Inverse Reinforcement Learning with Explicit Policy Estimates

Navyata Sanghvi[†1], Shinnosuke Usami[†1,2], Mohit Sharma[†1], Joachim Groeger[*1], Kris Kitani[1]

[1]Carnegie Mellon University
[2]Sony Corporation
navyatasanghvi@cmu.edu, shinnosuke.usami@sony.com,
mohitsharma@cmu.edu, jrg@joachimgroeger.com, kkitani@cs.cmu.edu

Abstract

Various methods for solving the inverse reinforcement learning (IRL) problem have been developed independently in machine learning and economics. In particular, the method of Maximum Causal Entropy IRL is based on the perspective of entropy maximization, while related advances in the field of economics instead assume the existence of unobserved action shocks to explain expert behavior (Nested Fixed Point Algorithm, Conditional Choice Probability method, Nested Pseudo-Likelihood Algorithm). In this work, we make previously unknown connections between these related methods from both fields. We achieve this by showing that they all belong to a class of optimization problems, characterized by a common form of the objective, the associated policy and the objective gradient. We demonstrate key computational and algorithmic differences which arise between the methods due to an approximation of the optimal soft value function, and describe how this leads to more efficient algorithms. Using insights which emerge from our study of this class of optimization problems, we identify various problem scenarios and investigate each method's suitability for these problems.

1 Introduction

Inverse Reinforcement Learning (IRL) – the problem of inferring the reward function from observed behavior – has been studied independently both in machine learning (ML) (Abbeel and Ng 2004; Ratliff, Bagnell, and Zinkevich 2006; Boularias, Kober, and Peters 2011) and economics (Miller 1984; Pakes 1986; Rust 1987; Wolpin 1984). One of the most popular IRL approaches in the field of machine learning is Maximum Causal Entropy IRL (Ziebart 2010). While this approach is based on the perspective of entropy maximization, independent advances in the field of economics instead assume the existence of unobserved action shocks to explain expert behavior (Rust 1988). Both these approaches optimize likelihood-based objectives, and are computationally expensive. To ease the computational burden, related methods in economics make additional assumptions to infer rewards (Hotz and Miller 1993; Aguirregabiria and Mira 2002). While the perspectives these four methods take suggest a relationship between them, to the best of our knowledge, we are the first to make explicit connections between them. The development of a common theoretical framework results in a unified perspective of related methods from both fields. This enables us to compare the suitability of methods for various problem scenarios, based on their underlying assumptions and the resultant quality of solutions.

To establish these connections, we first develop a common optimization problem form, and describe the associated objective, policy and gradient forms. We then show how each method solves a particular instance of this common form. Based on this common form, we show how estimating the optimal soft value function is a key characteristic which differentiates the methods. This difference results in two algorithmic perspectives, which we call optimization- and approximation-based methods. We investigate insights derived from our study of the common optimization problem towards determining the suitability of the methods for various problem settings.

Our contributions include: (1) developing a unified perspective of methods proposed by Ziebart (2010); Rust (1987); Hotz and Miller (1993); Aguirregabiria and Mira (2002) as particular instances of **a class of IRL optimization problems** that share a common objective and policy form (Section 4); (2) explicitly demonstrating **algorithmic and computational differences** between methods, which arise from a difference in soft value function estimation (Section 5); (3) investigating the **suitability of methods** for various types of problems, using insights which emerge from a study of our unified perspective (Section 6).

2 Related Work

Many formulations of the IRL problem have been proposed previously, including maximum margin formulations (Abbeel and Ng 2004; Ratliff, Bagnell, and Zinkevich 2006) and probabilistic formulations (Ziebart 2010). These methods are computationally expensive as they require repeatedly solving the underlying MDP. We look at some methods which reduce this computational burden.

One set of approaches avoids the repeated computation by casting the estimation problem as a supervised classification or regression problem (Klein et al. 2012, 2013). Structured Classification IRL (SC-IRL) (Klein et al. 2012) assumes a lin-

[†]Equal contribution.[*]Work done while at CMU.

early parameterized reward and uses expert policy estimates to reduce the IRL problem to a multi-class classification problem. However, SC-IRL is restricted by its assumption of a *linearly* parameterized reward function.

Another work that avoids solving the MDP repeatedly is Relative Entropy IRL (RelEnt-IRL) (Boularias, Kober, and Peters 2011). RelEnt-IRL uses a baseline policy for value function approximation. However, such baseline policies are in general not known (Ziebart 2010), and thus RelEnt-IRL cannot be applied in such scenarios.

One method that avoids solving the MDP problem focuses on linearly solvable MDPs (Todorov 2007). (Dvijotham and Todorov 2010) present an efficient IRL algorithm, which they call OptV, for such linearly solvable MDPs. However, this class of MDPs *assumes* that the Bellman equation can be transformed into a linear equation. Also, OptV uses a value-function parameterization instead of a reward-function parameterization, it can have difficulties with generalization when it is not possible to transfer value-function parameters to new environments (Ziebart 2010; Levine and Koltun 2012).

Recent work that avoids solving the MDP repeatedly is the CCP-IRL approach (Sharma, Kitani, and Groeger 2017), which observes a connection between Maximum Causal Entropy IRL (MCE-IRL) and Dynamic Discrete Choice models, and uses it to introduce a conditional choice probability (CCP)-based IRL algorithm. On the other hand, our work establishes formal connections between MCE-IRL and a suite of approximation-based methods, of which the CCP method is but one instance. Unlike recent work, we perform a comprehensive theoretical and empirical analysis of each algorithm in the context of trade-offs between the correctness of the inferred solution and its computational burden.

3 Preliminaries

In this section, we first introduce the forward decision problem formulation used in economics literature. We then familiarize the reader with the inverse problem of interest, *i.e.,* inferring the reward function, and the associated notation.

The Dynamic Discrete Choice (DDC) model is a discrete Markov Decision Process (MDP) with *action shocks*. A DDC is represented by the tuple $(\mathcal{S}, \mathcal{A}, T, r, \gamma, \mathcal{E}, F)$. $\mathcal{S}$ and $\mathcal{A}$ are a countable sets of states and actions respectively. $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function. $r : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function. γ is a discount factor. Distinct from the MDP, each action has an associated "shock" variable $\epsilon \in \mathcal{E}$, which is unobserved and drawn from a distribution F over $\mathcal{E}$. The vector of shock variables, one for each action, is denoted $\boldsymbol{\epsilon}$. The unobserved shocks ϵ_a account for agents that sometimes take seemingly sub-optimal actions (McFadden et al. 1973). For the rest of this paper, we will use the shorthand p' for transition dynamics $T(s'|s, a)$ and softmax$f(a) = \exp f(a) / \sum_a \exp f(a)$.

The Forward Decision Problem: Similar to reinforcement learning, the DDC forward decision problem in state $(s, \boldsymbol{\epsilon})$ is to select the action a that maximizes future aggregated utility: $\mathbb{E}\left[\sum_t \gamma^t \left(r(s_t, a_t, \boldsymbol{\theta}) + \epsilon_{a_t}\right) \mid (s, \boldsymbol{\epsilon})\right]$ where the state-action reward function is parametrized by $\boldsymbol{\theta}$. (Rust 1988) describes the following Bellman optimality equation for the optimal value function $V^*_{\boldsymbol{\theta}}(s, \boldsymbol{\epsilon})$:

$$
\begin{aligned}
&V^*_{\boldsymbol{\theta}}(s, \boldsymbol{\epsilon}) \\
&= \max_{a \in \mathcal{A}} \left\{ r(s, a, \boldsymbol{\theta}) + \epsilon_a + \gamma \mathbb{E}_{s' \sim p', \boldsymbol{\epsilon}'} \left[V^*_{\boldsymbol{\theta}}(s', \boldsymbol{\epsilon}') \right] \right\}. \quad (1)
\end{aligned}
$$

The optimal choice-specific value is defined as $Q^*_{\boldsymbol{\theta}}(s, a) \triangleq r(s, a, \boldsymbol{\theta}) + \mathbb{E}_{s' \sim p', \boldsymbol{\epsilon}'} \left[V^*_{\boldsymbol{\theta}}(s', \boldsymbol{\epsilon}') \right]$. Then:

$$
V^*_{\boldsymbol{\theta}}(s, \boldsymbol{\epsilon}) = \max\nolimits_{a \in \mathcal{A}} \left\{ \, Q^*_{\boldsymbol{\theta}}(s, a) + \epsilon_a \, \right\}. \quad (2)
$$

Solution: The choice-specific value $Q^*_{\boldsymbol{\theta}}(s, a)$ is the fixed point of the contraction mapping $\Lambda_{\boldsymbol{\theta}}$ (Rust 1988):

$$
\begin{aligned}
&\Lambda_{\boldsymbol{\theta}}(Q)(s, a) \\
&= r(s, a, \boldsymbol{\theta}) + \gamma \mathbb{E}_{s' \sim p', \boldsymbol{\epsilon}'} \left[\max_{a' \in \mathcal{A}} \left\{ Q(s', a') + \epsilon_{a'} \right\} \right]. \quad (3)
\end{aligned}
$$

We denote the indicator function as $\mathbf{I}\{\}$. From (2), the optimal policy at $(s, \boldsymbol{\epsilon})$ is given by:

$$
\pi(a|s, \boldsymbol{\epsilon}) = \mathbf{I}\left\{ a = \arg\max\nolimits_{a' \in \mathcal{A}} \left\{ Q^*_{\boldsymbol{\theta}}(s, a') + \epsilon_{a'} \right\} \right\}. \quad (4)
$$

Therefore, $\pi_{\boldsymbol{\theta}}(a|s) = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\pi(a|s, \boldsymbol{\epsilon})\right]$ is the optimal choice conditional on state alone. $\pi_{\boldsymbol{\theta}}$ is called the *conditional choice probability* (CCP). Notice, $\pi_{\boldsymbol{\theta}}$ has the same form as a *policy* in an MDP.

The Inverse Decision Problem: The inverse problem, *i.e.*, IRL in machine learning (ML) and structural parameter estimation in econometrics, is to estimate the parameters $\boldsymbol{\theta}$ of a state-action reward function $r(s, a, \boldsymbol{\theta})$ from expert demonstrations. The expert follows an unknown policy $\pi_E(a|s)$. A state-action trajectory is denoted: $(\mathbf{s}, \mathbf{a}) = (s_0, a_0, ..., s_T)$. The expert's distribution over trajectories is given by $P_E(\mathbf{s}, \mathbf{a})$. Considering a Markovian environment, the product oNf transition dynamics terms is denoted $P(\mathbf{s}^T || \mathbf{a}^{T-1}) = \prod_{\tau=0}^{T} P(s_\tau | s_{\tau-1}, a_{\tau-1})$, and the product of expert policy terms is denoted: $\pi_E(\mathbf{a}^T || \mathbf{s}^T) = \prod_{\tau=0}^{T} \pi_E(a_\tau | s_\tau)$. The expert distribution is $P_E(\mathbf{s}, \mathbf{a}) = \pi_E(\mathbf{a}^T || \mathbf{s}^T) \; P(\mathbf{s}^T || \mathbf{a}^{T-1})$. Similarly, for a policy $\pi_{\boldsymbol{\theta}}$ dependent on reward parameters $\boldsymbol{\theta}$, the distribution over trajectories generated using $\pi_{\boldsymbol{\theta}}$ is given by $P_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a}) = \pi_{\boldsymbol{\theta}}(\mathbf{a}^T || \mathbf{s}^T) \; P(\mathbf{s}^T || \mathbf{a}^{T-1})$.

4 A Unified Perspective

In order to compare various methods of reward parameter estimation that have been developed in isolation in the fields of economics and ML, it is important to first study their connections and commonality. To facilitate this, in this section, we develop a unified perspective of the following methods: Maximum Causal Entropy IRL (MCE-IRL) (Ziebart 2010), Nested Fixed Point Algorithm (NFXP) (Rust 1988), Conditional Choice Probability (CCP) (Hotz and Miller 1993), and Nested Pseudo-Likelihood Algorithm (NPL) (Aguirregabiria and Mira 2002).

To achieve this, we first describe a class of optimization problems that share a common form. While the methods we discuss were derived from different perspectives, we show how each method is a specific instance of this class. We characterize this class of optimization problems using a common form of the objective, the associated policy $\pi_{\boldsymbol{\theta}}$ and the objective gradient. In subsequent subsections, we discuss the critical point of difference between the algorithms: *the explicit specification of a policy* $\tilde{\pi}$.

Objective Form: The common objective in terms of $\boldsymbol{\theta}$ is to maximize expected log likelihood $L(\boldsymbol{\theta})$ of trajectories generated using a policy $\pi_{\boldsymbol{\theta}}$, under expert distribution $P_E(\mathbf{s}, \mathbf{a})$:

$$L(\boldsymbol{\theta}) = \mathbb{E}_{P_E(\mathbf{s},\mathbf{a})} \left[\log P_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{a})\right]. \tag{5}$$

Since transition dynamics $P(\mathbf{s}^T || \mathbf{a}^{T-1})$ do not depend on $\boldsymbol{\theta}$, maximizing $L(\boldsymbol{\theta})$ is the same as maximizing $g(\boldsymbol{\theta})$, *i.e.*, the expected log likelihood of $\pi_{\boldsymbol{\theta}}(\mathbf{a}^T || \mathbf{s}^T)$ under $P_E(\mathbf{s}, \mathbf{a})$:

$$\begin{aligned} g(\boldsymbol{\theta}) &= \mathbb{E}_{P_E(\mathbf{s},\mathbf{a})} \left[\log \pi_{\boldsymbol{\theta}}(\mathbf{a}^T || \mathbf{s}^T)\right] \\ &= \mathbb{E}_{P_E(\mathbf{s},\mathbf{a})} \left[\textstyle\sum_\tau \log \pi_{\boldsymbol{\theta}}(a_\tau | s_\tau)\right]. \end{aligned} \tag{6}$$

Policy Form: The policy $\pi_{\boldsymbol{\theta}}$ in objective (6) has a general form defined in terms of the state-action "soft" value function $Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s, a)$ (Haarnoja et al. 2017), under *some* policy $\tilde{\pi}$.

$$\begin{aligned} & Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s, a) \\ & = r(s, a, \boldsymbol{\theta}) + \mathbb{E}_{s', a' \sim \tilde{\pi}} \left[Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s', a') - \log \tilde{\pi}(a'|s')\right], \quad (7) \\ & \pi_{\boldsymbol{\theta}}(a|s) = \text{softmax } Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s, a). \quad (8) \end{aligned}$$

This policy form guarantees that $\pi_{\boldsymbol{\theta}}$ is an *improvement* over $\tilde{\pi}$ in terms of soft value, *i.e.,* $Q^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\theta}}(s, a) \geq Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s, a), \forall (s, a)$ (Haarnoja et al. 2018). In subsequent sections, we *explicitly* define $\tilde{\pi}$ in the context of each method.

Gradient Form: With this policy form (8), the gradient of the objective (6) is given by:

$$\begin{aligned} \frac{\partial g}{\partial \boldsymbol{\theta}} = \sum_\tau \mathbb{E}_{P_E(\mathbf{s}_{1:\tau}, \mathbf{a}_{1:\tau})} & \left[\frac{\partial Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s_\tau, a_\tau)}{\partial \boldsymbol{\theta}} \right. \\ & \left. - \sum_{a'} \pi_{\boldsymbol{\theta}}(a'|s_\tau) \frac{\partial Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s_\tau, a')}{\partial \boldsymbol{\theta}} \right]. \end{aligned} \tag{9}$$

Proof: Sanghvi et al. 2021 (Appendix A.1).

The general forms we detailed above consider no discounting. In case of discounting by factor γ, simple modifications apply to the objective, soft value and gradient (6, 7, 9).

Details: Sanghvi et al. 2021 (Appendix A.2).

We now show how each method is a specific instance of the class of optimization problems characterized by (6-9). Towards this, we explicitly specify $\tilde{\pi}$ in the context of each method. We emphasize that, in order to judge how suitable each method is for any problem, it is important to understand the assumptions involved in these specifications and how these assumptions cause differences between methods.

4.1 Maximum Causal Entropy IRL and Nested Fixed Point Algorithm

MCE-IRL (Ziebart 2010) and NFXP (Rust 1988) originated independently in the ML and economics communities respectively, but they can be shown to be equivalent. NFXP (Rust 1988) solves the DDC forward decision problem for $\pi_{\boldsymbol{\theta}}$, and maximizes its likelihood under observed data. On the other hand, MCE-IRL formulates the estimation of $\boldsymbol{\theta}$ as the dual of maximizing causal entropy subject to feature matching constraints under the observed data.

NFXP: Under the assumption that shock values ϵ_a are *i.i.d* and drawn from a TIEV distribution: $F(\epsilon_a) = e^{-e^{-\epsilon_a}}$, NFXP solves the forward decision problem (Section 3). At the solution, the CCP:

$$\pi_{\boldsymbol{\theta}}(a|s) = \text{softmax } Q^*_{\boldsymbol{\theta}}(s, a), \tag{10}$$

where $Q^*_{\boldsymbol{\theta}}(s, a)$ is the optimal choice-specific value function (3). We can show $Q^*_{\boldsymbol{\theta}}$ is the optimal soft value, and, consequently, $\pi_{\boldsymbol{\theta}}$ is optimal in the soft value sense.

Proof: Sanghvi et al. 2021 (Appendix A.3).

To estimate $\boldsymbol{\theta}$, NFXP maximizes the expected log likelihood of trajectories generated using $\pi_{\boldsymbol{\theta}}$ (10) under the expert distribution. We can show the gradient of this objective is:

$$\mathbb{E}_{P_E(\mathbf{s},\mathbf{a})} \left[\sum_t \frac{\partial r(s_t, a_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] - \mathbb{E}_{P_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})} \left[\sum_t \frac{\partial r(s_t, a_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right]. \tag{11}$$

Proof: Sanghvi et al. 2021 (Appendix A.4).

MCE-IRL: (Ziebart 2010) estimates $\boldsymbol{\theta}$ by following the dual gradient:

$$\mathbb{E}_{P_E(\mathbf{s},\mathbf{a})} \left[\textstyle\sum_t \mathbf{f}(s_t, a_t)\right] - \mathbb{E}_{P_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})} \left[\textstyle\sum_t \mathbf{f}(s_t, a_t)\right], \tag{12}$$

where $\mathbf{f}(s, a)$ is a vector of state-action features, and $\mathbb{E}_{P_E(\mathbf{s},\mathbf{a})} \left[\sum_t \mathbf{f}(s_t, a_t)\right]$ is estimated from expert data. The reward is a linear function of features $r(s, a, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{f}(s, a)$, and the policy $\pi_{\boldsymbol{\theta}}(a|s) = \text{softmax } Q^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\theta}}(s, a)$. This implies $\pi_{\boldsymbol{\theta}}$ is optimal in the soft value sense (Haarnoja et al. 2018).

Connections: From our discussion above we see that, for both NFXP and MCE-IRL, policy $\pi_{\boldsymbol{\theta}}$ is optimal in the soft value sense. Moreover, when the reward is a linear function of features $r(s, a, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{f}(s, a)$, the gradients (11) and (12) are equivalent. Thus, NFXP and MCE-IRL are equivalent.

We now show that both methods are instances of the class of optimization problems characterized by (6-9). From the discussion above, $\pi_{\boldsymbol{\theta}}(a|s) = \text{softmax } Q^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\theta}}(s, a)$. Comparing this with the forms (7, 8), for these methods, $\tilde{\pi} = \pi_{\boldsymbol{\theta}}$. Furthermore, by specifying $\tilde{\pi} = \pi_{\boldsymbol{\theta}}$, we can show that the gradients (11, 12) are equivalent to our objective gradient (9) (*Proof:* Sanghvi et al. 2021 (Appendix A.5)). From this we can conclude NFXP and MCE-IRL are solving objective (6).

Computing $Q^*_{\boldsymbol{\theta}}$: For NFXP and MCE-IRL, every gradient step requires the computation of optimal soft value $Q^*_{\boldsymbol{\theta}}$. The policy $\pi_{\boldsymbol{\theta}}$ (10) is optimal in the soft value sense. $Q^*_{\boldsymbol{\theta}}$ is computed using the following fixed point iteration. This is a computationally expensive dynamic programming problem.

$$Q(s, a) \leftarrow r(s, a, \boldsymbol{\theta}) + \gamma \mathbb{E}_{s' \sim p'} \left[\log \textstyle\sum_{a'} \exp Q(s', a')\right] \tag{13}$$

4.2 Conditional Choice Probability Method

As discussed in Section 4.1, NFXP and MCE-IRL require computing the optimal soft value $Q^*_{\boldsymbol{\theta}}$ (3) at every gradient step, which is computationally expensive. To avoid this, the CCP method (Hotz and Miller 1993) is based on the idea of approximating the optimal soft value $Q^*_{\boldsymbol{\theta}}$. To achieve this, they approximate $Q^*_{\boldsymbol{\theta}} \approx Q^{\hat{\pi}_E}_{\boldsymbol{\theta}}$, where $Q^{\hat{\pi}_E}_{\boldsymbol{\theta}}$ is the soft value under a simple, nonparametric estimate $\hat{\pi}_E$ of the expert's policy π_E. The CCP $\pi_{\boldsymbol{\theta}}$ (10) is then estimated as:

$$\pi_{\boldsymbol{\theta}}(a|s) = \text{softmax } Q^{\hat{\pi}_E}_{\boldsymbol{\theta}}(s, a) \tag{14}$$

In order to estimate parameters $\boldsymbol{\theta}$, the CCP method uses the method of moments estimator (Hotz and Miller 1993; Aguirregabiria and Mira 2010):

$$\mathbb{E}_{P_E(\mathbf{s},\mathbf{a})}\left[\sum_\tau \sum_a \mathbf{F}(s_\tau, a)\left(\mathbf{I}\{a_\tau = a\} - \pi_{\boldsymbol{\theta}}(a|s_\tau)\right)\right] = 0, \quad (15)$$

where $\mathbf{F}(s,a) = \frac{\partial Q_{\boldsymbol{\theta}}^{\hat{\pi}_E}(s,a)}{\partial \boldsymbol{\theta}}$ (Aguirregabiria and Mira 2002). An in-depth discussion of this estimator may be found in (Aguirregabiria and Mira 2010).

Connections: We show that CCP is an instance of our class of problems characterized by (6-9). Comparing (14) with (7, 8), for this method, $\tilde{\pi} = \hat{\pi}_E$. Further, by specifying $\tilde{\pi} = \hat{\pi}_E$, we obtain $\mathbf{F}(s,a) = \frac{\partial Q_{\boldsymbol{\theta}}^{\tilde{\pi}}(s,a)}{\partial \boldsymbol{\theta}}$. From (15):

$$\sum_\tau \mathbb{E}_{P_E(\mathbf{s}_{1:\tau},\mathbf{a}_{1:\tau})}\left[\frac{\partial Q_{\boldsymbol{\theta}}^{\tilde{\pi}}(s_\tau,a_\tau)}{\partial \boldsymbol{\theta}} - \sum_a \pi_{\boldsymbol{\theta}}(a|s_\tau)\frac{\partial Q_{\boldsymbol{\theta}}^{\tilde{\pi}}(s_\tau,a)}{\partial \boldsymbol{\theta}}\right] = 0 \quad (16)$$

Notice that this is equivalent to setting our objective gradient (9) to zero. This occurs at the optimum of our objective (6).

We highlight here that the CCP method is more computationally efficient compared to NFXP and MCE-IRL. At every gradient update step, NFXP and MCE-IRL require *optimizing* the soft value $Q_{\boldsymbol{\theta}}^*$ (13) to obtain $\pi_{\boldsymbol{\theta}}$ (10). On the other hand, the CCP method only *improves* the policy $\pi_{\boldsymbol{\theta}}$ (14), which only requires updating the soft value $Q_{\boldsymbol{\theta}}^{\hat{\pi}_E}$. We show in Section 5 how this is more computationally efficient.

4.3 Nested Pseudo-Likelihood Algorithm

Unlike the CCP method which solves the likelihood objective (6) once, NPL is based on the idea of repeated refinement of the approximate optimal soft value $Q_{\boldsymbol{\theta}}^*$. This results in a refined CCP estimate (10, 14), and, thus, a refined objective (6). NPL solves the objective repeatedly, and its first iteration is equivalent to the CCP method. The authors (Aguirregabiria and Mira 2002) prove that this iterative refinement converges to the NFXP (Rust 1988) solution, as long as the first estimate $\tilde{\pi} = \hat{\pi}_E$ is "sufficiently close" to the true optimal policy in the soft value sense. We refer the reader to (Kasahara and Shimotsu 2012) for a discussion on convergence criteria.

Connections: The initial policy $\tilde{\pi}^1 = \hat{\pi}_E$ is estimated from observed data. Subsequently, $\tilde{\pi}^k = \pi_{\boldsymbol{\theta}^*}^{k-1}$, where $\pi_{\boldsymbol{\theta}^*}^{k-1}$ is the CCP under optimal reward parameters $\boldsymbol{\theta}^*$ from the $k-1^{\text{th}}$ iteration. We have discussed that NPL is equivalent to repeatedly maximizing the objective (6), where expert policy $\hat{\pi}_E$ is explicitly estimated, and CCP $\pi_{\boldsymbol{\theta}}^k$ (8) is derived from refined soft value approximation $Q_{\boldsymbol{\theta}}^{\tilde{\pi}^k} \approx Q_{\boldsymbol{\theta}}^*$.

4.4 Summary

Commonality: In this section, we demonstrated connections between reward parameter estimation methods, by developing a common class of optimization problems characterized by general forms (6-9). Table 1 summarizes this section, with specifications of $\tilde{\pi}$ for each method, and the type of computation required at every gradient step.

Differences: In Section 4.1, we discussed that the computation of the NFXP (or MCE-IRL) gradient (11) involves solving the forward decision problem *exactly* in order to correctly infer the reward function. This requires computing a policy $\tilde{\pi} = \pi_{\boldsymbol{\theta}}$ that is *optimal* in the soft value sense. On the other hand, we discussed in Sections 4.2-4.3 that the computation of the CCP and NPL gradient involves an *approximation* of the optimal soft value. This only requires computing the policy $\pi_{\boldsymbol{\theta}}$ that is an *improvement* over $\tilde{\pi}$ in the soft value sense. This insight lays the groundwork necessary to compare the methods. The approximation of the soft value results in algorithmic and computational differences between the methods, which we make explicit in Section 5. Approximating the soft value results in a trade-off between correctness of the inferred solution and its computational burden. The implication of these differences (*i.e.,* approximations) on the suitability of each method is discussed in Section 6.

Objective: $g(\boldsymbol{\theta}) = \mathbb{E}_{P_E(\mathbf{s},\mathbf{a})}\left[\sum_\tau \log \pi_{\boldsymbol{\theta}}(a_\tau|s_\tau)\right]$.
Policy: $\pi_{\boldsymbol{\theta}}(a|s) = \text{softmax } Q_{\boldsymbol{\theta}}^{\tilde{\pi}}(s,a)$. $Q_{\boldsymbol{\theta}}^{\tilde{\pi}}$ is soft value. (7)

Method → **Characteristic ↓**	MCE-IRL = NFXP	CCP	NPL
Specification of $\tilde{\pi}$	$\pi_{\boldsymbol{\theta}}$	$\hat{\pi}_E$	$\hat{\pi}_E$, $\pi_{\boldsymbol{\theta}^*}^1$, $\pi_{\boldsymbol{\theta}^*}^2$, ...
Gradient step computation	Soft Value optimization	Policy improvement	Policy improvement

Table 1: Summary of the common objective and policy form, and specifications for each method.

5 An Algorithmic Perspective

In this section, we explicitly illustrate the algorithmic differences that arise due to differences in the computation of the soft value function. The development of this perspective is important for us to demonstrate how, as a result of approximation, NPL has a more computationally efficient reward parameter update compared to MCE-IRL.

Optimization-based Methods: As described in Section 4.1, NFXP and MCE-IRL require the computation of the optimal soft value $Q_{\boldsymbol{\theta}}^*$ (13). Thus, we call these approaches "optimization-based methods" and describe them in Alg. 1. We define the future state occupancy for s' when following policy π: $\text{Occ}^\pi(s') = \sum_t P^\pi(s_t = s')$. The gradient (11) can be expressed in terms of occupancy measures:

$$\boldsymbol{\mu}^{\hat{\pi}_E} = \sum_{s'} \text{Occ}^{\hat{\pi}_E}(s')\mathbb{E}_{a'\sim\hat{\pi}_E}\left[\frac{\partial r(s',a',\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right], \quad (17)$$

$$\boldsymbol{\mu}^{\pi_{\boldsymbol{\theta}}} = \sum_{s'} \text{Occ}^{\pi_{\boldsymbol{\theta}}}(s')\mathbb{E}_{a'\sim\pi_{\boldsymbol{\theta}}}\left[\frac{\partial r(s',a',\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] \quad (18)$$

$$\frac{\partial g}{\partial \boldsymbol{\theta}} = \boldsymbol{\mu}^{\hat{\pi}_E} - \boldsymbol{\mu}^{\pi_{\boldsymbol{\theta}}} \quad (19)$$

Approximation-based Methods: As described in Sections (4.2, 4.3), CCP and NPL avoid optimizing the soft value by approximating $Q_{\boldsymbol{\theta}}^* \approx Q_{\boldsymbol{\theta}}^{\tilde{\pi}}$ using a policy $\tilde{\pi}$. We call these approaches "approximation-based methods" and describe them in Algorithm 2. Note, $K = 1$ is the CCP Method. We define the future state occupancy for s' when beginning in (s,a) and following policy π : $\text{Occ}^\pi(s'|s,a) =$

Algorithm 1: Optimization-based Method

Input: Expert demonstrations.
Result: Reward params. $\boldsymbol{\theta}^*$, policy $\pi_{\boldsymbol{\theta}^*}$.
1 Estimate expert policy $\hat{\pi}_E$.
2 **Evaluate** $\text{Occ}^{\hat{\pi}_E}(s')\ \forall s'$.
3 **repeat (update reward)**
4 **Optimize** soft value $Q^*_{\boldsymbol{\theta}}$. (13).
5 Compute $\pi_{\boldsymbol{\theta}} = \text{softmax}\ Q^*_{\boldsymbol{\theta}}(s,a)$. (10).
6 **Evaluate** $\boldsymbol{\mu}^{\pi_{\boldsymbol{\theta}}}$. (17).
7 Update gradient $\frac{\partial g}{\partial \boldsymbol{\theta}}$. (19).
8 Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \frac{\partial g}{\partial \boldsymbol{\theta}}$.
9 **until** $\boldsymbol{\theta}$ *not converged*
10 $\pi_{\boldsymbol{\theta}^*} \leftarrow \pi_{\boldsymbol{\theta}}, \quad \boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$.

$\sum_t P^\pi(s_t = s' \mid (s_0, a_0) = (s,a))$. (7, 9) can be written in terms of occupancy measures as follows:

$$Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s,a) = r(s,a,\boldsymbol{\theta}) + \textstyle\sum_{s'} \text{Occ}^{\tilde{\pi}}(s'|s,a)\mathbb{E}_{a'\sim\tilde{\pi}}\big[r(s',a',\boldsymbol{\theta}) - \log\tilde{\pi}(a'|s')\big], \quad (20)$$

$$\frac{\partial Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s,a)}{\partial \boldsymbol{\theta}} = \frac{\partial r(s,a,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \textstyle\sum_{s'} \text{Occ}^{\tilde{\pi}}(s'|s,a)\mathbb{E}_{a'\sim\tilde{\pi}}\left[\frac{\partial r(s',a',\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] \quad (21)$$

$$\frac{\partial g}{\partial \boldsymbol{\theta}} = \textstyle\sum_{s'} \text{Occ}^{\hat{\pi}_E}(s')\Big(\mathbb{E}_{a'\sim\hat{\pi}_E}\left[\frac{\partial Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s',a')}{\partial \boldsymbol{\theta}}\right] - \mathbb{E}_{a'\sim\pi_{\boldsymbol{\theta}}}\left[\frac{\partial Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s',a')}{\partial \boldsymbol{\theta}}\right]\Big). \quad (22)$$

Reward Update: NPL has a very efficient reward parameter update (*i.e.*, inner) loop (Alg. 2: Lines 6-12), compared to the update loop of MCE-IRL (Alg. 1: Lines 3-9). Each gradient step in MCE-IRL (Alg. 1) involves expensive dynamic programming for: (1) optimizing soft value (Line 4, (13)), and (2) evaluating $\boldsymbol{\mu}^{\pi_{\boldsymbol{\theta}}}$ by computing occupancy measures $\text{Occ}^{\pi_{\boldsymbol{\theta}}}(s')$ (Line 6, (13)). On the other hand, each gradient step in NPL (Alg. 2) only involves: (1) updating soft value $Q^{\tilde{\pi}}_{\boldsymbol{\theta}}$ (Line 7, (20)), and (2) updating value gradient (Line 9, (21)). Both steps can be efficiently performed without dynamic programming, as the occupancy measures $\text{Occ}^{\tilde{\pi}}(s'|s,a)$ can be pre-computed (Line 5). The value and value gradient (20, 21) are linearly dependent on reward and reward gradient respectively, and can be computed in one step using matrix multiplication. We elaborate this point in Sanghvi et al. 2021 (Appendix B). The gradient update step in both algorithms (Alg. 1: Line 7, Alg. 2: Line 10) has the same computational complexity, *i.e.,* linear in the size of the environment.

The outer loop in NPL (Alg. 2: Lines 4-14) converges in very few iterations (< 10) (Aguirregabiria and Mira 2002). Although computing occupancy measures $\text{Occ}^{\tilde{\pi}}(s'|s,a)$ (Line 5) requires dynamic programming, the number of outer loop iterations is many order of magnitudes fewer than the number of inner loop iterations. Since Alg. 2 avoids dynamic programming in the inner reward update loop, approximation-based methods are much more *efficient* than optimization-based methods (Alg. 1). We make explicit the comparison of computational load in Sanghvi et al. 2021 (Appendix B).

Algorithm 2: Approximation-based method

Input: Expert demonstrations.
Result: Reward params. $\boldsymbol{\theta}^*$, policy $\pi_{\boldsymbol{\theta}^*}$.
1 Estimate expert policy $\hat{\pi}_E$.
2 Initialize $\tilde{\pi} \leftarrow \hat{\pi}_E$.
3 **Evaluate** $\text{Occ}^{\hat{\pi}_E}(s')\ \forall s'$.
4 **for** *k = 1 ... K* **do (update policy)**
5 **Evaluate** $\text{Occ}^{\tilde{\pi}}(s'|s,a)\ \forall (s',s,a)$.
6 **repeat (update reward)**
7 Update value $Q^{\tilde{\pi}}_{\boldsymbol{\theta}}$. (20).
8 Improve $\pi_{\boldsymbol{\theta}} = \text{softmax}\ Q^{\tilde{\pi}}_{\boldsymbol{\theta}}(s,a)$. (8).
9 Update $\frac{\partial Q^{\tilde{\pi}}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}$. (21).
10 Update gradient $\frac{\partial g}{\partial \boldsymbol{\theta}}$. (22).
11 Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \frac{\partial g}{\partial \boldsymbol{\theta}}$.
12 **until** $\boldsymbol{\theta}$ *not converged*
13 $\tilde{\pi} \leftarrow \pi_{\boldsymbol{\theta}}, \quad \pi_{\boldsymbol{\theta}^*} \leftarrow \pi_{\boldsymbol{\theta}}, \quad \boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$.
14 **end**

6 Suitability of Methods

In Section 5, we made explicit the computational and algorithmic differences between optimization (MCE-IRL, NFXP) and approximation-based (CCP, NPL) methods. While approximation-based methods outperform optimization-based methods in terms of computational efficiency, the approximation the soft value introduces a trade-off between the correctness of the inferred solution and its computational burden. For some types of problems, trading the quality of the inferred reward for computational efficiency is unreasonable, so optimization-based methods are more suitable. Using theory we developed in Section 4, in this section, we develop hypotheses about the hierarchy of methods in various types of problem situations, and investigate each hypothesis using an example. We quantitatively compare the methods using the following metrics (lower values are better):

- **Negative Log Likelihood (NLL)** evaluates the likelihood of the expert path under the predicted policy $\pi_{\boldsymbol{\theta}}*$, and is directly related to our objective (5).
- **Expected Value Difference (EVD)** is value difference of two policies under true reward: 1) optimal policy under true reward and 2) optimal policy under *output* reward $\boldsymbol{\theta}^*$.
- **Stochastic EVD** is the value difference of the following policies under *true* reward: 1) optimal policy under the true reward and 2) the *output* policy $\pi_{\boldsymbol{\theta}}*$. While a low Stochastic EVD may indicate a better output policy $\pi_{\boldsymbol{\theta}^*}$, low EVD may indicate a better output reward $\boldsymbol{\theta}^*$.
- **Equivalent-Policy Invariant Comparison (EPIC)** (Gleave et al. 2020) is a recently developed metric that measures the distance between two reward functions without training a policy. EPIC is shown to be invariant on an equivalence class of reward functions that always induce the same optimal policy. The EPIC metric $\in [0,1]$ with lower values indicates similar reward functions.

EVD and EPIC evaluate inferred reward $\boldsymbol{\theta}^*$, while NLL and Stochastic-EVD evaluate the inferred policy $\pi_{\boldsymbol{\theta}^*}$. In addition

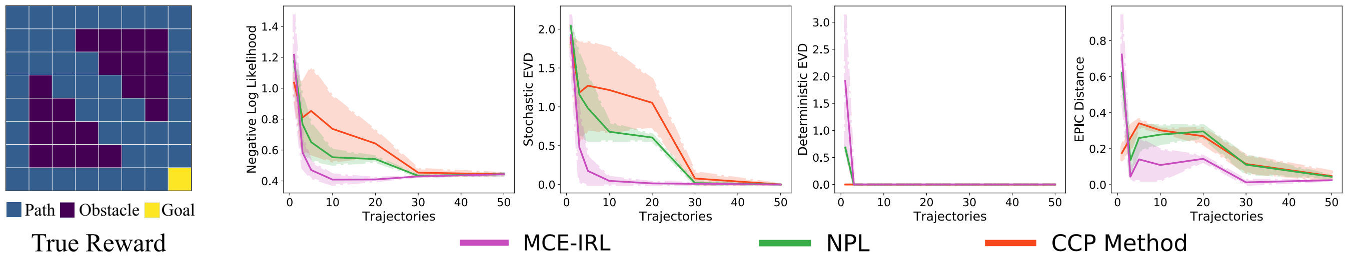

Figure 1: Obstacleworld. Leftmost: True reward. Graphs show comparative performance (metrics vs. number of trajectories).

to evaluating recovered reward, evaluating $\pi_{\boldsymbol{\theta}^*}$ is important because different rewards might induce policies that perform similarly well in terms of our objective (6). We provide experimental details in Sanghvi et al. 2021 (Appendices C, D).

Method Dependencies: From Section 4, we observe that the MCE-IRL gradient (12) depends on the expert policy estimate $\hat{\pi}_E$ only through the expected *feature count* estimate $\mathbb{E}_{\hat{P}_E(\mathbf{s},\mathbf{a})}\left[\sum_t \mathbf{f}(s_t, a_t)\right]$. In non-linear reward settings, the dependence is through the expected *reward gradient* estimate $\mathbb{E}_{\hat{P}_E(\mathbf{s},\mathbf{a})}\left[\sum_t \partial r(s_t, a_t, \boldsymbol{\theta})/\partial\boldsymbol{\theta}\right]$. On the other hand, the NPL (or CCP) gradient (16) depends on the expert policy estimate for estimating a state's *importance relative* to others (*i.e.,* state occupancies under the estimated expert policy), and for *approximating* the optimal soft value.

From these insights, we see that the suitability of a method for a problem depends on: (1) the amount of expert data, and on (2) how "good" the resultant estimates and approximations are in that scenario. In the following subsections, we introduce different problem scenarios, each characterized by the goodness of these estimates, and investigate our hypotheses regarding each method's suitability for those problems.

6.1 Well-Estimated Feature Counts

Scenario: We first investigate scenarios where feature counts can be estimated well even with little expert data. In such scenarios, the feature representation allows distinct states to be correlated. For example, the expert's avoidance of *any* state with obstacles should result in identically low preference for *all* states with obstacles. Such a scenario would allow expert feature counts to be estimated well, even when expert data only covers a small portion of the state space.

Hypothesis: Optimization-based methods will perform better than approximation-based methods in low data regimes, and converge to similar performance in high data regimes.

Reasons: If feature counts can be estimated well from small amounts of data, MCE-IRL (= NFXP) is expected to converge to the correct solution. This follows directly from method dependencies outlined above. On the other hand, NPL (and CCP) require good estimates of a state's relative importance and soft value in order to perform similarly well. Since low data regimes do not allow these to be well-estimated, approximation-based methods are expected to perform as well as optimization-based ones only in high data regimes.

Experiment: We test our hypothesis using the Obstacleworld environment (Figure 1). We use three descriptive feature representations *(path, obstacle, goal)* for our states. Since this representation is simultaneously informative for a *large* set of states, we can estimate feature counts well even with little expert data. The true reward is a linear function of features $(path:\ 0.2, obstacle:\ 0.0, goal:\ 1.0)$.

In Figure 1, we observe that in low data-regimes (i.e. with few trajectories) MCE-IRL performs well on all metrics. However, with low expert data, CCP and NPL perform poorly (*i.e.,* high NLL, Stochastic EVD, EPIC). With more expert data, CCP method and NPL converge to similar performance as MCE-IRL. This is in agreement with our hypothesis.

6.2 Correlation of Feature Counts and Expert Policy Estimates

Scenario: We now investigate the scenario where the goodness of feature count and expert policy estimates becomes correlated. In other words, a high amount of expert data is required to estimate feature counts well. This scenario may arise when feature representations either (1) incorrectly discriminate between states, or (2) are not informative enough to allow feature counts to be estimated from little data.

Hypothesis: Both optimization-based and approximation-based methods will perform poorly in low expert data regimes, and do similarly well in high expert data regimes.

Reasons: In this scenario, the goodness of feature count, relative state importance and optimal soft value estimates is similarly dependent on the amount of expert data. Thus we expect, optimization- and approximation-based methods to perform poorly in low data regimes, and similarly well in high data regimes.

Experiment: We investigate our hypothesis in the MountainCar environment, with a feature representation that discriminates between all states. Thus, state features are defined as one-hot vectors. MountainCar is a continuous environment where the state is defined by the position and velocity of the car. The scope of this work is limited to discrete settings with known transition dynamics. Accordingly, we estimate the transition dynamics from continuous expert trajectories using kernels, and discretize the state space to a large set of states (10^4). We define the true reward as distance to the goal.

In Figure 2, we observe that in low-data regimes all methods perform poorly with high values for all metrics. As the amount of expert data increases, the performance of each method improves. More importantly, around the same number of trajectories (≈ 400) all methods perform equally well, with a similar range of values across all metrics. This is in agreement with our hypothesis.

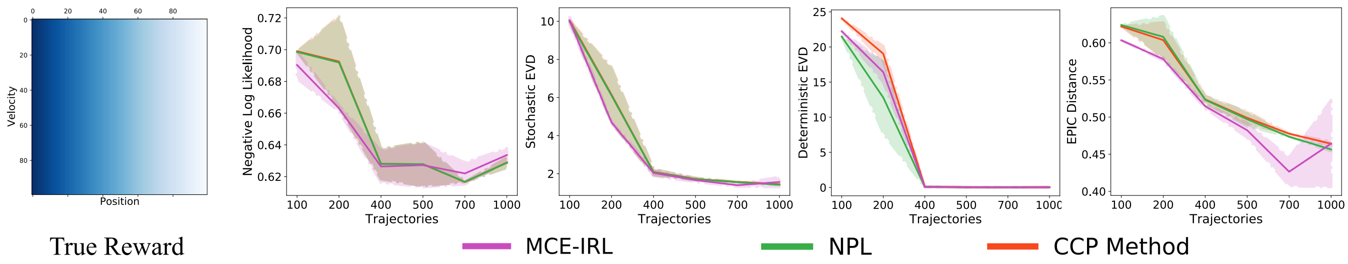

Figure 2: MountainCar. Leftmost: True reward. Lighter colors indicate higher reward. Graphs show comparative performance.

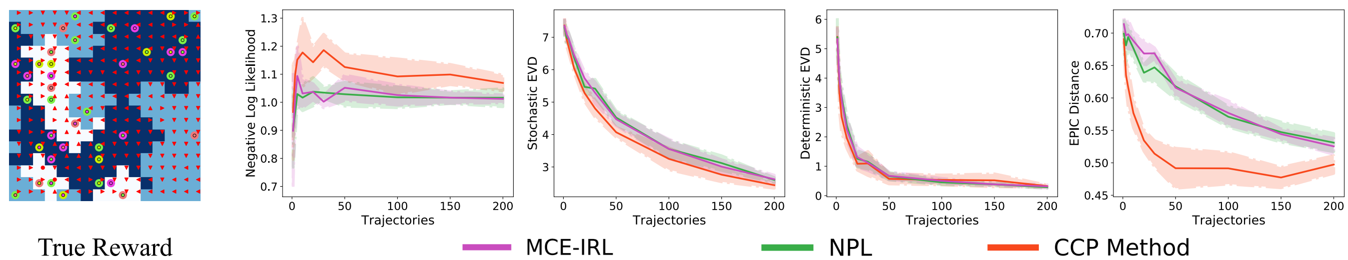

Figure 3: Objectworld. Leftmost: True reward in 16^2 grid with 4 colors. Dark and light colors indicate low and high reward respectively. Red arrows represent the optimal policy. Graphs show comparative performance in 32^2 grid with 4 colors.

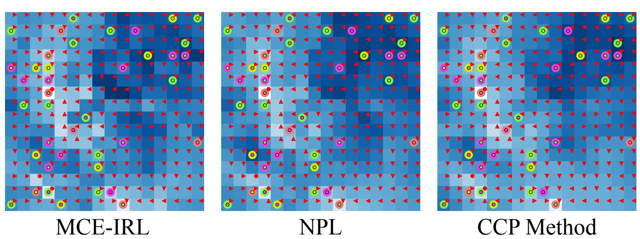

Figure 4: Objectworld. Recovered reward in 16^2 grid, 4 colors with 200 expert trajectories as input.

6.3 Deep Reward Representations

Scenario: We investigate scenarios where rewards are deep neural network representations of state-action, as opposed to linear representations explored in previous subsections.

Hypothesis: Optimization-based methods either perform better or worse than the approximation-based methods in low data regimes and perform similarly well in high data regimes.

Reasons: From (11), the gradient of optimization-based methods depends on the expert policy estimate through the expected reward gradient. Comparing (11) and (12), we can think of the vector of reward gradients $\partial r(s_t, a_t, \boldsymbol{\theta})/\partial \boldsymbol{\theta}$ as the state-action feature vector. During learning, since this feature vector is dependent on the current parameters $\boldsymbol{\theta}$, the statistic $\mathbb{E}_{\hat{P}_E(\mathbf{s},\mathbf{a})}\left[\sum_t \partial r(s_t, a_t, \boldsymbol{\theta})/\partial \boldsymbol{\theta}\right]$ is a non-stationary target in the MCE-IRL gradient. In the low data regime, at every gradient step, this could either be well-estimated (similar to Section 6.1) or not (similar to Section 6.2), depending on the capacity and depth of the network. On the other hand, in high data regimes, we can expect reward gradient, relative state importance and soft value to all be estimated well, since the expert policy can be estimated well.

Experiment: We investigate the hypothesis using the Objectworld environment (Wulfmeier, Ondruska, and Posner 2015) which consists of a non-linear feature representation. Objectworld consists of an N^2 grid and randomly spread through the grid are objects, each with an inner and outer color chosen from C colors. The feature vector is a set of continuous values $x \in \mathbb{R}^{2C}$, where x_{2i} and x_{2i+1} are state's distances from the *i'th* inner and outer color.

From Figure 3, in low-data regimes, all methods perform poorly with high values for all metrics. With more expert data, the performance for all methods improve and converge to similar values. Similar results for MCE-IRL were observed in (Wulfmeier, Ondruska, and Posner 2015). Consistent with our hypothesis, in this environment we observe no difference in the performance of optimization- and approximation-based methods in both low and high data regimes.

6.4 Discussion

In Sections 6.1-6.3, we described three problem scenarios and discussed the performance of optimization- and approximation-based methods. We now discuss the suitability of methods for these scenarios.

High data regimes: In all scenarios we discussed, in high data regimes, both optimization- and approximation-based methods perform similarly well on all metrics (Figures 1-3). Qualitatively, all methods also recover similar rewards in high data regimes (Figures 4, 7 (Appendix D, Sanghvi et al. 2021)). This is because, as stated in Section 4, NPL converges to NFXP when the expert policy is well-estimated, *i.e.*,, when more data is available. Further, approximation-based methods are significantly more computationally efficient than optimization-based methods (Section 5). This finding is em-

Settings	MCE-IRL	NPL	CCP Method
	MountainCar		
State Size: 100^2	4195.76	589.21 ($\times$7)	193.95 ($\times$**22**)
	ObjectWorld		
Grid: 16^2, C: 4	32.21	4.18 ($\times$8)	2.40 ($\times$ **13**)
Grid: 32^2, C: 2	638.83	30.63 ($\times$21)	14.00 ($\times$ **46**)
Grid: 32^2, C: 4	471.64	29.85 ($\times$16)	11.19 ($\times$ **42**)
Grid: 64^2, C: 4	10699.47	340.55 ($\times$31)	103.79 ($\times$ **103**)

Table 2: Training time (secs) averaged across multiple runs. Numbers in brackets indicate speed up against MCE-IRL.

pirically supported in Table 2. From these observations, we conclude that approximation-based methods are more suitable than optimization-based methods in high data regimes.

Low data regimes: In Sections 6.2-6.3, we introduced two scenarios where approximation- and optimization-based methods both perform similarly (poorly) in low-data regimes (Figures 2, 3). Since approximation-based methods always outperform optimization-based methods in terms of computational efficiency (Table 2), in these scenarios, approximation-based methods are more suitable. On the other hand, optimization-based methods are more suitable when feature counts can be estimated well from little data (Section 6.1).

6.5 Conclusions

In this work, we explicitly derived connections between four methods of reward parameter estimation developed independently in the fields of economics and ML. To the best of our knowledge, we are the first to bring these methods under a common umbrella. We achieved this by deriving a class of optimization problems, of which each method is a special instance. We showed how a difference in the estimation of the optimal soft value results in different specifications of the explicit policy $\tilde{\pi}$, and used our insights to demonstrate algorithmic and computational differences between methods. Using this common form we analyzed the applicability of each method in different problem settings. Our analysis shows how approximation-based methods are superior to optimization-based methods in some settings and vice-versa.

Additionally, approximation-based approaches have been applied to situations with continuous state or action spaces (Altuğ and Miller 1998). Such settings are outside of the scope of this paper and we leave their discussion for future work. In this work, our goal is to explicitly demonstrate connections in the discrete problem setting, to facilitate further inter-disciplinary work in this area.

Future Work: Finally, we touch upon interesting directions to explore based on the theoretical framework developed in this work. The first of these is leveraging our derived connections to investigate approximation-based methods from an optimization perspective. Specifically, we propose to work on the characterization of the primal-dual optimization forms of these methods. Since many IRL methods (including adversarial imitation learning) use an optimization perspective, we believe this will not only lead to new algorithmic advances, but will also shed more light on the similarities and differences between our approaches and more recent IRL methods. Another direction we plan to explore is to use our explicit algorithmic perspectives for practical settings where MCE-IRL is intractable, such as problems with very large state spaces, *e.g.* images in activity forecasting. For such situations, our work details how approximation-based methods can be applied in a principled manner when expert data is readily available. We hope to apply our insights to problems such as activity forecasting, social navigation and human preference learning.

Acknowledgments

This work was sponsored in part by IARPA (D17PC00340). We thank Dr. Robert A. Miller for his support in this work.

References

Abbeel, P.; and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, 1. ACM.

Aguirregabiria, V.; and Mira, P. 2002. Swapping the nested fixed point algorithm: A class of estimators for discrete Markov decision models. *Econometrica* 70(4): 1519–1543.

Aguirregabiria, V.; and Mira, P. 2010. Dynamic discrete choice structural models: A survey. *Journal of Econometrics* 156(1): 38–67.

Altuğ, S.; and Miller, R. A. 1998. The effect of work experience on female wages and labour supply. *The Review of Economic Studies* 65(1): 45–85.

Boularias, A.; Kober, J.; and Peters, J. 2011. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 182–189.

Dvijotham, K.; and Todorov, E. 2010. Inverse optimal control with linearly-solvable MDPs. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 335–342.

Gleave, A.; Dennis, M.; Legg, S.; Russell, S.; and Leike, J. 2020. Quantifying differences in reward functions. *arXiv preprint arXiv:2006.13900* .

Haarnoja, T.; Tang, H.; Abbeel, P.; and Levine, S. 2017. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165* .

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290* .

Hotz, V. J.; and Miller, R. A. 1993. Conditional Choice Probabilities and the Estimation of Dynamic Models. *Review of Economic Studies* 60: 497–529.

Kasahara, H.; and Shimotsu, K. 2012. Sequential estimation of structural models with a fixed point constraint. *Econometrica* 80(5): 2303–2319.

Klein, E.; Geist, M.; Piot, B.; and Pietquin, O. 2012. Inverse reinforcement learning through structured classification. In *Advances in Neural Information Processing Systems*, 1007–1015.

Klein, E.; Piot, B.; Geist, M.; and Pietquin, O. 2013. A cascaded supervised learning approach to inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 1–16. Springer.

Levine, S.; and Koltun, V. 2012. Continuous inverse optimal control with locally optimal examples. *arXiv preprint arXiv:1206.4617* .

McFadden, D.; et al. 1973. Conditional Logit Analysis of Qualitative Choice Behavior. In *Frontiers in Econometrics*, 105–142.

Miller, R. A. 1984. Job matching and occupational choice. *Journal of Political Economy* 92(6): 1086–1120.

Pakes, A. 1986. Patents as Options: Some Estimates of the Value of Holding European Patent Stocks. *Econometrica* 54(4): 755–84.

Ratliff, N. D.; Bagnell, J. A.; and Zinkevich, M. A. 2006. Maximum margin planning. *Proceedings of the 23rd International Conference on Machine Learning (ICML)* 729–736. doi:10.1145/1143844.1143936.

Rust, J. 1987. Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher. *Econometrica* 55: 999–1033.

Rust, J. 1988. Maximum likelihood estimation of discrete control processes. *SIAM Journal on Control and Optimization* 26(5): 1006–1024.

Sanghvi, N.; Usami, S.; Sharma, M.; Groeger, J.; and Kitani, K. 2021. Inverse Reinforcement Learning with Explicit Policy Estimates. *arXiv preprint arXiv:2103.02863* .

Sharma, M.; Kitani, K. M.; and Groeger, J. 2017. Inverse Reinforcement Learning with Conditional Choice Probabilities. *arXiv preprint arXiv:1709.07597* .

Todorov, E. 2007. Linearly-solvable Markov decision problems. In *Advances in neural information processing systems*, 1369–1376.

Wolpin, K. I. 1984. An estimable dynamic stochastic model of fertility and child mortality. *Journal of Political economy* 92(5): 852–874.

Wulfmeier, M.; Ondruska, P.; and Posner, I. 2015. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888* .

Ziebart, B. D. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Ph.D. thesis, Carnegie Mellon University.