$$\rho_{\mathbf{c}_{\text{true}}}(\pi_{\boldsymbol{\theta}}) - \rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{E}}) \leq 3\epsilon_{\text{sad}}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) + \varepsilon_{\text{approx},w} + \alpha^\star,$$

where the weak approximation error $\varepsilon_{\text{approx},w}$ is no larger than $\frac{2}{1-\gamma}\Big(\beta \min_{\boldsymbol{\theta}' \in \Theta} \|\boldsymbol{\mu}_{\boldsymbol{\theta}'} - \boldsymbol{\mu}_{\pi_{\text{A}}}\|_1 + 2\|\mathbf{V}^\star_{\mathbf{w}_{\text{A}}} - \boldsymbol{\Psi}\boldsymbol{\lambda}^\star\|_\infty\Big)$, where $(\mathbf{V}^\star_{\mathbf{w}_{\text{A}}}, \mathbf{w}_{\text{A}})$ is dual optimal for $(\mathbf{D}_{\pi_{\text{E}}})$ and $\boldsymbol{\lambda}^\star$ is a dual optimizer of (2).

Assumption 1 is satisfied when the rows of $\boldsymbol{\Psi}$ are orthonormal. As it is apparent from the proof of Proposition 5, Assumption 1 controls the *distribution mismatch* between $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\mu}_{\pi_\theta}$. In general $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ is not necessarily an occupancy measure generated by a policy. The term $\|\mathbf{T}_\gamma \boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\nu}_0\|_1$, i.e., the violation degree of the Bellman flow constraints is related to the quality of the extracted policy $\pi_{\boldsymbol{\theta}}$. Assumption 1 ensures that an approximate saddle point with small violation degree of the *aggregated* Bellman flow constraints $\boldsymbol{\Psi}^\intercal(\mathbf{T}_\gamma \boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\nu}_0) = 0$, has also small violation degree for the original Bellman flow constraints and thus produces a *good* policy.

The approximation error term $\|\mathbf{V}^\star_{\mathbf{w}_{\text{A}}} - \boldsymbol{\Psi}\boldsymbol{\lambda}^\star\|_\infty$ can be bounded by approximate dynamic programming techniques (De Farias & Van Roy, 2003; 2004; Shariff & Szepesvári, 2020). For example, under the *realizability assumption* (Chen et al., 2018; Bas-Serrano et al., 2020), we get that $\varepsilon_{\text{approx},w} = 0$. This is formally stated in Lemma **??** in Appendix B.

Moreover, assuming that there exists a set of *core* states whose features span those of other states, the weak value function features ensure a low approximation error $\|\mathbf{V}^\star_{\mathbf{w}_{\text{A}}} - \boldsymbol{\Psi}\boldsymbol{\lambda}^\star\|$ (Lakshminarayanan et al., 2018; Shariff & Szepesvári, 2020).

5. Algorithm and Finite-Sample Analysis

After the analysis of our saddle-point setup, the aim of this section is (i) to provide a computationally efficient stochastic primal-dual algorithm whose iteration and sample complexities do not grow with the size of state and action spaces, and (ii) to obtain explicit probabilistic performance bounds on the quality of the extracted policy with respect to the unknown true cost function.

5.1. Stochastic Primal-Dual LfD Algorithm

Having formulated the linearly-relaxed saddle-point problem (2) with a few variables and constraints, we will now propose an iterative stochastic approximation algorithm for the batch LfD problem. Assuming access to a generative-model oracle and a finite set of expert demonstrations, we propose a stochastic mirror descent primal-dual algorithm which keeps the advantages of simplicity of implementation, low memory requirements, and low computational complexity.

We will consider the slightly modified Lagrangian

$$\begin{aligned}\bar{\mathcal{L}}_r(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \triangleq &\left\langle \boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_{\pi_{\text{E}}}, \mathbf{c}_{\mathbf{w}} - \mathbf{T}^*_\gamma \mathbf{u}_{\boldsymbol{\lambda}} \right\rangle \\ &+ C_{\beta,\gamma} \langle \boldsymbol{\mu}_{\boldsymbol{\theta}}, \mathbf{1} \rangle - 2 \langle \mathbf{w}, \mathbf{1} \rangle ,\end{aligned}$$

where $C_{\beta,\gamma} \triangleq 2\beta/(1-\gamma)$. Note that $\bar{\mathcal{L}}_r$ differs from $\mathcal{L}_r$ up to a constant, since $\langle \boldsymbol{\mu}_{\boldsymbol{\theta}}, \mathbf{1} \rangle = \langle \mathbf{w}, \mathbf{1} \rangle = 1$, for all $\boldsymbol{\theta} \in \Theta$ and $\mathbf{w} \in \mathcal{W}$. Thus, using $\bar{\mathcal{L}}_r$ instead of $\mathcal{L}_r$ does not change the saddle-point residuals. In addition, we have that $\nabla_{\boldsymbol{\theta}} \bar{\mathcal{L}}_r(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \geq \mathbf{0}$ and $\nabla_{\mathbf{w}} \bar{\mathcal{L}}_r(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \leq \mathbf{0}$, for all $\boldsymbol{\theta} \in \Theta$, $\boldsymbol{\lambda} \in \Lambda$ and $\mathbf{w} \in \mathcal{W}$. The same is true for their stochastic gradients. We refer to Sec. 2.8 in (Shalev-Shwartz, 2012) for the theory behind this technique. We will need this property in order to derive finer error bounds by using local norm arguments. For a more detailed discussion, see Section 5.2.

In the forthcoming material we use the following notation. The cost matrix is denoted by $\mathbf{C} \triangleq \begin{bmatrix} \mathbf{c}_1 & \dots & \mathbf{c}_{n_c} \end{bmatrix}$ and the feature matrices by $\boldsymbol{\Phi} \triangleq \begin{bmatrix} \boldsymbol{\phi}_1 & \dots & \boldsymbol{\phi}_{n_\mu} \end{bmatrix}$ and $\boldsymbol{\Psi} \triangleq \begin{bmatrix} \boldsymbol{\psi}_1 & \dots & \boldsymbol{\psi}_{n_u} \end{bmatrix}$. The (x, a)-th row of $\mathbf{C}$ is denoted by $\mathbf{c}_{(x,a)}$ and and the x-th row of $\boldsymbol{\Psi}$ by $\boldsymbol{\psi}_x$.

Note that both the expert policy π_{E} and the transition law P are unknown and they do appear in the reduced Lagrangians $\mathcal{L}_r$ and $\bar{\mathcal{L}}_r$. We will now explain how to tackle this difficulty.

Algorithm 1 Stochastic Primal-Dual LfD

```
Input: cost matrix $\mathbf{C}$, feature matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$
Input: number of iterations $N$, step-size $\eta$, radius $\beta$
Input: expert demonstrations $\mathcal{D}_{\mathrm{E}}^{m,H}$, generative-model
Compute $\boldsymbol{\rho}_{\mathbf{C}}(\widehat{\pi_{\mathrm{E}}})$ using expert demonstrations.
Set $\boldsymbol{\theta}_{1,i} = \frac{1}{n_\mu}, i \in [n_\mu], \mathbf{w}_{1,i} = \frac{1}{n_c}, i \in [n_c], \boldsymbol{\lambda} = \mathbf{0}$
for $n = 1, \ldots N-1$ do
    // $\boldsymbol{\theta}$ gradient estimation
    Sample $i \sim \mathrm{Unif}([\mathrm{n}_\mu]), (x,a) \sim \boldsymbol{\phi}_i, y \sim P(\cdot|x,a)$
    Set
    $g_{n,\boldsymbol{\theta},j} = \begin{cases} \frac{n_\mu\left((1-\gamma)\mathbf{c}_{(x,a)}^\intercal \mathbf{w}_n - (\boldsymbol{\psi}_x - \gamma\boldsymbol{\psi}_y)^\intercal \boldsymbol{\lambda}_n + 2\beta\right)}{1-\gamma}, & j = i \\ 0, & \text{otherwise} \end{cases}$
    // $\boldsymbol{\lambda}$ gradient estimation
    Sample $x' \sim \boldsymbol{\nu}_0, i \sim \boldsymbol{\theta}_n, (x,a) \sim \boldsymbol{\phi}_i, y \sim P(\cdot|x,a)$
    Set $\mathbf{g}_{n,\boldsymbol{\lambda}} = \boldsymbol{\psi}_{x'} - \frac{\boldsymbol{\psi}_x - \gamma\boldsymbol{\psi}_y}{1-\gamma}$
    // $\mathbf{w}$ gradient estimation
    Sample $i \sim \boldsymbol{\theta}_n, (x,a) \sim \boldsymbol{\phi}_i$
    Set $\mathbf{g}_{n,\mathbf{w}} = \mathbf{c}_{(x,a)} - \boldsymbol{\rho}_{\mathbf{C}}(\widehat{\pi_{\mathrm{E}}}) - 2 \cdot \mathbf{1}$
    // Stochastic mirror descent steps
    Update $\theta_{n+1,j} \propto \theta_{n,j} e^{(-\eta g_{n,\boldsymbol{\theta},j})},\ j \in [n_\mu]$
    Update $\boldsymbol{\lambda}_{n+1} = \Pi_\Lambda(\boldsymbol{\lambda}_n + \frac{\eta\beta^2}{n_u}\mathbf{g}_{n,\boldsymbol{\lambda}})$
    Update $w_{n+1,j} \propto w_{n,j} e^{(\eta g_{n,\mathbf{w},j})},\ j \in [n_c]$
end for
Set $\hat{\boldsymbol{\theta}}_N = \frac{1}{N}\sum_{n=1}^N \boldsymbol{\theta}_n$
Output: $\hat{\pi}_N = \pi_{\mathbf{\Phi}\hat{\boldsymbol{\theta}}_N}$
```

We define the *feature expectation* vector $\boldsymbol{\rho}_{\mathbf{C}}(\pi_{\mathrm{E}}) \in \mathbf{R}^{n_c}$ of the expert policy π_{E} by $\boldsymbol{\rho}_{\mathbf{C}}(\pi_{\mathrm{E}}) \triangleq (\rho_{\mathbf{c}_1}(\pi_{\mathrm{E}}), \ldots, \rho_{\mathbf{c}_{n_c}}(\pi_{\mathrm{E}}))^\intercal$. Since in practice we do not have access to the whole policy π_{E}, but instead can observe a finite set of i.i.d. sample trajectories $\mathcal{D}_{\pi_{\mathrm{E}}}^{m,H} \triangleq \{(x_0^k, a_0^k, x_1^k, a_1^k, \ldots, x_H^k, a_H^k)\}_{k=1}^m \sim \pi_{\mathrm{E}}$, we consider the empirical feature expectation vector $\boldsymbol{\rho}_{\mathbf{C}}(\widehat{\pi_{\mathrm{E}}})$ by taking sample averages, i.e., for each $i = 1, \ldots, n_c$, $\rho_{\mathbf{c}_i}(\widehat{\pi_{\mathrm{E}}}) \triangleq (1-\gamma)\frac{1}{m}\sum_{t=0}^H \sum_{j=1}^m \gamma^t c_i(x_t^j, a_t^j)$. So the final empirical Lagrangian that our algorithm optimizes is given by

$$\begin{aligned}\widehat{\mathcal{L}}_r(\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \triangleq& \left\langle \boldsymbol{\mu}_{\boldsymbol{\theta}}, \mathbf{c}_{\mathbf{w}} - \mathbf{T}_\gamma^* \mathbf{u}_{\boldsymbol{\lambda}} + C_{\beta,\gamma} \cdot \mathbf{1} \right\rangle \\ &+ \langle \boldsymbol{\nu}_0, \mathbf{u}_{\boldsymbol{\lambda}} \rangle - \langle \mathbf{w}, \boldsymbol{\rho}_{\mathbf{C}}(\widehat{\pi_{\mathrm{E}}}) + 2 \cdot \mathbf{1} \rangle .\end{aligned}$$

Our optimization variable is $\mathbf{z} = (\boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{w}) \in \mathbf{Z}$, where $\mathbf{Z} \triangleq \Theta \times \Lambda \times \mathcal{W}$ is the decision space. The monotone operator $\mathbf{G}(\mathbf{z}) \triangleq \left[\nabla_{\boldsymbol{\theta}} \widehat{\mathcal{L}}_r(\mathbf{z})^\intercal \quad -\nabla_{\boldsymbol{\lambda}} \widehat{\mathcal{L}}_r(\mathbf{z})^\intercal \quad -\nabla_{\mathbf{w}} \widehat{\mathcal{L}}_r(\mathbf{z})^\intercal\right]^\intercal$ is given by

$$\mathbf{G}(\mathbf{z}) = \begin{pmatrix} \mathbf{\Phi}^\intercal \left(\mathbf{c}_{\mathbf{w}} - \mathbf{T}_\gamma^* \mathbf{u}_{\boldsymbol{\lambda}} + C_{\beta,\gamma} \cdot \mathbf{1}\right) \\ \mathbf{\Psi}^\intercal (\mathbf{T}_\gamma \boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\nu}_0) \\ \boldsymbol{\rho}_{\mathbf{C}}(\widehat{\pi_{\mathrm{E}}}) - \mathbf{C}^\intercal \boldsymbol{\mu}_{\boldsymbol{\theta}} + 2 \cdot \mathbf{1} \end{pmatrix}.$$

Note that when $\mathbf{u}_{\boldsymbol{\lambda}}$ is the value function of a policy, then the term $\mathbf{c}_{\mathbf{w}} - \mathbf{T}_\gamma^* \mathbf{u}_{\boldsymbol{\lambda}}$ is the corresponding *advantage function*. Moreover, in general $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ is not necessarily an occupancy generated by a policy. The term $\|\mathbf{\Psi}^\intercal (\mathbf{T}_\gamma \boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\nu}_0)\|_1$ is the violation degree of the aggregated Bellman flow constraints.

Although the dynamics are unknown, by having at our disposal the generative-model oracle we can compute cheap unbiased gradient estimates. Even in the case of known dynamics, we have expensive matrix computations and one can accelerate learning by randomization. The rationale behind the sampling procedure and the computation of the gradient estimates is similar to (Chen et al., 2018). Indeed, note that we can equivalently consider a stochastic saddle-point formulation, by writing the empirical Lagrangian $\widehat{\mathcal{L}}_r$ in the following sample-friendly form

$$\begin{aligned}&\widehat{\mathcal{L}}_r(\mathbf{z}) \\ &= \sum_{j=1}^{n_\mu} \theta_j \ \mathbf{E}_{\substack{(x,a)\sim\boldsymbol{\phi}_j \\ y\sim P(\cdot|x,a)}} \left[\frac{(1-\gamma)\mathbf{c}_{(x,a)}^\intercal \mathbf{w} - (\boldsymbol{\psi}_x - \gamma\boldsymbol{\psi}_y)^\intercal \boldsymbol{\lambda} + 2\beta}{1-\gamma}\right] \\ &\quad + \mathbf{E}_{x\sim\boldsymbol{\nu}_0}\left[\boldsymbol{\psi}_x^\intercal \boldsymbol{\lambda}\right] - \langle \mathbf{w}, \boldsymbol{\rho}_{\mathbf{C}}(\widehat{\pi_{\mathrm{E}}}) + 2 \cdot \mathbf{1} \rangle .\end{aligned}$$

In particular, in round n we make three independent calls of the generative oracle and compute the unbiased stochastic gradient $\mathbf{g}_n = \left[\mathbf{g}_{n,\boldsymbol{\theta}}^\intercal \quad -\mathbf{g}_{n,\boldsymbol{\lambda}}^\intercal \quad -\mathbf{g}_{n,\mathbf{w}}^\intercal\right]^\intercal$ of $\mathbf{G}(\mathbf{z}_n)$, as described in Algorithm 1. We defer the details to Appendix C.

Note that each iteration costs $\mathcal{O}(1)$ sample generation. Therefore the iteration and the sample complexity of the algorithm coincide. Moreover, the computation of gradient with respect to $\boldsymbol{\lambda}$ and $\mathbf{w}$ is $\mathcal{O}(n_u)$ and $\mathcal{O}(n_c)$, respectively. On the other hand the computation of gradient with respect to μ is $\mathcal{O}(1)$ since only one coordinate is updated per iteration.

The update rule of primal-dual mirror descent is given by

$$\mathbf{z}_{n+1} = \arg\min_{\mathbf{z}\in\mathbf{Z}} \left(\langle \mathbf{g}_n, \mathbf{z} \rangle + \tfrac{1}{\eta} B_R(\mathbf{z}||\mathbf{z}_n)\right),$$

where R is the following distance-generating function

$$R(\mathbf{z}) = \sum_{i=1}^{n_\mu} \theta_i \log \theta_i + \frac{n_u}{2\beta^2}\|\boldsymbol{\lambda}\|_2^2 + \sum_{i=1}^{n_c} w_i \log w_i .$$

The choice of Shannon entropy for the variables $\boldsymbol{\mu}$ and $\mathbf{w}$ which live in the probability simplex mitigates the effects of dimension. Moreover, the factor $\frac{n_u}{2\beta^2}$ is chosen to make the size of Bregman divergence dimension-free (Cheng et al., 2020).

The analytical form of the updates can be seen in Algorithm 1. Note once more that the offsets considered in the modified Lagrangian have no effect since the gradients are exp-transformed and the resulting distribution normalized. We denote by $\widehat{\mathbf{z}}_N \triangleq \frac{1}{N}\sum_{n=1}^N \mathbf{z}_n$ the iterate average after N iterations. Then, the output of the algorithm is the policy $\widehat{\pi}_N \triangleq \pi_{\boldsymbol{\mu}_{\widehat{\boldsymbol{\theta}}_N}}$.

5.2. Sample Complexity

The following theorem gives the sample complexity of Algorithm 1. The proof can be found in Appendix D and is based on high-confidence regret bounds of mirror descent with local norm arguments (Carmon et al., 2019; Jin & Sidford, 2020).

In particular, by combining the mirror descent bounds with Propositions 4–5 we get the following theorem.

Theorem 1. *Let* $\widehat{\pi}_N$ *be the output of running Algorithm 1 for* $N = \max\left\{\mathcal{O}\left(\frac{\beta^2 n_\mu \log\left(\frac{1}{\delta}\right)}{(1-\gamma)^2\varepsilon^2}\right), \mathcal{O}\left(\frac{\beta\sqrt{n_\mu^3 \log\left(\frac{1}{\delta}\right)}}{(1-\gamma)\varepsilon}\right)\right\}$ *iterations, with* $m = \frac{8\log(\frac{4n_c}{\delta})}{\varepsilon^2}$ *expert trajectories of length* $H = \frac{1}{1-\gamma}\log(\frac{2}{\varepsilon})$*, and learning rate* $\eta = \frac{1-\gamma}{\beta\sqrt{Nn_\mu}}$ *. Then, with probability* $1-\delta$ *it holds that* $\rho_{\mathbf{c}_{\text{true}}}(\widehat{\pi}_N) - \rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{E}}) \leq \varepsilon + \varepsilon_{\text{approx},\widehat{\boldsymbol{\theta}}_N} + \alpha^\star$*. If in addition Assumption 1 is satisfied, then with probability* $1-\delta$*, it holds that* $\rho_{\mathbf{c}_{\text{true}}}(\widehat{\pi}_N) - \rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{E}}) \leq \varepsilon + \varepsilon_{\text{approx},w} + \alpha^\star$.

Note that we do not have full access to the true dynamics and the expert policy but instead we can query the generative oracle and observe a batch finite set of truncated expert demonstrations. In our stochastic algorithm this is depicted to the fact that we replace the expert feature expectation vector $\boldsymbol{\rho}_{\mathbf{C}}(\pi_{\text{E}})$ by its empirical counterpart $\boldsymbol{\rho}_{\mathbf{C}}(\widehat{\pi_{\text{E}}})$, and we do not consider the full gradient vector but instead we use unbiased gradient estimates. The estimation error is quantified in terms of concentration inequalities. Indeed for the sample average of the expert feature expectation vector we use a variant of the Hoeffding's inequality (Syed & Schapire, 2007), and for the error due to the stochastic gradient we choose appropriate martingale concentration inequalities (McDiarmid, 1998).

By using the modified Lagrangian $\bar{\mathcal{L}}_r$ and making the unbiased gradient estimates of $\boldsymbol{\theta} \in \Delta_{[n_\mu]}$ and $\mathbf{w} \in \Delta_{[n_c]}$ nonnegative/nonpositive, our final bounds have a better dimension dependency. This trick, also used in (Jin & Sidford, 2020; Cheng et al., 2020), allows us to attain more refined convergence guarantees by exploiting the low variance bounds under the corresponding local norms (as opposed to the $\|\cdot\|_\infty$-norm). We refer to Sec. 2.8 in (Shalev-Shwartz, 2012) for the theory behind this technique. In our case, by replacing $\|\mathbf{g}_{n,\boldsymbol{\theta}}\|_\infty^2$ with the local norm $\|\mathbf{g}_{n,\boldsymbol{\theta}}\|_{\boldsymbol{\theta}_n}^2 \triangleq \sum_{i=1}^{n_\mu} \theta_{n,i} g_{n,\boldsymbol{\theta},i}^2$, we improve the regret bounds by a n_μ-factor.

The parameter β acts as a regularization in learning. If it is too small, the projection residuals $\min_{\boldsymbol{\lambda}\in\Lambda}\|\mathbf{u}_{\boldsymbol{\lambda}} - \mathbf{V}_{\mathbf{c}_{i_{\boldsymbol{\theta}}}}^{\pi_{\boldsymbol{\theta}}}\|_\infty$ and $\|\mathbf{V}_{\mathbf{w}_{\text{A}}}^\star - \boldsymbol{\Psi}\boldsymbol{\lambda}^\star\|_\infty$ in the approximation error terms $\varepsilon_{\text{approx},\boldsymbol{\theta}}$ and $\varepsilon_{\text{approx},w}$, respectively, are bigger. If it is too large, the learning becomes slower.

In Appendix E we provide preliminary empirical results on a simple tabular MDP in order to illustrate our formulations and theoretical results.

5.3. A No-Regret Online Learning View

In Appendix F, we argue that solving the online learning problem with $\mathbf{Z} = \Theta \times \Lambda \times \mathcal{W}$ and per-round loss function

$$\ell_n(\mathbf{z}) \triangleq \mathcal{L}_r(\boldsymbol{\theta}, \boldsymbol{\lambda}_n, \mathbf{w}_n) - \mathcal{L}_r(\boldsymbol{\theta}_n, \boldsymbol{\lambda}, \mathbf{w})$$

is equivalent to solving the linearly-relaxed saddle-point problem (2).

It is worth noting that this online learning approach differs from the one in (Ross et al., 2011) where interaction with the expert is required. It is also different from the game-theoretic approach in (Syed & Schapire, 2007) where the forward RL problem has to be solved repeatedly.

5.4. Linear Function Approximators

We are motivated to work with occupancy measures (OMs) $\boldsymbol{\mu}$ instead of policies π because of linearity and flexibility of $(\mathbf{P}_{\pi_{\text{E}}})$ and $(\mathbf{D}_{\pi_{\text{E}}})$. We highlight that Propositions 4–5 hold even for nonlinear parameterizations $\{\boldsymbol{\mu}_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta\}$ and $\{\mathbf{u}_{\boldsymbol{\lambda}} \mid \boldsymbol{\lambda} \in \Lambda\}$, as long as $\boldsymbol{\mu}_{\boldsymbol{\theta}} \in \mathcal{M}$ and $\|\mathbf{u}_{\boldsymbol{\lambda}}\|_\infty \leq \beta$ hold. Thus, the LfD problem is decoupled in two parts: minimization of SPR and function approximation. For the first part, we employ modern large-scale stochastic optimization methods while the second part can be quantified independently of the learning process. We choose LFA to preserve the convexity-concavity of the Lagrangian and design a scalable algorithm with theoretical guarantees (sublinear convergence rate), though at a cost of a potential approximation bias. Indeed, LFA require a careful choice of features. One can use prior knowledge of the MDP to choose appropriate basis functions for the value functions, e.g., in our case the value function basis $\{\boldsymbol{\psi}_i = \mathbf{V}_{\mathbf{c}_i}^\star \mid i \in [n_c]\}$) satisfies the realizability assumption. On the other hand the choice of OM features is trickier since only a restrictive class of policies can be represented (Banijamali et al., 2019). One can use OMs of policies extracted from "heuristic" methods or policies provided from multiple "cheaper" suboptimal experts (where one can sample a lot of rollouts) and use Algorithm 1 to improve upon them. We hope that our techniques will be useful for future algorithm designers and will lay foundations for more exhaustive research in this direction. In Appendix G we point out a few interesting directions.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement OCAL, No. 787845.

References

Abbasi-Yadkori, Y., Bartlett, P. L., and Malek, A. Linear programming for large-scale Markov decision problems. In *International Conference on Machine Learning (ICML)*, 2014.

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2004.

Abbeel, P., Dolgov, D., Ng, A. Y., and Thrun, S. Apprenticeship learning for motion planning with application to parking lot navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

Ahuja, R. K. and Orlin, J. B. Inverse optimization. *Operations Research*, 49(5):771–783, 2001.

Bagnell, J. A. D. An invitation to imitation. Technical Report CMU-RI-TR-15-08, Carnegie Mellon University, Pittsburgh, PA, March 2015.

Banijamali, E., Abbasi-Yadkori, Y., Ghavamzadeh, M., and Vlassis, N. Optimizing over a restricted policy class in MDPs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Bas-Serrano, J. and Neu, G. Faster saddle-point optimization for solving large-scale Markov decision processes. In *Conference on Learning for Dynamics and Control (L4DC)*, 2020.

Bas-Serrano, J., Curi, S., Krause, A., and Neu, G. Logistic Q-learning. *arXiv:2010.11151*, 2020.

Bertsekas, D. P. and Tsitsiklis, J. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Borkar, V. S. A convex analytic approach to Markov decision processes. *Probability Theory and Related Fields*, 78(4):583–602, 1988.

Brown, D., Niekum, S., and Petrik, M. Bayesian robust optimization for imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Brown, D. S., Coleman, R., Srinivasan, R., and Niekum, S. Safe imitation learning via fast Bayesian reward inference from preferences. In *International Conference on Machine Learning (ICML)*, 2020b.

Cai, Q., Hong, M., Chen, Y., and Wang, Z. On the global convergence of imitation learning: a case for linear quadratic regulator. *arXiv:1901.03674*, 2019.

Carmon, Y., Jin, Y., Sidford, A., and Tian, K. Variance reduction for matrix games. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Chen, M., Wang, Y., Liu, T., Yang, Z., Li, X., Wang, Z., and Zhao, T. On computation and generalization of generative adversarial imitation learning. *International Conference on Learning Representations (ICLR)*, 2020a.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020b.

Chen, Y., Li, L., and Wang, M. Scalable bilinear π learning using state and action features. In *International Conference on Machine Learning (ICML)*, 2018.

Chen, Y. F., Everett, M., Liu, M., and How, J. P. Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

Cheng, C.-A., des Combes, R. T., Boots, B., and Gordon, G. A reduction from reinforcement learning to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

De Farias, D. P. and Van Roy, B. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

De Farias, D. P. and Van Roy, B. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

Dufour, F. and Prieto-Rumeau, T. Finite linear programming approximations of constrained discounted Markov decision processes. *SIAM Journal on Control and Optimization*, 51(2):1298–1324, 2013.

Fazel, M., Ge, R., Kakade, S., and Mesbahi, M. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning (ICML)*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Hernández-Lerma, O. and Lasserre, J. B. *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. Springer-Verlag New York, 1996.

Hernández-Lerma, O., González-Hernández, J., and López-Martínez, R. Constrained average cost Markov control processes in Borel spaces. *SIAM Journal on Control and Optimization*, 42(2):442–468, 2003.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Ho, J., Gupta, J. K., and Ermon, S. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning (ICML)*, 2016.

Iyengar, G. and Kang, W. Inverse conic programming with applications. *Operations Research Letters*, 33(3):319–330, 2005.

Jin, Y. and Sidford, A. Efficiently solving MDPs with stochastic mirror descent. In *International Conference on Machine Learning (ICML)*, 2020.

Komanduru, A. and Honorio, J. On the correctness and sample complexity of inverse reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Lakshminarayanan, C., Bhatnagar, S., and Szepesvári, C. A linearly relaxed approximate linear program for Markov decision processes. *IEEE Transactions on Automatic Control*, 63(4):1185–1191, 2018.

Lee, D. and He, N. Stochastic primal-dual Q-learning algorithm for discounted MDPs. In *American Control Conference (ACC)*, 2019.

Levine, S., Popović, Z., and Koltun, V. Feature construction for inverse reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

Levine, S., Popović, Z., and Koltun, V. Nonlinear inverse reinforcement learning with Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

Manne, A. Linear programming and sequential decisions. *Management Science*, 6(3):259–267, 1960.

McDiarmid, C. *Concentration*, pp. 195–248. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, February 2015.

Mohajerin Esfahani, P., Sutter, T., Kuhn, D., and Lygeros, J. From infinite to finite programs: explicit error bounds with applications to approximate dynamic programming. *SIAM Journal on Optimization*, 28(3):1968–1998, 2018.

Nachum, O. and Dai, B. Reinforcement learning via Fenchel-Rockafellar duality. *arXiv:2001.01866*, 2020.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Neu, G. and Szepesvári, C. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

Ng, A. Y. and Russell, S. J. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2000.

Pauwels, E., Henrion, D., and Lasserre, J.-B. Linear conic optimization for inverse optimal control. *SIAM Journal on Control and Optimization*, 54(3):1798–1825, 2016.

Pomerleau, D. A. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994.

Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. Maximum margin planning. In *International Conference on Machine Learning (ICML)*, 2006.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Russell, S. Learning agents for uncertain environments (extended abstract). In *Annual Conference on Computational Learning Theory (COLT)*, 1998.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, M. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, 2015.

Shafieepoorfard, E., Raginsky, M., and Meyn, S. P. Rational inattention in controlled Markov processes. In *American Control Conference (ACC)*, 2013.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

Shariff, R. and Szepesvári, C. Efficient planning in large MDPs with weak linear function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Sion, M. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

Sutton, R. S. and Barto, A. G. *Introduction to Reinforcement Learning*. MIT Press, 1st edition, 1998.

Syed, U. and Schapire, R. E. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007.

Syed, U., Bowling, M., and Schapire, R. Apprenticeship learning using linear programming. In *International Conference on Machine Learning (ICML)*, 2008.

Szörényi, B., Kedenburg, G., and Munos, R. Optimistic planning in Markov decision processes using a generative model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Taleghan, M. A., Dietterich, T. G., Crowley, M., Hall, K., and Albers, H. J. PAC optimal MDP planning with application to invasive species management. *Journal of Machine Learning Research*, 16(117):3877–3903, 2015.

Tesauro, G. and Kephart, J. O. Pricing in agent economies using multi-agent Q-learning. *Autonomous Agents and Multi-Agent Systems*, 5(3):289–304, 2002.

Vamvoudakis, K. G. and Kokolakis, N.-M. T. Synchronous reinforcement learning-based control for cognitive autonomy. *Foundations and Trends in Systems and Control*, 8 (1-–2):1–175, 2020.

Wang, L., Cai, Q., Yang, Z., and Wang, Z. Neural policy gradient methods: Global optimality and rates of convergence. In *International Conference on Learning Representations*, 2020.

Wang, M. Randomized linear programming solves the Markov decision problem in nearly linear (sometimes sublinear) time. *Mathematics of Operations Research*, 45 (2):517–546, 2020.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

Zhang, Y., Cai, Q., Yang, Z., and Wang, Z. Generative adversarial imitation learning with neural network parameterization: global optimality and convergence rate. In *International Conference on Machine Learning (ICML)*, 2020.

Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *National Conference on Artificial Intelligence (AAAI)*, 2008.