Annual Reviews in Control xxx (xxxx) xxx

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

Objective learning from human demonstrations

Jonathan Feng-Shun Lin [a], Pamela Carreno-Medrano [b], Mahsa Parsapour [c], Maram Sakr [b,d], Dana Kulić [b,*]

[a] Systems Design Engineering, University of Waterloo, Canada
[b] Faculty of Engineering, Monash University, Australia
[c] Electrical and Computer Engineering, University of Waterloo, Canada
[d] Mechanical Engineering, University of British Columbia, Canada

ARTICLE INFO

Keywords:
Reward learning
Inverse optimal control
Inverse reinforcement learning

ABSTRACT

Researchers in biomechanics, neuroscience, human–machine interaction and other fields are interested in inferring human intentions and objectives from observed actions. The problem of inferring objectives from observations has received extensive theoretical and methodological development from both the controls and machine learning communities. In this paper, we provide an integrating view of objective learning from human demonstration data. We differentiate algorithms based on the assumptions made about the objective function structure, how the similarity between the inferred objectives and the observed demonstrations is assessed, the assumptions made about the agent and environment model, and the properties of the observed human demonstrations. We review the application domains and validation approaches of existing works and identify the key open challenges and limitations. The paper concludes with an identification of promising directions for future work.

1. Introduction

Understanding human intentions and objectives from observed actions is of interest in a number of fields, including physiology, neuroscience, biomechanics, human–machine interaction and robotics (Kulić et al., 2016). In robotics, imitation learning, or learning from demonstration (Argall et al., 2009; Billard et al., 2008; Fang et al., 2019; Hussein et al., 2017; Kober et al., 2013; Schaal, 1997) have been deployed to learn robot behaviors from human demonstration data. Demonstration data can include human movement, as well as human-piloted robot movement, through tele-operation or kinesthetic teaching (Argall et al., 2009).

To infer the objective of a demonstration, the trajectory executed by the human is assumed to be optimal with respect to some unknown objective function. Estimating this objective function provides an abstracted and parsimonious representation of the task (Ng & Russell, 2000), with the ability to: (1) model and generate new trajectories, (2) provide insight into why a given trajectory was selected, out of all possible trajectories, and (3) generalize demonstration motions to other robot embodiments or tasks. Objective learning methods have been developed from two research communities: the control community, where they are known as *inverse optimal control* (IOC) and the machine learning community, where they are known as *inverse reinforcement learning* (IRL) methods.

A number of prior survey papers have cataloged the state of the art in IOC/IRL, commonly as a section within a larger survey on robotics learning from demonstration (Argall et al., 2009; Kroemer et al., 2019; Liu et al., 2020; Ravichandar et al., 2020), reinforcement learning (Kober et al., 2013), and human movement modeling (Kulić et al., 2016). Zhifei and Er (2012) and Arora and Doshi (2018) provide comprehensive algorithmic overviews of IRL methods and recent applications. Recently, Ab Azar et al. (2020) provide the first IOC/IRL survey paper, providing an extensive overview of papers from both fields. While their paper provides a historical overview of IOC/IRL papers, this manuscript is focused on providing a holistic framework to provide a synthesizing view of both IOC and IRL methods, focusing on objective learning from human demonstrations.

2. Problem formulation

When analyzing human demonstrations using objective learning techniques, it is assumed that the human demonstrator is generating optimal trajectories according to the (unknown) objectives. As illustrated

* Corresponding author.
E-mail addresses: jonathan.lin@uwaterloo.ca (J.F.-S. Lin), pamela.carreno@monash.edu (P. Carreno-Medrano), mahsa.parsapour@uwaterloo.ca (M. Parsapour), maram.sakr@ubc.ca (M. Sakr), dana.kulic@monash.edu (D. Kulić).

[1] We adopt the optimal control notation (x, u) for state–action pairs, the RL notation is typically (s, a).

https://doi.org/10.1016/j.arcontrol.2021.04.003
Received 19 January 2021; Accepted 2 April 2021

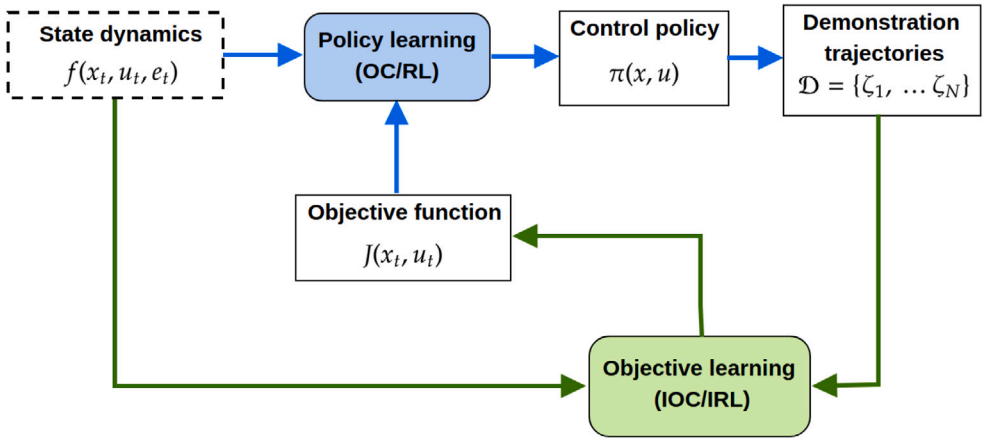

Fig. 1. Graphic depiction of the policy learning (forward) (blue arrows) and objective learning (inverse) problems (green arrows). The dashed lines around state dynamics indicate that for some algorithms, this function is unknown.

in Fig. 1, we start with a dynamical system – which may represent the human or robot body acting in an environment – which evolves according to some function dependent on the current system state and the control actions. In the most general case, the system dynamics are stochastic, *i.e.*, $\mathcal{T} = \{p(x_{t+1}|x_t, u_t)\}$. More commonly the system dynamics are assumed to be deterministic and are approximated by a function $x_{t+1} = f(x_t, u_t, e_t)$[1] that maps the current state of the system $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ and control action $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ at time t to a new state x_{t+1}, with some disturbance e_t. The agent can assess its current state x_t and the control action u_t via the objective function $J(x_t, u_t)$.

The goal of the *forward* problem is to find a control policy (or simply "a policy") π that optimizes the expected cumulative return starting from an initial state x_0. The policy π is a function that maps a sequence of states and control actions $\zeta_{0:t} = \{(x_0, u_0), \ldots, (x_t, u_t)\}$ up to time t to a new control action[2]. The cumulative return is given by:

$$V^{\pi}(x_0) = \mathbb{E}_{e_t}[\sum_{t=1}^{T} J(x_t, u_t)|x_0, \pi]$$

where T is the duration of the trajectory, and $\mathbb{E}_{e_t}[\cdot]$ indicates the expectation over the variability introduced by the disturbance. For continuous action problems, where T may be infinite, a discount factor $\gamma \in [0, 1]$ is added when computing the expected cumulative return (*i.e.*, $V^{\pi}(x_0) = \mathbb{E}_{e_t}[\sum_{t=1}^{T} \gamma^{t-1} J(x_t, u_t)|x_0, \pi]$). This discount factor decays future rewards as a function of time. $V^{\pi}(x) : \mathcal{X} \rightarrowtail \mathbb{R}$ is also known as the *state-value function* (or simply "value function") under π.

Formally, when the objective function is known, we aim to solve the following optimization problem:

$$\begin{aligned} \text{opt}_{\pi} \quad & V^{\pi}(x_0) \\ \text{s.t.} \quad & x_{t+1} = f(x_t, u_t, e_t) \\ & u_t = \pi(\zeta_{0:t}) \\ & g(x_t) \leq 0 \\ \text{with} \quad & x_0 \sim p_0(x_0 = x), \end{aligned} \tag{1}$$

where $p_0(x_0 = x)$ denotes the initial state distribution. If the initial state is known, it is represented as a constraint on the initial state of the system. From Eq. (1), we observe that policy learning algorithms aim to optimize the *objective function* $J(x_t, u_t)$ with respect to the control actions u_t, subject to the *state dynamics* $f(x_t, u_t, e_t)$ and any additional constraints $g(x_t)$, and where the *policy* $\pi(\zeta_{0:t})$ is the decision variable. Most policy learning problems assume Markovian states and thus only knowledge about the current state x_t is needed in order to determine the next control action u_t. That is, the policy is a function of the current state x_t only (*i.e.*, $u_t = \pi(x_t)$).

Given these key elements, we now introduce the objective learning problem, also known as the *inverse* problem (*i.e.*, IOC/IRL). Given an observed set of motion demonstrations $\mathcal{D} = \{\zeta^1, \ldots, \zeta^N\}$ sampled from the unknown control policy of the human expert π^E, objective learning algorithms seek to find an objective function $\hat{J}(x, u)$ such that any optimal trajectory $\hat{\zeta}$ generated with respect to this function would match in some way those provided by the expert. With $\hat{\pi}$ as the optimal control policy learned from the candidate objective function $J(x, u)$, we formally introduce the objective learning problem as follows:

$$\hat{J}(x, u) = \max_{J(x,u)} \sum_{\hat{\zeta} \sim \hat{\pi}, \zeta \in D} S(\hat{\zeta}, \zeta), \tag{2}$$

where $S(\hat{\zeta}, \zeta)$ corresponds to a similarity criterion used to guide the search for the expert's unknown objective function. We note that objective learning algorithms may also have knowledge of the dynamics of the system. A summary of the key definitions and main notation elements introduced so far and their equivalence between the IOC and IRL fields is provided in Table 1.

In the remainder of this paper we review the state of the art in both the IOC and IRL literature and detail how the objective function is modeled and approximated (Section 4); the different approaches used to define the similarity criterion (Section 5), and how the system dynamics are modeled and or approximated (Section 6), as well as the properties of the observed expert's demonstrations (Section 7). Section 8 provides an overview of the validation approaches taken. Section 9 details current limitations of the state of the art, while Section 10 provides insight into the future directions for objective learning from human demonstrations.

3. Survey methodology

The aim of this survey was to analyze the state of the art in objective learning, focused on human demonstrations. The initial pool of papers considered in this survey was identified by utilizing the following search terms in Google Scholar: (1) inverse optimal control, (2) inverse reinforcement learning, (3) inverse optimization, (4) reward learning, and (5) reward estimation human demonstration. Papers that were not IOC or IRL methods were discarded after each papers' abstract was reviewed independently by two of the authors. This resulted in an initial list of 215 (71 IOC, 145 IRL) references.

From these search terms, survey papers that included IOC and/or IRL content also had their bibliography added to the initial pool of papers considered. This resulted in 391 references considered.

[2] Policies can also be stochastic and specify a distribution over actions conditioned on states (*i.e.*, $\pi : \mathcal{X} \times \mathcal{U} \rightarrowtail [0, 1]$) instead of mapping directly to a single action u.

Table 1
List of key variables and definitions used in this manuscript. Common terminologies used in the IOC/IRL literature are also included.

Symbol	Definition	IRL	IOC
$x \in \mathcal{X}$	The current condition of the system x, as a subset of all possible states $\mathcal{X}$	State set, $s \in S$	State space
$u \in \mathcal{U}$	Actions for the agent u, as a subset of all possible actions $\mathcal{U}$	Action set, $a \in \mathcal{A}$	Control space
$\mathcal{T}$	The system model that describes how the system state evolves by applying the control inputs	Transition distribution, $\{p(x_{t+1}\|x_t, u_t)\}$	System dynamics, $f(x_t, u_t, e_t)$
$\pi(x, u)$	The agent's approach for choosing actions	Policy	Controller
$J(x, u)$	Instantaneous reward/cost of being in state x and taking action u	Reward function	Running cost
$V(x, u)$	The function optimized by the optimal control policy	State-value function, $V^{\pi}(x)$	Value function
ζ (D)	Demonstration trajectory (set) used to estimate the objective function.		
t (T)	Time (trajectory duration)		
γ	Discount factor		
$\{\cdot\}_0$	Initial conditions		
$\{\hat{\cdot}\}$	Estimated value		

Each paper was then read in detail by at least one author and included into the survey if it: (1) described an algorithm for estimating an objective function in some form, and (2) was motivated by or utilized human demonstration data. Papers were excluded if: (3) the method required interactive learning and feedback, such as active learning, or (4) it involved multi-agent interaction. This resulted a final tally of 118 (43 IOC, 75 IRL) papers.

4. Objective function representation

The majority of objective learning algorithms assume that the objective function $J(x, u)$ is given by a linear combination of basis features $\boldsymbol{\phi}(x, u) \in \mathbb{R}^k$ with weight parameters $\boldsymbol{\theta}$ such that

$$J^{\boldsymbol{\theta}}(x, u) = \boldsymbol{\theta}^T \boldsymbol{\phi}(x, u) \tag{3}$$

The objective learning problem then corresponds to estimating the weight vector $\boldsymbol{\theta}$. However, it is not immediately clear how to select these basis features. Most works manually identify features relevant to the specific task studied. For example, Park et al. (2011) minimize the combination of total force and moment of force for 4-finger pressing tasks, while Park and Levine (2013) consider torque minimization, pelvis position and velocity, joint angle regularization, foot motion periodicity and arm swing features for locomotion.

Berret et al. (2011) summarized four types of features that are commonly included for human motion analysis: *kinematic features* such as velocity, acceleration and jerk, *dynamic features* such as torque and torque change, *geodesic features* such as path length, and *energy features* such as kinetic energy, work, positive work, and total absolute work.

If the system dynamics and optimal controller are assumed to be linear, the objective function is often assumed to penalize the states and control signal in quadratic form (El-Hussieny et al., 2016; El-Hussieny & Ryu, 2019; Li et al., 2011; Priess et al., 2014; Unni et al., 2017) to facilitate analytic solutions to the policy learning problem.

More recently, researchers have proposed approaches to relax this assumption on the structure of the objective function by using non-parametric models such as radial basis functions (Li et al., 2011; Terekhov & Zatsiorsky, 2011), Gaussian processes (Joukov & Kulic, 2017; Levine et al., 2011; Qiao & Beling, 2011) or neural networks (Finn, Levine, & Abbeel, 2016; Fu et al., 2017; Wulfmeier et al., 2015). Although Gaussian processes can capture complex relationships between features and scalar-value rewards as well as determine the saliency of each feature with respect to the relevance of the expert's demonstrations, they suffer from poor scaling with the number of samples. Neural networks also allow to model complex, non-linear objective functions with the additional advantages of a favorable computational complexity and good scaling to problems with large, potentially high-dimensional state spaces (Wulfmeier et al., 2015). However, they lack the structure typically encoded in hand-engineered features and thus require additional regularization techniques (Finn, Levine, & Abbeel, 2016) or the inclusion of a discriminator (Fu et al., 2017) in order to robustly scale to complex tasks. While these approaches can represent a richer and non-linear objective function structure, the interpretability of the objectives may be lost.

Several works proposed to directly learn both the features to be included in the objective function and the objective function parametrization. For instance, Levine et al. (2010) employs regression trees to construct basis features from a large collection of potential components (*e.g.*, aspects of the environment such as color of the road, presence of police, or the speed of the car). The constructed basis features correspond to logical conjunctions that are relevant to the observed demonstrations. Similarly, Choi and Kim (2013) propose an approach for learning the objective function as a non-linear function of atomic features.

The above algorithms assume that a single reward function guides the demonstrated behavior. However, for longer demonstrations, multiple sub-goals may partition the demonstration. A number of works have examined how to automatically detect and identify these sub-goals. The assumption that the whole trajectory shares a common objective function can be relaxed by applying objective learning on a fixed-length sliding window (Lin et al., 2016) or with a dynamically sized window based on the recoverability of the windowed data (Jin, Kulić, et al., 2019).

Michini and How (2012) and Michini et al. (2013) propose an approach to partition the demonstration space and find simple objective functions for each partition using a Bayesian non-parametric mixture model. In effect, the approach partitions the demonstration into sub-goals, each of which is represented with a single positive reward at a single coordinate in the state (or feature) space, and zero elsewhere. Park et al. (2020) propose an approach for simultaneously inferring the task objectives and constraints from a single demonstration. They build on Michini and How (2012), by hypothesizing a trajectory that consists of multiple partitions, each with its own goal and constraint. The goal is modeled as a simple state value, while the constraint places restrictions on which states the agent can visit when the constraint is active, *i.e.* each partition has its own constrained transition function. The number of partitions is not known *a priori*, and is estimated using a Dirichlet process.

In Sermanet et al. (2016), unsupervised and reinforcement learning methods are combined to simultaneously learn visual-based objective functions and sub-goals. To do so, the authors exploit the semantic features learned by pre-trained deep neural networks to automatically infer task goals, sub-goals, and visually relevant features from few demonstrations. Close to this idea, Jin, Petrich, et al. (2019) propose an algorithm that can directly infer task specifications from raw videos of human demonstrations. Under the assumption that each video image corresponds to a state, the authors propose to model task specifications

as task functions with unknown parameters (a task function is the equivalent of an objective function). Task functions map state changes to a vector of reward values and are approximated using neural networks whose parameters are learned from the observed demonstrations. The same algorithm was later used to infer the association relationship between geometric features from video demonstrations where such associations fully characterize the task performed by the expert demonstrator (Jin, Petrich, Dehghan, & Jagersand, 2020).

Conversely, if demonstration trajectories were generated by different experts, some experts might follow different objective criteria, resulting in multi-intent problems. Babes-Vroman et al. (2011) consider the case when there are multiple demonstration trajectories, corresponding to multiple intents. The objective is to simultaneously cluster the trajectories by intent, and estimate the intents. Their approach requires the number of clusters to be specified *a priori*. Dimitrakakis and Rothkopf (2011) and Choi and Kim (2012) address the case when the number of clusters is unknown, by using a Dirichlet process mixture model to model the distribution over objective functions. In Dimitrakakis and Rothkopf (2011), the prior is shared between demonstrations but for each demonstration the reward is estimated separately, while Choi and Kim (2012) also develop an approach to update the posterior over the distribution of objective functions using Metropolis–Hastings sampling.

Summary: The great majority of objective learning works assume a known objective function structure, as a weighted sum of state or action-state features. The objective learning problem then simplifies to estimating the feature weights. The selection of appropriate features remains an open question. Features can generally be classified into task-based or regularization features, or into kinematic, dynamic, geodesic or energy features. Recently, researchers have begun to relax the assumptions on objective function structure, by automatically identifying the features (from a large available set), considering non-parametric objective function models, as well as considering multiple cost function hypotheses, either along a single trajectory or between trajectories in the demonstration set.

5. Formulating and assessing similarity

As shown in Fig. 2, at a high-level, estimating an objective function from demonstrated expert behavior can be framed as an iterative process. Starting from an initial guess of the objective function, the estimate is improved in a two-step process: (1) a comparison step in which the similarity between the behavior induced by the current estimate of the objective function and the observed demonstrations is measured; and (2) an update step in which the current estimate is modified so as to increase the similarity between the induced and observed behaviors. Thus to solve the inverse problem, an accurate measure of similarity between the induced and observed behavior is critical. Note that the framing in Fig. 2 also implies that a solver for the forward problem for each new objective function candidate is required.

A key step in any objective learning algorithm is the definition of a similarity criterion that accurately characterizes the notion of closeness or similarity between the current estimate of the objective function and the true human objective function. This similarity function can then be used as an *optimization criterion* to find the best estimate of the objective function. Note that, since the human's objective function may be unknown (even to the demonstrator), a direct comparison between J and $\hat{J}$ may not be possible, and instead, proxy measures must be used, for example, by comparing the observed trajectories with those generated by optimizing the estimated objective function as shown in Fig. 2.

Furthermore, since objective learning is an ill-posed problem, that is, many objective functions can potentially explain or generate policies that realize the observed expert demonstrations (with degenerate solution such as a zero or constant reward included), it may be useful to choose an optimization criterion that encourages the selection of objective functions that make the observed expert behavior the only optimal behavior (Neu & Szepesvári, 2009), or that give preference to objective functions that maximally differentiate the observed expert policy from other, sub-optimal policies (Ng & Russell, 2000).

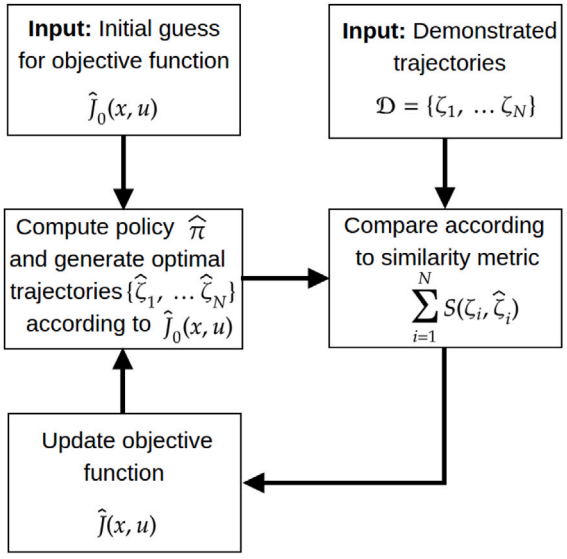

Fig. 2. High level overview of typical objective learning algorithms.

This section categorizes the existing methods based on how the similarity between the demonstrations and estimates is computed. We classify the approaches into the following categories: (1) feature expectations, where the objective function features are used to formulate the similarity criterion, (2) trajectory, where the demonstrated and estimated trajectories are directly compared using deterministic, ranked, or stochastic methods, (3) controller, where the controller parameters are compared, and (4) optimality criteria, where violations of optimality criteria of trajectories are directly minimized without an explicit similarity evaluation step.

5.1. Feature expectations

Given the parametrization of the objective function as a weighted sum of features (Eq. (3)), early works in the IRL literature proposed to compare demonstrations and generated trajectories based on the *feature expectations*. With this approach, the expected cumulative discounted feature counts, or more succinctly the feature expectations for a policy π, are defined as

$$\begin{aligned}\boldsymbol{\mu}(\pi) &= \mathbb{E}_{\zeta\sim\pi}\Big[\boldsymbol{\mu}(\zeta)\Big] \\ &= \mathbb{E}_{\zeta\sim\pi}\Big[\sum_{t=1}^{T}\gamma^{t-1}\boldsymbol{\phi}(x_t,u_t)\Big],\end{aligned} \tag{4}$$

where T indicates the duration of the demonstration trajectory and $\boldsymbol{\mu}(\zeta) = \sum_{t=1}^{T}\gamma^{t-1}\boldsymbol{\phi}(x_t,u_t)$ corresponds to the feature expectations along any trajectory sampled from a policy π. Using this notation, the value function of a policy π can be rewritten as $V^{\pi}(x) = \boldsymbol{\theta}^T\boldsymbol{\mu}(\pi)\forall\, x \in \mathcal{X}$. Given a set of expert demonstrations $D = \{\zeta_1,\ldots,\zeta_N\}$, we denote the empirical estimate of the expert's feature expectations by

$$\bar{\boldsymbol{\mu}}(\pi^E) = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\mu}(\zeta_i) \tag{5}$$

Finally, we define the occupancy measure $\boldsymbol{\psi}(\pi) \in \mathbb{R}^{|\mathcal{U}\times\mathcal{X}|}$ of a policy π, also referred to as state–action visitation frequency, as the expected discounted number of visits to any state–action pair (x,u) when following the policy π, given by

$$\boldsymbol{\psi}(\pi) = \mathbb{E}_{\zeta\sim\pi}\Big[\boldsymbol{\psi}(\zeta)\Big] \tag{6}$$

$$\boldsymbol{\psi}(\pi) = \mathbb{E}_{\zeta\sim\pi}\left[\sum_{t=1}^{T}\gamma^{t-1}\mathbf{1}_{(x_t=x\wedge u_t=u)}\right]\forall\,(x,u)\in\mathcal{X}\times\mathcal{U},$$

where $\boldsymbol{\psi}(\zeta) = \sum_{t=1}^{T}\gamma^{t-1}\mathbf{1}_{(x_t=u\wedge u_t=u)}$ corresponds to the occupancy measure along any path sampled from a policy π. Similarly, we define the expected empirical estimate of the expert's occupancy measure as

$$\bar{\boldsymbol{\psi}}(\pi^E) = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\psi}(\zeta). \tag{7}$$

Given these definitions, in this section we describe IRL algorithms whose optimization criteria are based on finding the objective function weights that maximize the match in feature expectations between the observed expert trajectories and optimal paths recovered from a candidate objective function. That is, a parameter vector $\hat{\boldsymbol{\theta}}$ that induces a policy $\hat{\pi}$ such that $\boldsymbol{\mu}(\hat{\pi}) \approx \bar{\boldsymbol{\mu}}(\pi^E)$.

5.1.1. Apprenticeship learning algorithms

First introduced by Abbeel and Ng (2004), the apprenticeship learning (AL) algorithms aim at finding a policy $\tilde{\pi}$ whose performance (or expected value) is close to the expert's policy π^E maximizing the unknown reward function $R^{\boldsymbol{\theta}}(s,a) = \boldsymbol{\theta}^T\boldsymbol{f}(s,a)$. It is important to notice that for these linear reward function approximations, feature expectations completely determine the expected cumulative reward for any policy (Abbeel & Ng, 2004). Hence, a match in feature expectations under 2 policies implies a match in the expected value of both policies. Formally, this relation is defined as

$$\begin{aligned}|V^{\hat{\pi}}(x) - V^{\pi^E}(x)| &= |\boldsymbol{\theta}^T\boldsymbol{\mu}(\hat{\pi}) - \boldsymbol{\theta}^T\boldsymbol{\mu}(\pi^E)|\;\forall\,x\in\mathcal{X}\\ &\le \|\boldsymbol{\theta}\|_2\|\boldsymbol{\mu}(\hat{\pi}) - \boldsymbol{\mu}(\pi^E)\|_2\\ &\le 1\cdot\epsilon = \epsilon\end{aligned} \tag{8}$$

where the first and second inequalities follow from the Cauchy–Schwarz inequality and $\|\boldsymbol{\theta}^*\|_2 \le \|\boldsymbol{\theta}^*\|_1 \le 1$.

From Eq. (8) it follows that a policy $\hat{\pi}$ that induces feature expectations $\boldsymbol{\mu}(\hat{\pi})$ close to $\boldsymbol{\mu}(\pi^E)$, *i.e.*, $\|\boldsymbol{\mu}(\tilde{\pi}) - \boldsymbol{\mu}(\pi^E)\| < \epsilon$ will result in a performance close to the expert's. To find this policy $\hat{\pi}$, AL algorithms solve the following optimization problem

$$\begin{aligned}\max_{p,\boldsymbol{\theta}}\quad & p\\ \text{s.t.}\quad & \boldsymbol{\theta}^T\boldsymbol{\mu}(\pi^E) \ge \boldsymbol{\theta}^T\boldsymbol{\mu}(\tilde{\pi}^{(j)}) + p,\; j = 0,\dots,i-1\\ & \|\boldsymbol{\theta}\|_2 \le 1.\end{aligned} \tag{9}$$

The solution to Eq. (9) is an objective function $J^{\boldsymbol{\theta}}(s,a) = \boldsymbol{\theta}^{(i)}\cdot\boldsymbol{\phi}(s,a)$ such that the expert's policy (as illustrated by the observed trajectories) does better, by a p margin, than any of the policies found so far. The solutions $\Pi = \{\hat{\pi}^{(0)},\dots,\hat{\pi}^{(j)}\}$ and $\Theta = \{\boldsymbol{\theta}^{(0)},\dots,\boldsymbol{\theta}^{(j)}\}$ are found through a three-step iterative process, as illustrated in Fig. 2: (1) find an optimal policy $\tilde{\pi}^{(i)}$ under the current objective function estimate $\boldsymbol{\theta}^{(i)}$, (2) compute feature expectations $\boldsymbol{\mu}(\hat{\pi}^{(i)})$ of the optimal policy obtained in the previous step, and (3) update the objective function estimate so as to reduce the difference in feature expectations between the optimal and expert's policies until convergence. Notice that steps (1) and (2) require access to a RL method that computes an optimal policy from a given objective function.

Although AL algorithms estimate an objective function as part of the optimization process, they do not necessarily recover the expert's underlying objective function correctly. These algorithms are only guaranteed to find an objective function that matches feature expectations and results in a policy whose performance is bounded by the expert's observed performance (Boularias & Chaib-draa, 2010).

Extensions of the AL base algorithms include the introduction of *linear programming* formulations so as to reduce the computational complexity associated with finding an optimal policy for each objective function estimate (Syed et al., 2008); bootstrapping methods to increase the accuracy of the expert's empirical feature expectations estimates when only a small part of the state space is covered in the observed demonstrations (Boularias & Chaib-draa, 2010); and the addition of game-theory ideas such that policies that are substantially better than the expert's policy can also be learned (Syed & Schapire, 2007). Similarly, AL algorithms have been also adapted to cases in which the environment state is partially observable (Choi & Kim, 2011a), the transition dynamics are unknown (Mori et al., 2011) or objective function and policy learning are done in an online manner (Jin et al., 2011).

5.1.2. Maximum margin planning

Although apprenticeship learning algorithms aim at finding an objective function that maximizes the similarity between the feature expectations underlying the optimal and expert's policies, this optimization criterion alone fails to provide a mechanism for explicitly matching the expert's behavior (Silver et al., 2010). To address this issue, the Maximum Margin Planning (MMP) algorithm proposed in Ratliff, Bradley, and Zinkevich (2006) learns an objective function for which a single deterministic and stationary policy with a guaranteed upper-bound (or margin) on the dissimilarity between the expert's and policy demonstrations can be obtained.

Starting from the same assumption of a linear objective function, MMP augments the optimization criterion based on the similarity between the expected value of the learned and expert policy with a loss function $l(\zeta,\hat{\zeta})$ that penalizes all state–action pairs for which the optimal path $\hat{\zeta}$ sampled from the learned policy $\hat{\pi}$ fails to match the observed expert's trajectory $\zeta\in\mathcal{D}$. Formally, the MMP algorithm aims to solve the following optimization problem

$$\begin{aligned}\min_{\boldsymbol{\theta},\beta_i}\quad & \lambda\|\boldsymbol{\theta}\|_2 + \sum_{i=1}^{N}\beta_i\\ \text{s.t.}\quad & \boldsymbol{\theta}^T\boldsymbol{\mu}(\zeta_i)\boldsymbol{\psi}(\zeta_i) + \beta_i \le \max_{\hat{\zeta}\sim\hat{\pi}}\boldsymbol{\theta}^T\boldsymbol{\mu}(\hat{\zeta})\boldsymbol{\psi}(\hat{\zeta}) + l(\zeta_i,\hat{\zeta}),\;\forall\,i = 1,\dots,N\end{aligned} \tag{10}$$

where N is the total number of demonstrated trajectories, $\boldsymbol{\psi}(\zeta)$ are the state–action visitation counts along a trajectory ζ, β_i is a slack variable that accounts for the error in the margin constraint for the $i-$ th trajectory, and λ balances the trade-off between regularization and meeting the constraints. The loss function $l(\zeta,\hat{\zeta})$ is proportional to the empirical visitation frequencies of each state–action pair so as to make highly visited state–action pairs take on larger reward values. By doing so, a preference over learned policies that frequently visit these states and thus closely mimic the observed behavior is induced (Ratliff, Bradley, & Zinkevich, 2006).

Improvements of the base MMP algorithm include the extension to non-linear objective functions through the adoption of a variant of the general ANYBOOST algorithm (Ratliff, Bradley, et al., 2006) or deep neural networks (Xia & El Kamel, 2016); and the introduction of model-free methods for the cases in which the transition dynamics are unknown (Xia & El Kamel, 2016).

5.1.3. Maximum entropy inverse reinforcement learning

Based on the observation that a policy π can be also interpreted as a distribution over the entire class of possible trajectories or paths, Ziebart, Maas, Bagnell, and Dey (2008) proposed to leverage the principle of maximum entropy (MaxEnt) so as to deal with the degeneracy of the IRL problem. Given that typically many different distributions of paths (*i.e.*, policies) can match the empirical feature expectations obtained from the expert's observed trajectories $\mathcal{D} = \{\zeta_1,\dots,\zeta_N\}$, the principle of maximum entropy resolves this ambiguity by choosing "the least committed" distribution, that is, the distribution (or policy) that does not exhibit any additional preferences beyond matching the expert's feature expectations. In addition to dealing with the inherent degeneracy of the IRL problem, the MaxEnt formulation also offers a principled way of accounting for potentially imperfect or sub-optimal behavior in the expert's demonstrations. Formally, the maximum entropy IRL algorithm aims at matching

$$\sum_{\zeta\sim\hat{\pi}} p(\zeta)\boldsymbol{\mu}(\zeta) = \bar{\boldsymbol{\mu}}(\pi^E), \tag{11}$$

where the empirical feature expectations $\bar{\boldsymbol{\mu}}(\pi^E)$ are computed according to Eq. (5) and $p(\zeta)$ corresponds to the distribution over paths induced by the optimal policy $\hat{\pi}$ learned from the parameterized objective function $J^{\boldsymbol{\theta}}(x,u)$. Thus, $p(\zeta|\boldsymbol{\theta})$, the probability of observing a trajectory ζ given the weights $\boldsymbol{\theta}$, is defined as

$$p(\zeta|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} q(\zeta) \exp(\boldsymbol{\theta}^T \boldsymbol{\mu}(\zeta)), \tag{12}$$

where $q(\zeta)$ is the (un-normalized) probability of any trajectory ζ to occur according to the system dynamics $\mathcal{T}$

$$q(\zeta) = p_0(x_1) \prod_{t=1}^{T} p(x_{t+1}|x_t, u_t), \tag{13}$$

and $Z(\boldsymbol{\theta}) = \sum_{\zeta' \sim \pi} q(\zeta') \exp(\boldsymbol{\theta}^T \boldsymbol{\mu}(\zeta'))$ is the normalization constant often referred to as the partition function. We note that according to Eq. (12) equally rewarded trajectories have equal probabilities, trajectories with higher rewards have the highest likelihood and the expert can still generate sub-optimal trajectories with a probability that decreases exponentially as the trajectories become less rewarded. In the case of deterministic dynamics, Eq. (12) reduces to $p(\zeta|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(\boldsymbol{\theta}^T \boldsymbol{\mu}(\zeta))$.

Learning "the least committed" distribution over paths can be formally defined as finding the objective function parameters $\boldsymbol{\theta}$ that maximize the casual entropy $\mathcal{H}(\cdot)$ of π, subject to the constraint of matching the observed feature expectations

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \mathcal{H}(\pi) \\ \text{s.t.} \quad & \boldsymbol{\theta}^T \boldsymbol{\mu}(\pi) = \boldsymbol{\theta}^T \bar{\boldsymbol{\mu}}(\pi^E), \\ & \sum_{\zeta \sim \pi} p(\zeta|\boldsymbol{\theta}) = 1, \\ & p(\zeta|\boldsymbol{\theta}) \geq 0 \; \forall \zeta \sim \pi. \end{aligned} \tag{14}$$

Ziebart, Maas, Bagnell, and Dey (2008) demonstrated that solving Eq. (14) corresponds to choosing the objective function parameters $\boldsymbol{\theta}$ that maximize the log-likelihood $L(\mathcal{D}|\boldsymbol{\theta})$ of the expert's trajectories $\mathcal{D}$ under the maximum entropy path distribution derived in Eq. (12)

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} L(\mathcal{D}|\boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \boldsymbol{\theta}^T \boldsymbol{\mu}(\pi^E) + \frac{1}{N} \sum_{\zeta \sim \mathcal{D}} \log q(\zeta) - \log Z(\boldsymbol{\theta}) \end{aligned} \tag{15}$$

This dual formulation can be solved using gradient-based optimization methods in which the gradient corresponds to the difference between the empirical $\bar{\boldsymbol{\mu}}(\pi^E)$ and expected feature expectations $\boldsymbol{\mu}(\hat{\pi})$ computed from the expert's demonstrations and learned optimal policy respectively. As with other IRL algorithms (*e.g.*, AL, MMP and Bayesian IRL), the computation of the partition function $Z(\boldsymbol{\theta})$ and feature expectations $\boldsymbol{\mu}(\hat{\pi})$ necessary to calculate both the likelihood and gradient require solving the forward problem for every candidate objective function (Ziebart, Maas, Dey, & Bagnell, 2008). Thus, as illustrated in Fig. 2, the MaxEnt IRL algorithm iterates between an inner loop in which the optimal policy for the current objective function $\hat{J}^{\boldsymbol{\theta}}(x,u)$ is learned, and an outer loop in which the likelihood of the expert's demonstrations is evaluated and later used to update the parameters of the objective function.

Boularias et al. (2011) showed that the MaxEnt IRL problem can be reformulated as the minimization of the relative entropy, as measured by the Kullback–Leibler divergence metric, between the target distribution $p(\zeta)$ induced by a policy that matches the empirical feature expectations of the demonstrations (*i.e.*, the expert's policy) and the reference or sampling distribution $q_{\pi_0}(\zeta) = q(\zeta) \prod_{t=1}^{|\zeta|-1} \pi_0(x_t, u_t)$ defined by the Markov decision process (MDP) dynamics $\mathcal{T}$ and a baseline policy π_0

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & D_{KL}(p \parallel q_{\pi_0}) \\ \text{s.t.} \quad & \boldsymbol{\theta}^T \boldsymbol{\mu}(\pi) = \boldsymbol{\theta}^T \bar{\boldsymbol{\mu}}(\pi^E), \\ & \sum_{\zeta \sim \pi} p(\zeta|\boldsymbol{\theta}) = 1, \\ & p(\zeta|\boldsymbol{\theta}) \geq 0 \; \forall \zeta \sim \pi. \end{aligned} \tag{16}$$

As in the original formulation, the feature matching constraint ensures that the optimal policy will follow the state visitation preferences observed in the expert's demonstrations and the minimization of the relative entropy between these 2 distributions solves the objective function ambiguity problem. In addition to these convergence guarantees, the reformulation of the MaxEnt IRL problem as a Relative Entropy IRL problem facilitates the adoption of sampling-based methods for both the computation of the feature expectations under the current objective function and the partition function $Z(\boldsymbol{\theta})$. Thus, it allows the extension of the gradient-based solution to cases in which non-linear function approximations of the objective function are preferred (Finn, Levine, & Abbeel, 2016) and the instances in which the environment dynamics are unknown or hard to specify (Boularias et al., 2011).

One of the main limitations of the MaxEnt IRL formulation initially proposed by Ziebart, Maas, Bagnell, and Dey (2008) is the need for an exact computation of the partition function $Z(\boldsymbol{\theta})$. Although this can be easily done in discrete environments for which a full knowledge of the dynamics of the system is available, it becomes computationally unfeasible for large, continuous spaces for which the dynamics of the system are likely unknown. Several extensions to the MaxEnt formulation have been proposed to address this issue and can be divided into two main groups: discretization and continuous approximations. Discretization-based approaches (Boularias et al., 2012; Byravan et al., 2015) propose to approximate the space of all possible trajectories induced by a policy through a coarse and sparse discrete graph representation. This choice of representation requires no adjustments of the discrete gradient-based algorithm proposed in Ziebart, Maas, Bagnell, and Dey (2008), makes the computation of the forward reinforcement learning problem and partition function tractable, and allows to account for potentially miss-specified or noisy features by leveraging and propagating information among adjacent states in the graph (Boularias et al., 2012). However, these approaches still require full knowledge of the state transition probability function and often call for a post-processing step such as trajectory optimization when used for generating trajectories in new environments or tasks. Furthermore, since they still employ the basic gradient-based algorithm proposed in Ziebart, Maas, Bagnell, and Dey (2008), these methods do not support stochastic state transition dynamics. To address this limitation, Herman et al. (2016) proposed a novel gradient-based algorithm that not only accounts for stochastic dynamics, but also allows to recover an objective function when this transition dynamics is unknown. The proposed algorithm also accounts for the possibility that the expert's belief about the transition dynamics might be differ from the real system dynamics. Specifically, the authors proposed a combined optimization problem which aims to maximize the likelihood of the expert demonstrations with respect to the reward function, the real transition dynamics and the expert's belief about these dynamics.

The approaches included in the continuous approximations group make use of simplifying assumptions that allow for closed-form solutions or approximate computations of the partition function, gradient and/or feature expectations. For instance, based on the assumption that the expert demonstrations are locally optimal, Levine and Koltun (2012) propose to use Laplace approximations around the expert's demonstrations to locally model the distribution over trajectories induced by the current objective function as a Gaussian distribution. This approximation preempts the computation of the forward reinforcement learning problem at each iteration and allows for an approximate computation of the log-likelihood. In Yin et al. (2016), the authors assume that the expert's unknown objective function can be accurately expressed as a local quadratic cost function for which a closed-form estimation of the partition function necessary for an efficient evaluation of the likelihood and associated gradient can be obtained on locally consistent demonstrations. As with the discretization approaches, these approximation methods also assume full knowledge about the state dynamics.

5.2. Trajectory – deterministic

An alternative to comparing objective function features is to compare the system states along the demonstrated trajectories. Given an estimate of the objective function, methods in this category solve the forward problem to generate a trajectory. The error between this generated trajectory and the demonstration trajectory is used as the metric that estimates the quality of the learned objective function.

5.2.1. Bi-level IOC

In the bi-level IOC approach (Mombaur et al., 2010), as illustrated in Fig. 2, the update step is implemented via an "upper-level" optimization, minimizing the error between the demonstration trajectory and a simulated trajectory. A nested "lower-level" optimization solves the forward problem to generate the simulated trajectory. Due to the nested forward problem, the upper-level problem is both non-linear and lacks an analytical gradient, thus requiring a derivative-free optimization method.

Formally, the upper-level is formulated as the minimization of the error between $\hat{\zeta}(x,u,\theta)$, the trajectory resulting from the minimization of the objective expected cumulative cost $V(x,u,\theta)$, and the demonstration trajectory ζ, where θ denotes the parameters of the objective function, *i.e.*, the basis function $\boldsymbol{\phi}(x,u)$ weights

$$\min_{\theta} \|\hat{\zeta}(x,u,\theta) - \zeta\|^2 \tag{17}$$

while the lower-level solves the forward problem to generate $\hat{\zeta}(x,u,\theta)$

$$\min_{\hat{\zeta}} V(x,u,\theta) := \sum_{t=0}^{T} \theta^T \boldsymbol{\phi}(x_t,u_t) \tag{18}$$

$$\text{s.t.} \quad \dot{x} = f(x_t,u_t), \quad t = 0,\dots,T \tag{19}$$

where $f_j(x,u)$ is the deterministic and continuous time version of the dynamic function first introduced in Eq. (1). Eq. (17) can then be solved to obtain the objective function (Mombaur et al., 2010).

Alternative formulations include minimizing trajectory error while calculating the weights using Pontryagin's maximum principal (Jin, Wang, et al., 2019) or linear quadratic regulator (LQR) (El-Hussieny et al., 2016; El-Hussieny & Ryu, 2019; Unni et al., 2017) as an optimization framework.

5.2.2. One-level IOC

Instead of the bi-level approach, Hatz et al. (2012) proposed replacing the lower-level direct problem with optimality conditions, to combine the two stages into a single step. This formulation consists of utilizing Lagrangian multipliers (Terekhov et al., 2010) or Karush–Kuhn–Tucker (KKT) conditions (Albrecht et al., 2012; Hatz et al., 2012) to solve the lower-level forward problem, allowing the two levels to be combined into a single level optimization problem. Given the Lagrangian $\mathcal{L} := V + \lambda_{\text{eq}}^T \boldsymbol{r}_{\text{eq}} + \lambda_{\text{ineq}}^T \boldsymbol{r}_{\text{ineq}}$ where λ_{eq} and λ_{ineq} denote the Lagrangian multipliers associated to the equality $\boldsymbol{r}_{\text{eq}}$ and inequality $\boldsymbol{r}_{\text{ineq}}$ constraints respectively, the optimization problem is formulated as follows:

$$\min_{(x,u,\theta,\lambda_{\text{eq}},\lambda_{\text{ineq}})} \quad \sum_{t=0}^{T-1} (\hat{\zeta}(x_t,u_t,\theta) - \zeta_t)^2 \tag{20}$$

$$\begin{aligned}
\text{s.t.} \quad & \dot{x} = f(x_t,u_t),\\
& 0 = \boldsymbol{r}_{\text{eq}}(x_t,u_t),\\
& 0 = \boldsymbol{r}_{\text{ineq}}(x_t,u_t),\\
& 0 = \nabla_{(x,u,q)} \mathcal{L}(x_t,u_t,\theta,\lambda_{\text{eq}},\lambda_{\text{ineq}}),\\
& 0 \le \lambda_{\text{ineq}},\\
& 0 = \lambda_{\text{ineq}}^T \boldsymbol{r}_{\text{ineq}}(x_t,u_t), \quad t = 0,\dots,T
\end{aligned}$$

where ζ_t indicates the observed state-control pair at time t in the demonstration trajectory ζ. The one-level method's main advantage is that the problem can be formulated into a single stage, reducing the problem complexity during implementation.

5.3. Trajectory – ranking

Brown et al. (2019) address a variant of the inverse reinforcement learning problem in which (1) demonstrations are assumed to be noisy and sub-optimal, (2) demonstration trajectories only include state observations, and (3) demonstration trajectories are qualitatively ranked by the demonstrators (*i.e.*, ζ_k for $k = 1,\dots,m$ where $\zeta_i \prec \zeta_j$ if $i < j$). To solve this problem, the authors aim at finding a parameterized objective function $\hat{J}^{\theta}(x)$ that preserves the ranking among demonstrations (*i.e.*, $\sum_{x\in\zeta_i} \hat{J}^{\theta}(x) < \sum_{x\in\zeta_j} \hat{J}^{\theta}(x)$ when $\zeta_i \prec \zeta_j$). Formally, ranking error IRL aims to solve the following optimization problem

$$\min_{\theta} \sum_{\zeta_i<\zeta_j} \log \frac{\exp \sum_{x\in\zeta_i} \hat{J}^{\theta}(x)}{\exp \sum_{x\in\zeta_i} \hat{J}^{\theta}(x) + \exp \sum_{x\in\zeta_j} \hat{J}^{\theta}(x)}. \tag{21}$$

Intuitively, this optimization problem trains a classifier that can predict whether one trajectory is preferable to another one based on the expected returns of each trajectory as predicted by the learned reward function $\hat{J}^{\theta}(x)$. In practice, Brown et al. (2019) use a neural network to approximate $\hat{J}^{\theta}(x)$ and split the observed demonstrations into partial trajectories pairs so as to increase the number of training samples. Furthermore, the authors have shown that policies learned from the reward functions recovered from ranked demonstrations can results in better performance than the demonstrated one.

The idea behind Structured Classification and Cascaded Supervised IRL (SCIRL and CSIRL) algorithms (Klein et al., 2012, 2013) is that although defining an operator that goes from expert demonstrations to a objective function is a hard problem, it is possible to define a two-step process in which one operator goes from demonstrations to a score function, *i.e.*, $\mathcal{D} \rightarrowtail Q(x,u,\theta)$, and a second operator goes from the score function to the corresponding objective function, *i.e.*, $Q(x,u,\theta) \rightarrowtail J^{\theta}(x,u)$. This two-step process is motivated by the fact that the optimal Bellman equation

$$Q^*(x,u,\theta) = J^{\theta}(x,u) + \gamma \sum_{x'\in\mathcal{X}} p(x'|x,u) \max_{b\in U} Q^*(x',b,\theta) \tag{22}$$

imposes a one-to-one relation between the optimal-action value function (*i.e.*, optimal score function) and the objective function $J^{\theta}(x,u)$. Thus, if the score function is known or learned from demonstrations, it can be later used to recover the reward function associated to it.

Klein et al. (2012) propose to approximate the first operator using a score function-based multi-class classifier that rates the association of a given action u as the class labels, with the input state x. We notice that according to this definition, a good classifier, and implicitly a good score function, is one that matches the expert action choices as closely as possible. Once the score function is learned, under the assumption of known transition dynamics the second operator can be omitted and the objective function $\hat{J}^{\theta}(x,u)$ is readily obtained by inverting Eq. (22). In a more general setting where the transition dynamics are unknown, the objective function $\hat{J}^{\theta}(x,u)$ can be approximated using a standard ML regression algorithm (Klein et al., 2013). Munzer et al. (2015) show an application of CSIRL to the context of IRL problem in which relational MDPs *i.e.*, a generalization of the standard MDP to high level logical representations are used.

5.4. Trajectory – stochastic

A number of papers consider trajectory comparison within a probabilistic setting. Given an assumed objective function structure, most commonly in the form of Eq. (3), and an assumed conditional probability distribution for observing a trajectory given the objective function, these methods estimate the objective function weights by maximizing either the posterior probability distribution over the space of objective functions (Section 5.4.1), or the probability of observing the demonstrated trajectories (Section 5.4.2). A few methods also consider a comparison between the distributions of the observed and generated trajectories (Section 5.4.3).

5.4.1. Bayesian inverse reinforcement learning

Bayesian approaches estimate a probability distribution over the objective function given the observed demonstrations. Ramachandran (2007) proposed the first Bayesian formulation for IRL. Similar to the MaxEnt approach, they model the probability of observing a trajectory ζ given a particular objective function J^θ via an exponential distribution:

$$p(\zeta|J^\theta) = \frac{1}{Z}\exp \alpha \mathbb{E}(\zeta, J^\theta) \tag{23}$$

where α is a parameter representing the confidence in the demonstrator's expertise; $\mathbb{E}(\zeta, J^\theta) = \sum_i V(x_i, \theta)$, with $\hat{\pi}$ denotes the optimal policy with respect to the objective function J^θ and Z is a normalizing constant. Given this model, the posterior probability of J^θ can be computed by applying Bayes theorem

$$p(J^\theta|\zeta) = \frac{p(\zeta|J^\theta)p(J^\theta)}{p(\zeta)} = \frac{1}{Z'}\exp \alpha \mathbb{E}(\zeta, J^\theta)p(J^\theta)$$

such that $p(J^\theta)$ captures any prior knowledge about the reward.

Given the posteriori distribution $p(J^\theta|\zeta)$, different point estimates can be interpreted to optimize different similarity criteria. The posteriori mean minimizes the least squared loss function between the actual and estimated objective function, while the median minimizes the linear loss function. The optimal policy corresponding to the mean objective function is also shown to minimize the policy loss function (*i.e.*, the difference between the optimal and estimated value functions). Therefore, depending on which posteriori point estimate is used, the Bayesian IRL (BIRL) framework either uses objective function or value function comparison for the similarity estimate.

In the original formulation from Ramachandran (2007), the state space is discretized and the objective function is modeled as a look up table over states. Rothkopf and Dimitrakakis (2011) generalize this approach to admit other policy and objective function structure assumptions and develop additional algorithms for estimation given the generalized framework.

Choi and Kim (2011b) argue that using the maximum a-posteriori (MAP) as the estimate of the objective function yields better performance than using the posterior mean. This is because the mean integrates over the entire objective function space, including those objective functions inconsistent with the behavior data. They also demonstrate that previous IRL algorithms can be formulated as MAP-BIRL by considering that all algorithms optimize a two part objective function: an assessment term evaluating the compatibility of the reward function with the behavior data and a regularization term specifying a preference about the reward function. This optimization can be converted into the Bayesian framework by encoding the regularization term into the prior and the compatibility term into the likelihood.

5.4.2. Maximum likelihood

In this class of approaches, given a model of the likelihood of observing a given trajectory given an objective function, the likelihood is directly optimized. For example, Kalakrishnan et al. (2013) use the PI^2 algorithm (Theodorou et al., 2010) as the basis for IRL. They assume that the state-dependent cost function is linearly parameterized (as in Eq. (3)), and the weight vector to be learned is the concatenation of the state-dependent basis function weights, the control cost scaling (assuming the shape of the quadratic control cost is known) and the terminal cost scaling. Given an exponential probability of observing a given trajectory conditioned on the reward (as in Eq. (23)), the weights are found by minimizing the negative log of the probability of observing the demonstrated trajectories, with an added L-1 norm regularization term over the weights, using a quasi-Newton optimization approach. The proposed approach is demonstrated for learning inverse kinematics and optimal motion trajectories for reaching with a 7 DoF arm. Mainprice and Berenson (2014) apply the path integral IRL algorithm (Kalakrishnan et al., 2013) to recover the objective function of segmented human–human collaborative motions.

Doerr et al. (2015) reformulate inverse optimal control as a reinforcement learning policy search that uses similarity to demonstrated behavior as the reward. Additional objectives can be added to incorporate additional robot or context objectives. The objective function is formulated as a weighted sum of features, parameterized by a weight vector. Given this formulation, the policy search is implemented using a black box Covariance Matrix Adaptation optimiser REIMO. The proposed approach is demonstrated on kinesthetically taught robot arm trajectories.

5.4.3. Comparison of distributions

Aware of how critical the choice of sampling distribution is when using sampled trajectories to estimate the objective function, Finn, Levine, and Abbeel (2016) and Fu et al. (2017) proposed to exploit deep policy optimization and generative adversarial networks to simultaneously learn the sampling distribution that best matches the maximum entropy trajectory distribution with respect to the current reward function parameters and the reward parameters themselves. By doing so, the proposed methods can simultaneously learn the expert's objective function and policy.

Guided Cost Learning (GCL) (Finn, Levine, & Abbeel, 2016) and Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016) present efficient algorithms to learn from high-dimensional continuous inputs without knowledge of the dynamics and hand-crafted features. They have a very similar algorithmic structure which consists of matching the distribution of the expert trajectories. To do so, they simultaneously learn the reward and the policy that imitates the expert demonstrations. At each step, sampled trajectories of the current policy and the expert policy are used to produce a reward function. Then, this reward is (partially) optimized to produce an updated policy and so on. In GAIL, the reward is obtained from a network trained to discriminate between expert trajectories and (partial) trajectories sampled from a generator (the policy). In GCL, the reward is obtained by minimization of the Maximum Entropy IRL cost.

5.5. Value function estimation

In this class of approaches, the problem is reformulated to estimate the value function, from which the objective function is subsequently obtained. Dvijotham and Todorov (2010) propose a reward learning approach that parameterizes and infers the *value* function instead of the cost function. The cost function can then be computed using an explicit formula. They derive formulations for both discrete and continuous domain problems, and show that the resulting problem is convex and more easily solveable than when the estimation is done on the cost function.

Li and Burdick (2017a) reformulate the IRL problem by approximating the *VR function*, which represents the summation of the reward function and the discounted optimal value function, $c(s) = J(s)+\gamma V^*(s)$. The objective function is then estimated from the VR function by using the Bellman optimality equation – this ensures that the resulting value and reward functions always meet the Bellman optimality criteria. The VR function is modeled as a neural network or a Gaussian Process, and solved by assuming a motion model $p(a|s)$ proposed in Ramachandran (2007), and maximizing the likelihood of observing the training trajectories. The proposed approach is demonstrated both in simulations and with a human motion dataset, where the objective is to estimate the preferences during patient movement from the discretized center-of-pressure (CoP) trajectory.

Li and Burdick (2017b) describe an IRL approach that considers demonstrations for multiple tasks. They hypothesize that the demonstrator may have similar preferences for the tasks, where the underlying reward function is only slightly modified from an *innate* reward function. They adapt their previous method (Li & Burdick, 2017a) to add a reward sharing loss (*i.e.*, penalizing the task reward difference to the innate reward). The proposed approach is demonstrated in simulation and with a human motion dataset, where the objective is to estimate the preferences during patient movement from the discretized CoP trajectory.

5.6. Controller

A small number of papers approach the IOC/IRL problem by minimizing the error with respect to the gains. Priess et al. (2014) and Menner and Zeilinger (2018) employ a LQR framework. The optimal gain K_e is either known, or estimated using least squares from the observation data $y = A - B * K$, assuming known system dynamics matrix A and B. They then estimate the state Q and controller R matrix via gradient descent to minimize the Frobenius norm of $K - K_e$.

Under the assumption that reward functions are paramterizations of a policy class, Neu and Szepesvari (2012) propose a novel gradient algorithm that learns a reward function such that the resulting optimal policy matches closely an expert's observed trajectories. To do so, the authors combine ideas from supervised learning and apprenticeship learning (Abbeel & Ng, 2004). Specifically, the proposed algorithm seeks to minimize an optimization criterion that penalizes deviations from the expert's policy (*i.e.*, supervised learning). The policy against which the expert's policy is compared is obtained by tuning a reward function and learning the optimal policy with respect to this function (*i.e.*, AL). Formally, the proposed gradient algorithm aims to solve the following optimization problem

$$\begin{aligned} \min_{\theta} \quad & \sum_{x\in\mathcal{X},u\in\mathcal{U}} \bar{\psi}^E(x)(\hat{\pi}(x,u) - \bar{\pi}^E(x,u))^2 \\ \text{s.t.} \quad & \hat{\pi}(x,u) = G(Q(x,u,\theta))\ \forall (x,u) \in \mathcal{X}\times\mathcal{U}, \end{aligned} \tag{24}$$

where $\bar{\psi}^E(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\sum_{x_t\in\zeta_i}\mathbf{1}_{x_t=x}$ is the empirical occupation frequency of state x under the expert's policy,

$$\bar{\pi}^E(u|x) = \frac{\sum_{i=1}^{N}\sum_{(x_t,u_t)\in\zeta_i}\mathbf{1}_{x_t=x,u_t=u}}{\sum_{i=1}^{N}\sum_{x_t\in\zeta_i}\mathbf{1}_{x_t=x}} \tag{25}$$

corresponds to the empirical estimate of expert's policy, $Q(x,u,\theta)$ is the optimal action-value function of the optimal policy $\hat{\pi}$ learned from the parameterized objective function $J^{\theta}(x,u)$, and G is a suitable smooth mapping that returns a greedy policy with respect to its argument. Since the mapping from the space of reward parameters θ to action-value functions $Q(x,u,\theta)$ is non-smooth and the primary objective of this optimization problem to find the policy that is the closest to the expert's policy, the authors proposed to use sub-differentials and natural gradients when solving Eq. (24). While the former allows to approximate the gradient of the non-differentiable reward to action-value function mapping, the latter determines the gradient direction in each step such that $\hat{\pi}$ moves in the steepest descent direction.

5.7. Optimality violations

The most indirect set of IOC/IRL methodologies do not assess similarity directly, but instead minimize the residual or optimality violations. Since the input trajectory is assumed to be optimally generated from a dynamic system and a set of cost functions, the generated trajectory should be optimal with respect to the cost function, and therefore satisfy optimality criteria. However, practical factors like noise in the trajectory or uncertainty in the cost function formulation lead to deviations from the optimal trajectory. In this class of approaches, these optimality violations are minimized to recover the objective function.

These methods are computationally fast because they do not require the trajectory or features to be computed, *i.e.*, they avoid the need for solving the forward problem. However, the majority of these methods require computing the cost function gradient along the trajectory, thus requiring higher order derivatives of the system dynamics and features to be available. The minimization of the gradient may not also lead to a minimization of the cost function value (Betts, 1998).

5.7.1. Karush-Kuhn–Tucker conditions

A common approach is to rely on the KKT conditions (Boyd & Vandenberghe, 2004), which specify that the gradient of the objective function should be zero along the optimal trajectory. To calculate the cost weights, the KKT equations can be re-formulated into a linear residual matrix and minimized. Given an objective function modeled as a weighted sum of basis cost functions $\boldsymbol{\phi}(\mathbf{x})$ to be minimized with respect to some given equality $\boldsymbol{r}_{\text{eq}}$ (inequality terms excluded for brevity):

$$\begin{aligned} &\min_{\mathbf{x}^*} V(\mathbf{x}^*) = \boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}^*) \\ &\text{s.t.} \quad \boldsymbol{r}_{\text{eq}}(\mathbf{x}^*) = 0 \end{aligned} \tag{26}$$

the KKT Lagrangian $L(\mathbf{x})$ and gradient $\nabla_{\mathbf{x}}L(\mathbf{x})$ are defined as:

$$L(\mathbf{x}^*) = \boldsymbol{\theta}^T\boldsymbol{\phi}(\mathbf{x}^*) + \lambda_{\text{eq}}\boldsymbol{r}_{\text{eq}}(\mathbf{x}^*) = 0 \tag{27}$$

$$\nabla_{\mathbf{x}}L(\mathbf{x}^*) = \boldsymbol{\theta}^T\nabla_{\mathbf{x}}\boldsymbol{\phi}(\mathbf{x}^*) + \lambda_{\text{eq}}\nabla_{\mathbf{x}}\boldsymbol{r}_{\text{eq}}(\mathbf{x}^*) = 0 \tag{28}$$

where the partial differential of the gradient $\nabla_{\mathbf{x}}$ is calculated with respect to the state $\mathbf{x}$, $\lambda_{\mathbf{eq}}$ are the Lagrangian multipliers on $r_{\text{eq}}(\mathbf{x}^*)$. The condition that must be met to ensure optimality is:

$$\nabla_{\mathbf{x}}L(\mathbf{x}^*) = 0 \tag{29}$$

If it is assumed that the system is not strictly optimal, but rather only approximately optimal (Keshavarz et al., 2011), then Eq. (30) is minimized but is not strictly zero:

$$\begin{aligned} &\min_{\hat{\theta},\lambda} \nabla_{\mathbf{x}}L(\mathbf{x}) \\ &\in \hat{\theta} \geq 0 \end{aligned} \tag{30}$$

Since the KKT equations are linear with respect to the unknown variables $\hat{\theta}$ and λ_{eq}, Eq. (30) can be written as a least square problem and solved computationally efficiently (Puydupin-Jamin et al., 2012).

While Puydupin-Jamin et al. (2012) hand-selected a significant basis cost function to prevent the zero weight ($\theta = 0$) trivial solution, Panchea (2015) employed a basis function pivot to estimate the significant basis function, while (Englert et al., 2017) add a regularizing constraint to force $\|\hat{\theta}\|_1 = 1$. Key improvements to the inverse KKT method include checks to ensure that the residual matrix is well formed (Panchea et al., 2017), and factoring out $\hat{\lambda}$ terms using the Hessian (Englert et al., 2017).

5.7.2. Other optimality conditions

Other optimality conditions that have been used in the literature include the Euler–Lagrange equation (Aghasadeghi & Bretl, 2014; Terekhov & Zatsiorsky, 2011), the Pontryagin's maximum principle (Johnson et al., 2013; Molloy et al., 2018; Zhang et al., 2019), as well as the Hamilton–Jacobi–Bellman (Li et al., 2011; Menner et al., 2019; Moylan & Anderson, 1973).

5.8. Representative papers summary

Table 2 summarizes the surveyed papers. This table highlights each paper's assumption on the form of the objective function (Section 4), similarity measure (Section 5), modeling assumptions (Section 6), and the optimality of the demonstration trajectories (Section 7). Each row depicts a unique combination of the columns, and summarizes the citations corresponding to this combination.

From Table 2, we note that the large majority of the surveyed papers use an objective function that is parametric. Non-parametric modeling and unstructured machine-learning based methods are a small but growing segment of IRL methods. Additional analysis into this aspect can be found in Section 4.

Table 2 also show significant overlap between IOC and IRL approaches. For example, nearly all of the literature utilizes explicitly defined system models, emphasizing that model-free approaches that

Table 2
An overview of representative papers, grouped by unique combination of algorithm compositions denoted by the framework proposed in this survey. Each row denote a representative paper (rep) and number of papers that contains this combination. The criteria used align with the various sections in this manuscript. Color highlighting is for ease of reading.

Rep	Papers	Similarity Measure	Model	State	Controller	Obj. Function	Trajectory
C1	2	Feature Expectation	Known	Continuous	Deterministic	Linear	Optimal
C2	4	Feature Expectation	Known	Continuous	Stochastic	Linear	Noisy
C3	1	Feature Expectation	Known	Continuous	Stochastic	Non-linear	Noisy
C4	12	Feature Expectation	Known	Discrete	Deterministic	Linear	Optimal
C5	1	Feature Expectation	Known	Discrete	Deterministic	Non-linear	Optimal
C6	1	Feature Expectation	Known	Discrete	Deterministic	Non-linear	Suboptimal
C7	1	Feature Expectation	Known	Discrete	Deterministic	Non-parametric	Optimal
C8	9	Feature Expectation	Known	Discrete	Stochastic	Linear	Noisy
C9	1	Feature Expectation	Known	Discrete	Stochastic	Linear	Suboptimal
C10	1	Feature Expectation	Known	Discrete	Stochastic	Non-linear	Noisy
C11	3	Feature Expectation	Known	Discrete	Stochastic	Non-parametric	Noisy
C12	1	Feature Expectation	Unknown	Continuous	Deterministic	Linear	Optimal
C13	5	Feature Expectation	Unknown	Continuous	Stochastic	Non-parametric	Noisy
C14	1	Feature Expectation	Unknown	Discrete	Deterministic	Linear	Suboptimal
C15	2	Feature Expectation	Unknown	Discrete	Stochastic	Linear	Noisy
C16	1	Feature Expectation	Unknown	Discrete	Stochastic	Linear	Optimal
C17	1	Feature Expectation	Unknown	Discrete	Stochastic	Non-parametric	Suboptimal
C18	20	Trajectory Deterministic	Known	Continuous	Deterministic	Linear	Optimal
C19	1	Trajectory Ranking	Unknown	Continuous	Stochastic	Non-parametric	Suboptimal
C20	1	Trajectory Stochastic	Known	Continuous	Deterministic	Linear	Noisy
C21	1	Trajectory Stochastic	Known	Continuous	Deterministic	Linear	Optimal
C22	1	Trajectory Stochastic	Known	Continuous	Deterministic	Non-parametric	Noisy
C23	3	Trajectory Stochastic	Known	Continuous	Deterministic	Non-parametric	Optimal
C24	3	Trajectory Stochastic	Known	Continuous	Stochastic	Linear	Optimal
C25	1	Trajectory Stochastic	Known	Continuous	Stochastic	Non-linear	Optimal
C26	1	Trajectory Stochastic	Known	Continuous	Stochastic	Non-parametric	Noisy
C27	9	Trajectory Stochastic	Known	Discrete	Deterministic	Linear	Noisy
C28	1	Trajectory Stochastic	Known	Discrete	Deterministic	Linear	Suboptimal
C29	1	Trajectory Stochastic	Known	Discrete	Deterministic	Non-linear	Noisy
C30	2	Trajectory Stochastic	Known	Discrete	Deterministic	Non-parametric	Noisy
C31	1	Trajectory Stochastic	Unknown	Discrete	Deterministic	Linear	Noisy
C32	2	Controller	Known	Continuous	Deterministic	Linear	Optimal
C33	3	Controller	Known	Continuous	Stochastic	Non-parametric	Optimal
C34	1	Controller	Known	Discrete	Deterministic	Linear	Optimal
C35	16	Optimality	Known	Continuous	Deterministic	Linear	Optimal
C36	2	Optimality	Known	Continuous	Deterministic	Linear	Optimal
C37	1	Optimality	Known	Continuous	Stochastic	Linear	Optimal

C1 Abbeel et al. (2008), Coates et al. (2009) C2 Aghasadeghi and Bretl (2011), Byravan et al. (2015), Park and Levine (2013), Yin et al. (2016) C3 Levine and Koltun (2012) C4 Abbeel and Ng (2004), Choi and Kim (2011a), Ng and Russell (2000), Ratliff, Bradley, and Zinkevich (2006), Syed et al. (2008) and others C5 Ratliff, Bradley, et al. (2006) C6 Silver et al. (2010) C7 Chen et al. (2010) C8 Henry et al. (2010), Kitani et al. (2012), Ziebart, Maas, Bagnell, and Dey (2008), Ziebart, Maas, Dey, and Bagnell (2008), Ziebart et al. (2009) and others C9 Shiarlis et al. (2016) C10 Levine et al. (2010) C11 Boularias et al. (2012), Levine et al. (2011), Wulfmeier et al. (2015) C12 Mori et al. (2011) C13 Finn, Levine, and Abbeel (2016), Fu et al. (2017), Jin, Petrich, et al. (2019), Jin, Petrich, Zhang, et al. (2020), Sermanet et al. (2016) C14 Muelling et al. (2014) C15 Boularias et al. (2011), Herman et al. (2016) C16 Boularias and Chaib-draa (2010) C17 Xia and El Kamel (2016) C18 Berret et al. (2011), Hatz (2014), Liu et al. (2005), Mombaur et al. (2010), Terekhov et al. (2010) and others C19 Brown et al. (2019) C20 Okal and Arras (2016) C21 Doerr et al. (2015) C22 Michini et al. (2013) C23 Joukov and Kulic (2017), Li and Burdick (2017a, 2017b) C24 Kalakrishnan et al. (2013, 2010), Mainprice and Berenson (2014) C25 Dvijotham and Todorov (2010) C26 Chen et al. (2020) C27 Babes-Vroman et al. (2011), Choi and Kim (2012), Dimitrakakis and Rothkopf (2011), Kim and Pineau (2016), Ramachandran (2007) and others C28 Melo et al. (2007) C29 Choi and Kim (2013) C30 Michini and How (2012), Qiao and Beling (2011) C31 Rothkopf and Ballard (2013) C32 Menner and Zeilinger (2018), Priess et al. (2014) C33 Klein et al. (2012, 2013), Munzer et al. (2015) C34 Neu and Szepesvari (2012) C35 Englert et al. (2017), Johnson et al. (2013), Papadopoulos et al. (2016), Puydupin-Jamin et al. (2012), Terekhov and Zatsiorsky (2011) and others C36 Majumdar et al. (2017), Zhang et al. (2019) C37 Li et al. (2011).

are gaining traction in current IRL research are a small component of the IRL literature. IRL papers also tend to use discrete state spaces, while IOC papers tend to utilize continuous state space modeling. Additional analysis into these aspects can be found in Section 6.

A core assumption of many methods is that the input trajectory is optimal, or optimal and corrupted with random noise. Fewer works have considered sub-optimal trajectories, or even trajectories that failed to achieve the task, as will be discussed in Section 7.

6. System and environment modeling

Approaches to objective learning differ in how they model the body and environment of the demonstrator, and what knowledge of the model is available during objective inference. Most early approaches (Abbeel & Ng, 2004) assume that the system model is fully known, while more recent approaches move towards inferring the objective when the model is unknown.

6.1. Discrete vs. continuous space models

Early IRL methods (Ng & Russell, 2000) were mostly demonstrated on grid problems with discrete states and actions, exhibiting less than a hundred states and four actions. Having a finite state and action space is the easiest scenario (Zhifei & Er, 2012) and even allows for online testing (Jin et al., 2011).

To deploy IRL approaches in continuous space, it requires either discretizing the space (*e.g.*, Byravan et al., 2015) or using an approximation function such as a neural network (*e.g.*, Finn, Levine, & Abbeel, 2016). Aghasadeghi and Bretl (2011) and Kalakrishnan et al. (2010) extend the MaxEnt IRL formulation to continuous-time stochastic systems

with continuous state and action spaces by replacing feature counts in the objective function with a path integral formulation based on continuous states. Similarly, Kretzschmar et al. (2016) applies MaxEnt IRL to learn the probability distribution over navigation trajectories of interacting pedestrians using a subset of their continuous space trajectories. A mixture distribution models both the discrete and continuous navigation decisions. Levine and Koltun (2012) utilizes MaxEnt IRL in high dimensional continuous domains by using a local approximation to the reward function likelihood. Coates et al. (2009) utilizes differential dynamic programming that approximately solves continuous state–space MDP. This is done by iteratively approximating it as an LQR control problem.

In most IOC works, the state and action spaces are continuous, either using continuous-time (*e.g.*, Clever & Mombaur, 2017; Johnson et al., 2013) or discrete-time (*e.g.* Byravan et al., 2015; Englert & Toussaint, 2018; Molloy et al., 2018) representations.

6.2. Human body models

For demonstrations of articulated body movement, most papers assume that there is some knowledge of the kinematics/dynamics of the limb or the whole body either in 2D or 3D. These models can be linear or nonlinear. Linear models for studying human motion are used to simplify the problem formulation and are generally formulated for specific tasks, such as reach to grasp behavior (El-Hussieny et al., 2016), seated balancing system (Priess et al., 2014), left shoulder flexion to study neuromuscular disorders (Unni et al., 2017), and gaze movements (El-Hussieny & Ryu, 2019). On the other hand, nonlinear models are used to represent the task with more details. Examples include squat motions (Jin, Kulić, et al., 2019; Lin et al., 2016), human arm movement (Albrecht et al., 2012; Berret et al., 2011; Carreno-Medrano et al., 2019; Li et al., 2011; Menner et al., 2019; Oguz et al., 2018; Panchea et al., 2018; Sylla et al., 2014), human locomotion (Aghasadeghi & Bretl, 2014; Clever et al., 2018; Clever & Mombaur, 2017; Clever et al., 2016; Liu et al., 2005; Mombaur & Clever, 2017; Mombaur et al., 2010), and human running (Hatz, 2014; Liu et al., 2005; Mombaur & Clever, 2017; Mombaur et al., 2013; Papadopoulos et al., 2016; Puydupin-Jamin et al., 2012).

Clever and Mombaur (2017) highlights this question: What is a good mechanical model that is able to reproduce the essential characteristics of the motions under investigation? Rebula et al. (2019) suggests that models should be chosen based on the experimental protocol and hypothesis under consideration. Common nonlinear models are: (1) torque-driven models (robot models), (2) musculoskeletal models (biomechanical inspired models). Torque driven models consider joint torques as control inputs, and joint angles and velocities as outputs. Examples include Albrecht et al. (2012), Carreno-Medrano et al. (2019), Jin, Kulić, et al. (2019), Li et al. (2011), Lin et al. (2016), Oguz et al. (2018), Panchea et al. (2018), Sylla et al. (2014). Musculo-skeletal models consider muscle activation as control inputs. Berret et al. (2011) models the musculoskeletal arm dynamics in the sagittal plane by adding the actuator dynamics (*i.e.*, set as the acceleration of torques equals to the neural input to muscles) to the torque-driven model. Albrecht et al. (2012) models the human arm by presenting the joint torques as a combination of torques generated by the muscle forces and the moment arms, the torques resulting from passive properties of the human arm with joint damping and the torques induced on the arm by external forces and the Jacobian of the hand position. In this model the muscle behavior is considered as the second-order low-pass filter.

Berret et al. (2011) observe that a more complex model (*i.e.*, modeling agonist/antagonist muscles as second order low-pass filters) does not improve the prediction results for the cost functions drastically. Therefore, the choice of the model depends on the re-targetting objective. Also, there might me some properties that are difficult to estimate. For example, Oguz et al. (2018) models the musculoskeletal system for human arm reaching motion with torque-driven model. It is mentioned that viscous frictions and elastic properties of the tissues are difficult to estimate, so they are neglected in the dynamics.

Focusing on gait, Clever and Mombaur (2017) classifies walking models into two classes. First, template models, which represent some major characteristics of human gait, and second, full body models, which describe motion at the joint level, with kinematic and dynamic properties that are close to a real human body. The clear advantage of full body models lies in their anthropomorphic kinematics and dynamics. However, even though considering a template model does not give insight into human behavior at the individual joint level, those models can reveal characteristic behavior of human gait. Furthermore, template models can be used for human gait analysis and humanoid gait generation. Clever et al. (2018) explains that making use of template models for the identification of optimality criteria is an interesting approach for robotic applications for the following reasons: (1) the same model can be used for different walking scenarios; (2) the same model with different parameters can be used for human gait analysis and humanoid gait generation; (3) a sequence of several steps can be considered; (4) computational results are directly usable for robot controllers if they are based on the same template model; (5) it has potential to be used for robot control in real time. The walking model in Rebula et al. (2019) is based on a ballistic walker that contains key aspects of gait, such as a heavy swinging leg, ground impacts, and torso balancing. This model omits many aspects of human locomotion, such as muscles, ligaments, and detailed anatomical joints. However, the goal is to capture the role of the major joints involved in walking, which are often analyzed in terms of overall rotational motion and simple torques. Papadopoulos et al. (2016) considers a high-level kinematic model perspective for human path planning. So the walking human can be modeled with the unicycle kinematic model and the complex activities performed during walking by muscles and brain in commanding and coordinating many elementary motor acts can be neglected.

6.3. Unknown dynamics

To relax the assumption of known dynamics, one approach is to use the demonstrated trajectories to both infer the objective and identify the system model (Abbeel et al., 2010). A second approach is to simultaneously learn the expert's objective function and policy and thus learn the optimal policy directly.

Sample-based methods use trajectories sampled from the optimal policy (Aghasadeghi & Bretl, 2011) or a reference distribution (Boularias et al., 2011; Finn, Levine, & Abbeel, 2016; Fu et al., 2017) to solve the IRL problem when the model of the system dynamics is unknown. In Aghasadeghi and Bretl (2011), sampled trajectories are used to estimate the parameters of a close form maximum entropy distribution over trajectories as well as the reward function parameters. Boularias et al. (2011) uses model-free reinforcement learning methods and importance sampling to approximate both the partition function and log-likelihood gradient. Aware of how critical the choice of sampling distribution is when using sampled trajectories to estimate the function, Finn, Levine, and Abbeel (2016) and Fu et al. (2017) proposed to exploit deep policy optimization and generative adversarial networks to simultaneously learn the sampling distribution that best matches the maximum entropy trajectory distribution with respect to the current reward function parameters and the objective function parameters themselves. By doing so, the proposed methods can simultaneously learn the expert's objective function and policy.

Mori et al. (2011) proposed a model-free apprenticeship learning for transferring human behavior to the robot, evaluated on a ball-in-a-cup scenario. They rely on the implicitly encoded dynamics information in the human demonstrations rather than the need for explicit dynamics model. Similarly, Abbeel and Ng (2004), Brown and Niekum (2019), and Cockcroft et al. (2020) implicitly model the agent through expert demonstrations.

6.4. Stochastic vs. deterministic policy/controller

In a situation where randomness is oblivious to the agent's intended actions, deterministic policies should be optimal in theory (Ng & Russell, 2000; Syed et al., 2008). However, in practice, it is almost always the case that the agent does not have access to a perfect model of the environment and necessarily has to approximate a policy or value function that aliases many different underlying environmental states. In this case, a deterministic policy may have a systematic bias. Adding some stochasticity to the policy allows the agent to eventually break out these situations.

While both deterministic (*e.g.*, Ng & Russell, 2000; Reddy et al., 2012; Syed et al., 2008) and stochastic (*e.g.*, Boularias & Chaib-draa, 2010; Klein et al., 2013; Levine et al., 2011) policies have been assumed in the literature, almost all of the IOC approaches applied on human motion analysis have been formulated for deterministic systems. An exception is the work done in Li et al. (2011) in which both deterministic and stochastic systems are considered. The authors include noise as control dependent and additive term in the dynamics of the system. Then, the necessary and sufficient condition of the control signal to be optimal is defined based on the Hamilton–Jacobi–Bellman equation, and then the IOC approach is applied on the planar biological arm movement in simulation.

Summary: Most objective learning approaches assume known system and environment dynamics. Works considering human body movement typically assume known kinematic or dynamic constraints, with continuous state and action spaces. These models are mostly deterministic, but a few consider controller noise. Recent approaches have begun to relax these assumptions towards inferring the objective without knowledge of the model, and considering stochastic demonstrator policies.

7. Properties of the observations

Most IOC and IRL objective learning approaches assume that the trajectory has been generated by an expert demonstrator and is therefore strictly optimal (Argall et al., 2009). However, in real-life situations, strictly optimal noiseless observations are rarely available from human demonstrators.

7.1. Quality

In practical applications, in addition to expert, successful demonstrations, failed and/or incomplete demonstrations, demonstrations that were not necessarily performed by an expert (*i.e.*, sub-optimal demonstration trajectories) may also be provided. Shiarlis et al. (2016) argue and later demonstrate that by including these imperfect demonstrations during objective learning the degeneracy of the IRL problem can be further reduced and faster and better reward learning can be achieved. Therefore, it is essential to take into account the expert performance factors to allow objective learning algorithms to generalize beyond expert demonstrators.

BIRL (Ramachandran, 2007; Rothkopf & Dimitrakakis, 2011) and MaxEnt IRL (Ziebart, Maas, Bagnell, & Dey, 2008) approaches explicitly model the belief about the expertise of the demonstrator as a parameter, which can be specified for each demonstration. However, these methods still required demonstration trajectories for which the intended goal was achieved. If the demonstration fails to achieve this goal, a critical challenge for objective learning is how to detect, interpret and leverage these failed trajectories. On the one hand, detecting that a demonstration trajectory is a failure often requires manual clustering (Grollman & Billard, 2011) or labeling (Shiarlis et al., 2016), which can be a tedious and error-prone process. On the other hand, a failed trajectory is ambiguous because it is not clear what is wrong with it: was the entire trajectory incorrect, or was it almost correct but wrong with respect to one particular feature? Thus failed trajectories can be hard to exploit during learning.

In regards to the detection of failed demonstrations, Freire da Silva et al. (2006) used an evaluator, which can distinguish between two policies, but has difficulties in giving direct instructions. Relating to the interpretation and use of failed demonstration for objective learning, in Zheng et al. (2014), failures are treated as noise (Zheng et al., 2014) and filtered out during learning. Shiarlis et al. (2016) propose to extended MaxEnt IRL to include failed demonstrations during learning. To do so, they augment Eq. (14) with a new equality constraint and a regularization term in the optimization objective. While the former encourages feature expectations under the current reward function to be dissimilar to the empirical expectations of the failed demonstrations, the latter balances the maximization of the policy's entropy against the maximization of the dissimilarity between the optimal policy's feature expectations and the empirical expectations of the failed demonstrations. Leveraging some of the principles behind semi-supervised learning, Audiffren et al. (2015) extend the MaxEnt IRL framework to include both expert demonstrations and demonstrations for which their quality cannot be ascertained. The latter are referred to as unsupervised trajectories and can include trajectories from other experts, trajectories for which a different reward function was maximized or noisy data. The core idea of the proposed algorithm is to bias the learning towards reward functions that assign similar rewards to intrinsically similar trajectories. To do so, the authors include a pairwise penalty on the unsupervised trajectories in the optimization objective of Eq. (14). This penalty discourages reward functions that assign very different rewards to similar trajectories.

7.2. Observability

Most of the methods covered so far assume that the system's state is fully observable and the objective learning algorithm has access to all potentially relevant information. However, in real-word applications, the input data can be noisy (Kitani et al., 2012), there is limited sensing for relevant features (Henry et al., 2010), or a relevant component of the expert's actions is non directly observable in the collected expert demonstrations (Bogert et al., 2016). Different approaches have been proposed to address these issues. To deal with noisy vision-based observations, Kitani et al. (2012) proposed to include a notion of observation reliability in the computation of the maximum entropy trajectories probabilities (*i.e.*, Eq. (12)). In this way the reliance on the noisy input is minimized when the tracking algorithm used to observe the scene features indicates low precision. Henry et al. (2010) addressed the case in which relevant features are only observable for a short duration in time. They used Gaussian Process to estimate the changes of these features along the observed trajectories and replaced the computation of the likelihood gradient along a complete trajectory for a local approximation in which only the observations over a predefined window are taken into account. Finally, to account for missing, yet relevant state–action observations, Bogert et al. (2016) model the missing observations as hidden variables and propose to compute the conditioned feature expectations of the hidden state–action pairs given the observed portions of the demonstrations.

Summary: Most approaches in the literature assume that human motions are (approximately) optimal and noise does not change the general properties of the trajectory. However, real human demonstrations may include suboptimal and failed demonstrations, as well as variability in the demonstrator's expertise. Similarly, the system may not be fully observable. Recent approaches have proposed methods that can handle variability in the quality and observability of the demonstrations.

8. Applications and validation

As noted in Section 1, three key motivations are commonly cited for employing objective learning methods: (1) to model and generate trajectories, (2) provide insight into why a given trajectory was selected out of all possible trajectories, and (3) generalize demonstration motions to other embodiments or tasks. This section explores the extent that these goals have been demonstrated through applications and experiments.

Table 3 highlights the 71 papers that utilize human motion demonstration data, highlighting the 22 papers that perform movement insight analysis, 21 papers that generalize from the specific task learned, and 7 papers that generalize from the demonstration system model to another system. Another 35 papers validate using simulations only.

This table highlights that a majority of the tasks used to validate objective learning algorithms tend to be some form of planar locomotion, where a rigid body system model moves in 2 dimensional space from one location to another. These include vehicle navigation (Abbeel et al., 2008; Silver et al., 2010; Vogel et al., 2012; Ziebart, Maas, Dey, & Bagnell, 2008), helicopter control (Coates et al., 2009), and navigating through a room with obstacles or people (Chung & Huang, 2010; Lee et al., 2014; Lee & Popović, 2010; Okal & Arras, 2016; Pfeiffer et al., 2016). Another common task category are simple arm motions such as pick-and-place (Albrecht et al., 2012; Kalakrishnan et al., 2010; Lin et al., 2018), grasping (El-Hussieny et al., 2016; Kalakrishnan et al., 2013; Ratliff et al., 2007), or object sorting/manipulation tasks (Bogert et al., 2016; Jin, Petrich, et al., 2019; Sermanet et al., 2016).

8.1. Validation techniques

To validate the proposed approaches, most works consider one or more of the following strategies: validation with simulated data, human data, or noise-corrupted data.

Simulation Data: Objective learning methods aim to recover the underlying objective function from demonstration trajectories. As the ground truth objective functions are impossible to obtain from human demonstrators, simulations are commonly used to validate that the recovered objective function is accurate. This is typically accomplished by implementing a controller that generates optimal demonstration trajectories given a pre-defined objective function. The generated demonstration trajectories are then used in the proposed algorithms to recover the original objective function. A majority of these tasks are discrete-space gridworld type applications (Babes-Vroman et al., 2011; Herman et al., 2016; Neu & Szepesvari, 2012), but also have been applied to continuous-space models (Aghasadeghi & Bretl, 2011; Johnson et al., 2013) as well.

Human Data: For human data, methods are typically validated by assessing to what extent the optimal trajectory corresponding to the recovered objective function matches either the original demonstration trajectory, or some metric derived from the demonstration. This type of validation has been applied to gait (Liu et al., 2005) and locomotion (Levine & Koltun, 2012; Levine et al., 2011; Puydupin-Jamin et al., 2012) tasks.

A notable subset of human data validation are methods that use kinesthetic teaching to provide demonstration data and use the resultant objective function to regenerate the demonstrations. While they do not tend to verify the trajectory error, they replay the trajectory on a robot to verify that the task can be replicated. These tasks have been carried on object manipulation tasks on the Barrett WAM (Kalakrishnan et al., 2013) and the PR-2 (Finn, Levine, & Abbeel, 2016).

Non-optimal Assumptions: While a majority of the algorithms require the demonstration trajectory to be strictly optimal, in real-life applications, strictly optimal data is impossible to guarantee if the data was not simulated, due to suboptimal trajectories or noisy sensors (Byravan et al., 2015; Jin, Kulić, et al., 2019; Jin, Wang, et al., 2019; Johnson et al., 2013; Oguz et al., 2018; Park & Levine, 2013; Rebula et al., 2019; Yin et al., 2016). Papers tend to apply strong pre-filtering (Lin et al., 2016; Panchea et al., 2018; Westermann et al., 2020), or minimize the degree of optimality violations (Puydupin-Jamin et al., 2012).

To investigate the sensitivity to approximately optimal demonstration trajectories, some researchers validate their results on noisy data (Menner et al., 2019; Ramachandran, 2007; Ziebart, Maas, Bagnell, & Dey, 2008), or manually corrupt observation data with injected control dependent (Kalakrishnan, 2014; Li et al., 2011; Zhang et al., 2019, 2018), or state dependent (Albrecht et al., 2012; Puydupin-Jamin et al., 2012) noise. Other researchers use multiple demonstrations to mitigate the impact of suboptimality affecting objective function accuracy (Rebula et al., 2019).

8.2. Movement insight analysis

The human body is a complex machine, comprising over 650 named muscles and 200 mechanical DoFs. Determining a single trajectory to carry out a specific task is a complex interplay of resolving joint redundancies, meeting specific movement objectives, and accounting for environment interactions.

As such, a number of papers focus on analyzing the recovered objectives to learn more about the nature of human movement. While this analysis is dependent on the composition of the objective functions (see Section 4) and the task being performed, several basis functions have commonly been found to be significant: Kinematic features, such as distance from rest/target (Majumdar et al., 2017; Yao & Billard, 2020), velocity (Menner et al., 2019; Westermann et al., 2020), acceleration (El-Hussieny et al., 2016; Jin, Kulić, et al., 2019; Lin et al., 2016), and jerk (Albrecht et al., 2012; Oguz et al., 2018) are common important features. Dynamic features, such as torque-related terms (Albrecht et al., 2012; Clever et al., 2016; Mombaur & Clever, 2017; Mombaur et al., 2013), power (Berret et al., 2011; Jin, Kulić, et al., 2019; Lin et al., 2016; Panchea et al., 2018; Sylla et al., 2014), and kinetic energy (Berret et al., 2011; Panchea et al., 2018; Sylla et al., 2014) have all be found to be significant terms in gait, full body exercise, and object manipulation and reaching tasks.

For locomotion/driving tasks, common significant terms include avoidance of excessive velocity (Choi & Kim, 2013), penalizing off-roading and backward driving Abbeel et al. (2008) and favoring periodicity in gait (Park & Levine, 2013).

8.3. Generalization

A key hypothesized benefit of objective learning is that the resulting representation of the demonstration is in abstract form so that it is easy to generalize to other contexts. However, fewer papers actually validate this hypothesis with experiments.

8.3.1. To different tasks

A majority of the papers focus on replicating the trajectories of the demonstration. However, it is generally not useful in the real world to be able to replicate only a specific trajectory without some ability to generalize to different initial/ending conditions or to a different environment.

Task generalization to different conditions has been demonstrated in several domains. These range from simple tasks, such as varied starting and ending position of vehicles in driving simulators (Abbeel & Ng, 2004; Ziebart, Maas, Bagnell, & Dey, 2008; Ziebart, Maas, Dey, & Bagnell, 2008), position and orientation of doors and other graspable objects for pick-and-place and object manipulation tasks (Bogert et al., 2016; Byravan et al., 2015; Englert et al., 2017; Jin, Petrich, et al., 2019; Jin, Petrich, Zhang, et al., 2020; Melo et al., 2007; Ratliff et al., 2007; Sermanet et al., 2016; Shukla et al., 2017), or varying height, position, or orientation of steps and stairs (Aghasadeghi & Bretl, 2014; Clever et al., 2018; Mombaur et al., 2010; Park & Levine, 2013).

Table 3
Algorithm applications and validations grouped by general task categories. For each task category, the number of papers that validated on simulation only (PS) or had human data (PH) included is denoted. The demonstrator model, significant objective basis function terms (SOBF). For kinematic and torque SOBFs, higher order derivatives may also be used as significant basis functions. The tasks (gen task) and system model (gen model) generalization are also specified.

Task Category	Task Examples	PS	PH	Model	SOBF	Gen Task	Gen Model
Arm, complex	Table tennis, handwriting, fine manipulation with tools	0	5	Articulated upper body, rigid	Kinematics	Handwriting	C1
Arm, simple	Grasping, turning, moving, rotating, stacking, insertion, and sorting objects.	4	26	Articulated upper body	Kinematics, torque, kinetic energy, power	Object type, environment, location, pose	C2
Cartesian locomotion	Gridworld, point-to-point navigation, following an object, video game playthrough	35	26	Rigid	Kinematics, kinetic energy	Environment, location, pose	-
Full body exercise	Squat, stairs, table tennis, long jump	1	5	Articulated full body, rigid	Kinematics, power	Height and length	-
Gait	Walking, running	1	9	Articulated full body	Kinematics, torque, stability, step length	Height, length, orientation	C3
Others	Gaze tracking, balancing, simulated dynamics	3	3	Articulated full body, rigid	–	–	–

C1 Yin et al. (2016) C2 Bogert et al. (2016), Jin, Petrich, Zhang, et al. (2020), Mori et al. (2011), Sermanet et al. (2016) C3 Clever et al. (2018), Mombaur et al. (2010)

Generalization to varying environments, such as positions of obstacles or walls (Kim & Pineau, 2016; Kitani et al., 2012; Xia & El Kamel, 2016), or moving crowds of people (Chung & Huang, 2010; Okal & Arras, 2016; Pfeiffer et al., 2016; Ziebart et al., 2009) have also been demonstrated.

8.3.2. To different embodiments

While commonly cited as a motivating factor, few papers take the recovered objective function and demonstrate that it is feasible on a different agent.

An early example is Mombaur et al. (2010), using the bi-level approach, where the Cartesian and heading trajectory of 10 participants walking to a set position and heading were recorded. Using the recovered objective function based on the minimization of time elapsed, acceleration, and orientation error from goal, they were able to demonstrate re-targetting on the HRP-2 (Mombaur & Clever, 2017; Mombaur et al., 2010) and the iCub (Mombaur & Clever, 2017) humanoid robots in simulation. Using a more complex objective function composing of angular torque, momentum, center of mass oscillations, and foot velocity generated from 6 participants walking on raised surfaces, Clever et al. (2018) successfully generated trajectories that allowed the iCub imitate the demonstrated behavior with some tuning to account for joint range and velocity differences between humans and robot. While tracking error of the actual robot to the generated trajectory was much higher than the iCub simulation, it nevertheless was successful at stepping on all the raised surfaces without missing.

Bogert et al. (2016), Jin, Petrich, Zhang, et al. (2020), Mori et al. (2011), and Sermanet et al. (2016) utilized motion capture or video data to estimate objective functions for ball sorting, ball hitting, and cup pouring tasks with the MACCEPA, Phantom Pincher, and R3D32 robot arms, while Yin et al. (2016) utilized a human handwriting dataset consisting of Cartesian position and velocity information (Llorens et al., 2008) to learn the objective functions for the English alphabet and replicated the task on a Baxter manipulator.

9. Challenges and limitations

9.1. A priori known basis features

Our survey shows that although most algorithms assume that demonstrations can be readily obtained from experts, a significant amount of time and effort is still required to design basis features that effectively capture the nature of the task being considered. In particular, not all tasks can be readily translated into a set of features and their corresponding mathematical form. Furthermore, even though novel methods relying on Gaussian processes (Joukov & Kulic, 2017) and deep neutral networks (Finn, Christiano, et al., 2016) avoid the manual specification of potentially relevant basis features, they still require the addition of well engineered regularization terms in order to accomplish good performance.

A critical, and still open, challenge in objective learning is that many objective functions could explain the observed behavior. That is, several different objective functions may map to the same optimal trajectory as components of the objective function may be singular given the trajectory constraints. Thus, it is hard to determine whether the set of *a priori* defined basis features or those implicitly learned by a neural network are indeed correct and relevant for the task at hand. Furthermore, most methods rely on the performance of the control policy learned from the recovered objective function in order to assess the quality of the learned objective. This can be a very costly, and sometimes prohibitive, process to execute with incomplete or potentially irrelevant sets of candidate features. Jin, Kulić, et al. (2019) proposed a possible approach to assess the suitability of candidate features over a given window, but only validated against a small dataset and simple movements.

9.2. Task complexity and evaluation

A common dividing line between IOC and IRL papers is that model complexity tends to be higher in IOC methods. Of the papers examined in Table 3, only 2 of the 14 IRL papers included in the table were found to target an application that utilized an articulated full body model. This is likely due to the computational complexity of most IRL methods. Indeed, IRL methods require to solve the forward problem in the inner loop of an iterative optimization process. This makes them difficult to apply to complex, high-dimensional tasks, where the forward problem is itself already challenging (Finn, Levine, & Abbeel, 2016). A large body of IRL work considers 2D locomotion tasks that do not require complex dynamic models. While both IOC and IRL methods demonstrate their recovered trajectories on real-life robots,

IOC methods tend to focus more on articulated structures while IRL focus on rigid body form factors.

IOC methods are typically applied to a single demonstration trajectory. This is possible since IOC methods rely on a robustly defined dynamic model, but may result in overfitting to the demonstration trajectory and/or dynamic model. In contrast, IRL methods typically infer the objective function from a larger number of trajectories. While this is in part of due to enable learning the components of the MDP that are not pre-defined, it also may lead to improved ability to resist overfitting.

9.3. Optimality and properties of demonstration trajectories

A major limitation of many methods is the assumption that the demonstration trajectory is strictly and globally optimal. This means that the demonstration trajectories must be extensively filtered to remove noise (Lin et al., 2016; Panchea et al., 2017), which may remove important characteristics for the objective function recovery, and also means that the algorithm cannot handle suboptimal trajectories or failed demonstrations. Several papers attempt to demonstrate robustness to noise (Englert et al., 2017; Puydupin-Jamin et al., 2012).

Although methods such as MaxEnt or BIRL are less strict on the assumptions they made about the overall quality of the demonstration trajectories, most approaches assume that the observed human behavior is in some way globally optimal with respect to the unknown objective function and might fail to learn when provided with trajectories that are locally optimal instead (e.g., a skilled driver might execute every car manoeuvre correctly and still follow a sub-optimal path). Some papers attempt to address this limitation by only considering the shape of the objective function around the demonstration trajectories (Levine & Koltun, 2012), explicitly including sub-optimal (Brown et al., 2020), or failed demonstrations (Shiarlis et al., 2016). However, these extensions still require an *a priori* method that can ascertain the quality of the demonstration trajectories and whether they are successfully or failed examples.

Finally, independently of the quality and optimality of the demonstration trajectories, most existing methods require complete system trajectories within an entire time horizon. Jin et al. (2018) proposed a possible approach to assess the suitability of candidate features over a given window, but was only validated against a small dataset and simple movements. Jin, Murphey, and Mou (2020) have proposed an approach that can learn an objective function from a collection of demonstration segments. However, their method is limited to objective functions with a small number of features and that are parameterized as weighted sums of these features.

9.4. Local constraints estimation

With the exception of a few papers (e.g., Choi & Kim, 2012; Jin, Kulić, et al., 2019; Lin et al., 2016; Michini & How, 2012), the majority of IOC and IRL methods examined in this survey work under the assumptions that (1) the expert's demonstration trajectories were produced according to single objective function, and (2) this unknown objective function alone can fully explain the observed expert's behavior. However, in real-world applications, humans are plausibly switching between multiple locally consistent objective functions (Nguyen et al., 2015). This is of particular interest when demonstrations are globally sub-optimal, yet they are likely consistent with objective functions that apply only to a specific region of the state space (*i.e.*, locally optimal).

Some approaches seek to discover and learn multiple objective functions from a single demonstration. However, as noted by Chou et al. (2020a), an expert's action may also be motivated by unknown local constraints that apply to one or multiple segments along the demonstration trajectory and that are critical for guaranteeing aspects such as safety during motion. Although some of these constraints can be reformulated as objective functions, the latter approach can relax the constraints and allow violations. To our knowledge, only the work done in Park et al. (2020) and Chou et al. (2020a) aims at simultaneously inferring the task function objectives and constraints from a single demonstration. However, these approaches require a carefully designed feature space or constraint space, and full knowledge about the system dynamics.

9.5. Limited performance and generalizability

The generalizability and performance of the recovered objective function and any policy trained using this function is typically upper-bounded by the quality and performance of the expert's demonstrations (Brown et al., 2019). Thus, when presented with sub-optimal noisy behavior, policies trained using the objective functions recovered using IRL methods such as AL (Abbeel & Ng, 2004) and MMP (Ratliff, Bradley, & Zinkevich, 2006) might never perform better than the expert.

A few methods aim at solving this problem. Syed et al. (2008) showed that it is possible to obtain a policy that is substantially better than the expert's if additional information about which features have a positive or negative contribution to unknown objective function is incorporated during learning. However, their method requires carefully hand-crafted features and prior knowledge about the true signs of these features. Recently, Brown et al. (2019) proposed an approach that aims at learning an objective function from ranked demonstrations. Their empirical results indicate that better-than-expert demonstration can be achieved with the proposed approach. However, their method relies on a subjective manual raking process and lacks of theoretical guarantees on when an improvement over an expert's performance can be successfully achieved.

9.6. Choice of dynamic model

An underlying characteristic of the objective learning problem is that the human's expert policy as seen through the demonstration trajectories is influenced by both the unknown objective function and (unknown) system dynamics. Thus, the robustness and accuracy of objective learning algorithms is affected by the choice of dynamic model. In other words, an erroneous and/or inaccurate dynamics model can potentially lead to the miss-estimation of the human expert's objective function (Herman et al., 2016).

Our survey shows that most IOC methods assume that whatever dynamic model has been chosen is true and accurately captures all relevant information about the dynamic system under consideration. However, prior work indicates that when learning from human experts, there may be a potential mismatch between the what the IOC algorithm beliefs to be the dynamic model of the human and the actual human dynamics (Golub et al., 2013).

IRL algorithms attempt to bypass the need to choose a model by either learning it from the demonstration trajectories or exploiting model-free techniques. On the one hand, model-based techniques use the provided demonstrations to approximate the unknown dynamic model. However, since these demonstrations are issued by the expert's optimal policy, they are biased towards states with high expected rewards and thus can result in inaccurate dynamic transition estimates for the states and actions that were seldom observed. On the other hand, model-free approaches fully omit the need of a model of the dynamics at the cost of an increase in computation complexity. For instance, Kostrikov et al. (2018) note that although model-free IRL approach based Generative Adversarial Networks (*e.g.*, Fu et al., 2017; Ho & Ermon, 2016) can learn from a small sample of expert demonstrations (*e.g.*, 4 expert demonstration trajectories), they can potentially require millions of transition dynamic samples from the real and/or simulated environment.

10. Conclusion and future directions

This survey provides an overview of approaches for objective learning from human demonstrations. We provide a unifying view of both control and learning approaches, and review algorithms based on the choice of objective function form, assumed dynamics, similarity metrics between the observed and estimated behavior, properties of observations, and the validation approach. Our survey shows that the great majority of papers assume that the objective function can be modeled as a weighted sum of known basis functions, that the dynamics of the demonstrator and their environment are known, that complete demonstrations are available and that the demonstrator is an expert. Most approaches use a computationally expensive bi-level solution where the forward optimal problem is solved in the inner loop, as illustrated in Fig. 2. A key differentiator among algorithms is how the demonstrator data and the output of the inferred objective are compared: comparing the features, or more generally the value function, or comparing the trajectories. More recent works have begun to address some of these limiting assumptions, but a number of open challenges remain:

Objective function structure: The choice of objective function structure, e.g., the basis functions, is often ad-hoc and poorly motivated. Methods for systematically identifying the basis function set are needed. While non-parametric methods avoid the need for basis function specification, they lack interpretability.

Estimating both objectives and constraints: Most approaches focus on estimating task objectives, but assume that the constraints are absent or known *a priori*. A recent exception are Park et al. (2020) and Chou et al. (2020b). Park et al. (2020) propose an approach for simultaneously inferring the task objectives and constraints from a single demonstration. However, their approach considers only very simple goal and constraint hypotheses, and relies on a carefully designed feature space. Chou et al. (2020b) propose an approach based on KKT conditions and a known constraint parametrization, and demonstrate with high-dimensional constraints and systems by learning constraints for 7-DOF arm and quadrotor examples.

Learning from multiple experts: Most approaches assume a single demonstrator, or multiple demonstrators who share the same objectives and expertise. To enable robots to learn from teams of users, approaches that can handle multiple experts are desirable. Chen et al. (2020) propose an IRL approach that can disambiguate between task objectives and heterogeneous demonstrator preferences, titled multi-style reward distillation. The reward function is modeled as a combination of the task goal and the demonstrator's preferences. They learn a neural network for the task reward and strategy reward functions, based on Adverse IRL (Fu et al., 2017) using network distillation. The proposed approach is tested in simulation and on a tennis motion dataset, collected from kinesthetic teaching.

Handling demonstrators with varying expertise: The great majority of current approaches assume that the demonstrations are either given by an expert, or that the demonstrator's level of expertise is known *a priori*. When obtaining demonstrations from a new user, it would be useful to simultaneously estimate both the objective and the expertise, as recently proposed in Carreno-Medrano et al. (2020).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was undertaken, in part, thanks to funding from the Canada Excellence Research Chairs Program and the Natural Sciences and Engineering Research Council Alexander Graham Bell Canada Graduate Scholarship. All funding agencies had no involvement in the writing and submission of this work.

References

Ab Azar, N., Shahmansoorian, A., & Davoudi, M. (2020). From inverse optimal control to inverse reinforcement learning: A historical review. *Annual Reviews in Control*, http://dx.doi.org/10.1016/j.arcontrol.2020.06.001.

Abbeel, P., Coates, A., & Ng, A. Y. (2010). Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotics Research*, *29*(13), 1608–1639. http://dx.doi.org/10.1177/0278364910371999.

Abbeel, P., Dolgov, D., Ng, A. Y., & Thrun, S. (2008). Apprenticeship learning for motion planning with application to parking lot navigation. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 1083–1090). http://dx.doi.org/10.1109/IROS.2008.4651222.

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *International conference on machine learning* (pp. 1–8). http://dx.doi.org/10.1145/1015330.1015430.

Aghasadeghi, N., & Bretl, T. (2011). Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 1561–1566). http://dx.doi.org/10.1109/IROS.2011.6094679.

Aghasadeghi, N., & Bretl, T. (2014). Inverse optimal control for differentially flat systems with application to locomotion modeling. In *IEEE international conference on robotics and automation* (pp. 6018–6025). http://dx.doi.org/10.1109/ICRA.2014.6907746.

Albrecht, S., Leibold, M., & Ulbrich, M. (2012). A bilevel optimization approach to obtain optimal cost functions for human arm movements. *Numerical Algebra, Control & Optimization*, *2*(1), 105–127. http://dx.doi.org/10.3934/naco.2012.2.105.

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, *57*(5), 469–483. http://dx.doi.org/10.1016/j.robot.2008.10.024.

Arora, S., & Doshi, P. (2018). A survey of inverse reinforcement learning: Challenges, methods and progress. arXiv:1806.06877, arXiv:1806.06877 [cs.LG].

Audiffren, J., Valko, M., Lazaric, A., & Ghavamzadeh, M. (2015). Maximum entropy semi-supervised inverse reinforcement learning. In *International joint conference on artificial intelligence*.

Babes-Vroman, M., Marivate, V., Subramanian, K., & Littman, M. (2011). Apprenticeship learning about multiple intentions. In *International conference on machine learning* (pp. 897–904).

Berret, B., Chiovetto, E., Nori, F., & Pozzo, T. (2011). Evidence for composite cost functions in arm movement planning: An inverse optimal control approach. *PLoS Computational Biology*, *7*(10), Article e1002183. http://dx.doi.org/10.1371/journal.pcbi.1002183.

Betts, J. T. (1998). Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, *21*(2), 193–207. http://dx.doi.org/10.2514/2.4231.

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot programming by demonstration. In *Springer handbook of robotics* (pp. 1371–1394). http://dx.doi.org/10.1007/978-3-540-30301-5_60.

Bogert, K., Lin, J. F.-S., Doshi, P., & Kulic, D. (2016). Expectation–maximization for inverse reinforcement learning with hidden data. In *International conference on autonomous agents & multiagent systems* (pp. 1034–1042).

Boularias, A., & Chaib-draa, B. (2010). Bootstrapping apprenticeship learning. In *Advances in neural information processing systems (vol. 23)* (pp. 289–297).

Boularias, A., Kober, J., & Peters, J. (2011). Relative entropy inverse reinforcement learning. In *JMLR workshop and conference (vol. 15)* (pp. 182–189).

Boularias, A., Krömer, O., & Peters, J. (2012). Structured apprenticeship learning. In *Lecture notes in computer science, Machine learning and knowledge discovery in databases* (pp. 227–242). http://dx.doi.org/10.1007/978-3-642-33486-3_15.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*.

Brown, D., Goo, W., Nagarajan, P., & Niekum, S. (2019). Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning (vol. 97)* (pp. 783–792).

Brown, D. S., Goo, W., & Niekum, S. (2020). Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on robot learning*. arXiv:1907.03976, (in print).

Brown, D. S., & Niekum, S. (2019). Machine teaching for inverse reinforcement learning: Algorithms and applications. In *AAAI conference on artificial intelligence* (pp. 7749–7758). http://dx.doi.org/10.1609/aaai.v33i01.33017749.

Byravan, A., Monfort, M., Ziebart, B., Boots, B., & Fox, D. (2015). Graph-based inverse optimal control for robot manipulation. In *International joint conference on artificial intelligence* (pp. 1874–1890).

Carreno-Medrano, P., Harada, T., Lin, J. F.-S., Kulic, D., & Venture, G. (2019). Analysis of affective human motion during functional task performance: An inverse optimal control approach. In *IEEE/RAS international conference on humanoid robots* (pp. 461–468). http://dx.doi.org/10.1109/Humanoids43949.2019.9035007.

Carreno-Medrano, P., Smith, S. L., & Kulic, D. (2020). Joint estimation of expertise and reward preferences from human demonstrations. arXiv:2011.04118, arXiv:2011.04118 [cs.RO].

Chen, L., Paleja, R., Ghuy, M., & Gombolay, M. (2020). Joint goal and strategy inference across heterogeneous demonstrators via reward network distillation. In *ACM/IEEE international conference on human-robot interaction* (pp. 659–668). http://dx.doi.org/10.1145/3319502.3374791.

Chen, S.-y., Qian, H., Fan, J., Jin, Z.-j., & Zhu, M.-l. (2010). Modified reward function on abstract features in inverse reinforcement learning. *Journal of Zhejiang University Science C*, *11*(9), 718–723. http://dx.doi.org/10.1631/jzus.C0910486.

Choi, J., & Kim, K.-E. (2011a). Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, *12*, 691–730.

Choi, J., & Kim, K.-e. (2011b). MAP Inference for Bayesian inverse reinforcement learning. In *Advances in neural information processing systems* (pp. 1989–1997).

Choi, J., & Kim, K.-e. (2012). Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In *Advances in neural information processing systems* (pp. 305–313).

Choi, J., & Kim, K.-E. (2013). Bayesian nonparametric feature construction for inverse reinforcement learning. In *International joint conference on artificial intelligence* (pp. 1287–1293).

Chou, G., Ozay, N., & Berenson, D. (2020a). Learning constraints from locally-optimal demonstrations under cost function uncertainty. *IEEE Robotics and Automation Letters*, *5*(2), 3682–3690.

Chou, G., Ozay, N., & Berenson, D. (2020b). Learning constraints from locally-optimal demonstrations under cost function uncertainty. arXiv:2001.09336.

Chung, S.-Y., & Huang, H.-P. (2010). A mobile robot that understands pedestrian spatial behaviors. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 5861–5866). http://dx.doi.org/10.1109/IROS.2010.5649718.

Clever, D., Hu, Y., & Mombaur, K. (2018). Humanoid gait generation in complex environments based on template models and optimality principles learned from human beings. *International Journal of Robotics Research*, *37*(10), 1184–1204. http://dx.doi.org/10.1177/0278364918765620.

Clever, D., & Mombaur, K. (2017). On the relevance of common humanoid gait generation strategies in human locomotion: An inverse optimal control approach. In *Modeling, simulation and optimization of complex processes HPSC* (pp. 27–40). http://dx.doi.org/10.1007/978-3-319-67168-0_3.

Clever, D., Schemschat, R. M., Felis, M. L., & Mombaur, K. (2016). Inverse optimal control based identification of optimality criteria in whole-body human walking on level ground. In *IEEE international conference on biomedical robotics and biomechatronics* (pp. 1192–1199). http://dx.doi.org/10.1109/BIOROB.2016.7523793.

Coates, A., Abbeel, P., & Ng, A. Y. (2009). Apprenticeship learning for helicopter control. *Communications of the ACM*, *52*(7), 97–105. http://dx.doi.org/10.1145/1538788.1538812.

Cockcroft, M., Mawjee, S., James, S., & Ranchod, P. (2020). Learning options from demonstration using skill segmentation. In *International SAUPEC/RobMech/PRASA conference* (pp. 1–6). http://dx.doi.org/10.1109/SAUPEC/RobMech/PRASA48453.2020.9040988.

Dimitrakakis, C., & Rothkopf, C. A. (2011). Bayesian multitask inverse reinforcement learning. In *Lecture notes in computer science*: *vol. 7188*, *Recent advances in reinforcement learning* (pp. 273–284). http://dx.doi.org/10.1007/978-3-642-29946-9_27.

Doerr, A., Ratliff, N., Bohg, J., Toussaint, M., & Schaal, S. (2015). Direct loss minimization inverse optimal control. In *Robotics: Science and systems*. http://dx.doi.org/10.15607/RSS.2015.XI.013.

Dvijotham, K., & Todorov, E. (2010). Inverse optimal control with linearly-solvable MDPs. In *International conference on machine learning* (pp. 335–342).

El-Hussieny, H., Abouelsoud, A. A., Assal, S. F., & Megahed, S. M. (2016). Adaptive learning of human motor behaviors: An evolving inverse optimal control approach. *Engineering Applications of Artificial Intelligence*, *50*, 115–124.

El-Hussieny, H., & Ryu, J.-H. (2019). Inverse discounted-based LQR algorithm for learning human movement behaviors. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *49*(4), 1489–1501.

Englert, P., & Toussaint, M. (2018). Learning manipulation skills from a single demonstration. *International Journal of Robotics Research*, *37*(1), 137–154. http://dx.doi.org/10.1177/0278364917743795.

Englert, P., Vien, N. A., & Toussaint, M. (2017). Inverse KKT: Learning cost functions of manipulation tasks from demonstrations. *International Journal of Robotics Research*, *36*(13–14), 1474–1488. http://dx.doi.org/10.1177/0278364917745980.

Fang, B., Jia, S., Guo, D., Xu, M., Wen, S., & Sun, F. (2019). Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications*, *3*(4), 362–369. http://dx.doi.org/10.1007/s41315-019-00103-5.

Finn, C., Christiano, P., Abbeel, P., & Levine, S. (2016). A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. arXiv:1611.03852, arXiv:1611.03852 [cs.LG].

Finn, C., Levine, S., & Abbeel, P. (2016). Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning* (pp. 49–58).

Freire da Silva, V., Reali Costa, A., & Lima, P. (2006). Inverse reinforcement learning with evaluation. In *IEEE international conference on robotics and automation* (pp. 4246–4251). http://dx.doi.org/10.1109/ROBOT.2006.1642355.

Fu, J., Luo, K., & Levine, S. (2017). Learning robust rewards with adversarial inverse reinforcement learning. arXiv:1710.11248, arXiv:1710.11248 [cs.LG].

Golub, M., Chase, S., & Yu, B. (2013). Learning an internal dynamics model from control demonstration. In Sanjoy Dasgupta, David McAllester (Eds.), *Proceedings of machine learning research: vol. 28*, *International conference on machine learning* (pp. 606–614).

Grollman, D. H., & Billard, A. (2011). Donut as I do: Learning from failed demonstrations. In *IEEE international conference on robotics and automation* (pp. 3804–3809). http://dx.doi.org/10.1109/ICRA.2011.5979757.

Hatz, K. (2014). *Efficient numerical methods for hierarchical dynamic optimization with application to cerebral palsy gait modeling* (Dissertation), Ruprecht Karl University of Heidelberg, http://dx.doi.org/10.11588/heidok.00016803.

Hatz, K., Schlöder, J. P., & Bock, H. G. (2012). Estimating parameters in optimal control problems. *SIAM Journal on Scientific Computing*, *34*(3), A1707–A1728. http://dx.doi.org/10.1137/110823390.

Henry, P., Vollmer, C., Ferris, B., & Fox, D. (2010). Learning to navigate through crowded environments. In *IEEE international conference on robotics and automation* (pp. 981–986). http://dx.doi.org/10.1109/ROBOT.2010.5509772.

Herman, M., Gindele, T., Wagner, J., Schmitt, F., & Burgard, W. (2016). Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *International conference on artificial intelligence and statistics (vol. 51)* (pp. 102–110).

Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in neural information processing systems (vol. 29)* (pp. 4565–4573).

Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys*, *50*(2), 1–35.

Jin, W., Kulić, D., Lin, J. F.-S., Mou, S., & Hirche, S. (2019). Inverse optimal control for multiphase cost functions. *IEEE Transactions on Robotics*, *35*(6), 1387–1398. http://dx.doi.org/10.1109/TRO.2019.2926388.

Jin, W., Kulić, D., Mou, S., & Hirche, S. (2018). Inverse optimal control with incomplete observations. arXiv:1803.07696, arXiv:1803.07696 [cs].

Jin, W., Murphey, T. D., & Mou, S. (2020). Learning from incremental directional corrections. arXiv:2011.15014, arXiv:2011.15014 [cs.RO].

Jin, J., Petrich, L., Dehghan, M., & Jagersand, M. (2020). A geometric perspective on visual imitation learning. arXiv:2003.02768, arXiv:2003.02768 [cs.RO].

Jin, J., Petrich, L., Dehghan, M., Zhang, Z., & Jagersand, M. (2019). Robot eye-hand coordination learning by watching human demonstrations: A task function approximation approach. In *IEEE international conference on robotics and automation* (pp. 6624–6630).

Jin, J., Petrich, L., Zhang, Z., Dehghan, M., & Jagersand, M. (2020). Visual geometric skill inference by watching human demonstration. arXiv:1911.04418, arXiv:1911.04418 [cs.RO].

Jin, Z.-j., Qian, H., Chen, S.-y., & Zhu, M.-l. (2011). Convergence analysis of an incremental approach to online inverse reinforcement learning. *Journal of Zhejiang University Science C*, *12*(1), 17–24.

Jin, W., Wang, Z., Yang, Z., & Mou, S. (2019). Pontryagin differentiable programming: An end-to-end learning and control framework. arXiv:1912.12970, arXiv:1912.12970 [cs.LG].

Johnson, M., Aghasadeghi, N., & Bretl, T. (2013). Inverse optimal control for deterministic continuous-time nonlinear systems. In *IEEE conference on decision and control* (pp. 2906–2913). http://dx.doi.org/10.1109/CDC.2013.6760325.

Joukov, V., & Kulic, D. (2017). Gaussian process based model predictive controller for imitation learning. In *IEEE/RAS international conference on humanoid robotics* (pp. 850–855).

Kalakrishnan, M. (2014). Learning objective functions for autonomous motion generation.

Kalakrishnan, M., Pastor, P., Righetti, L., & Schaal, S. (2013). Learning objective functions for manipulation. In *IEEE international conference on robotics and automation* (pp. 1331–1336).

Kalakrishnan, M., Theodorou, E., & Schaal, S. (2010). Inverse reinforcement learning with PI2.

Keshavarz, A., Wang, Y., & Boyd, S. (2011). Imputing a convex objective function. In *IEEE international symposium on intelligent control* (pp. 613–619). http://dx.doi.org/10.1109/ISIC.2011.6045410.

Kim, B., & Pineau, J. (2016). Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics*, *8*(1), 51–66. http://dx.doi.org/10.1007/s12369-015-0310-2.

Kitani, K. M., Ziebart, B. D., Bagnell, J. A., & Hebert, M. (2012). Activity forecasting. In *Lecture notes in computer science*, *European conference on computer vision* (pp. 201–214). http://dx.doi.org/10.1007/978-3-642-33765-9_15.

Klein, E., Geist, M., Piot, B., & Pietquin, O. (2012). Inverse reinforcement learning through structured classification. In *Advances in neural information processing systems (vol. 25)* (pp. 1007–1015).

Klein, E., Piot, B., Geist, M., & Pietquin, O. (2013). A cascaded supervised learning approach to inverse reinforcement learning. In *Lecture notes in computer science*, *Joint European conference on machine learning and knowledge discovery in databases* (pp. 1–16). http://dx.doi.org/10.1007/978-3-642-40988-2_1.

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, *32*(11), 1238–1274.

Kostrikov, I., Agrawal, K. K., Dwibedi, D., Levine, S., & Tompson, J. (2018). Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. arXiv:1809.02925, arXiv:1809.02925 [cs.LG].

Kretzschmar, H., Spies, M., Sprunk, C., & Burgard, W. (2016). Socially compliant mobile robot navigation via inverse reinforcement learning. *International Journal of Robotics Research*, *35*(11), 1289–1307. http://dx.doi.org/10.1177/0278364915619772.

Kroemer, O., Niekum, S., & Konidaris, G. (2019). A review of robot learning for manipulation: Challenges, representations, and algorithms. arXiv:1907.03146, arXiv:1907.03146 [cs.RO].

Kulić, D., Venture, G., Yamane, K., Demircan, E., Mizuuchi, I., & Mombaur, K. (2016). Anthropomorphic movement analysis and synthesis: A survey of methods and applications. *IEEE Transactions on Robotics*, *32*(4), 776–795.

Lee, G., Luo, M., Zambetta, F., & Li, X. (2014). Learning a super mario controller from examples of human play. In *IEEE congress on evolutionary computation* (pp. 1–8). http://dx.doi.org/10.1109/CEC.2014.6900246.

Lee, S. J., & Popović, Z. (2010). Learning behavior styles with inverse reinforcement learning. *ACM Transactions on Graphics*, *29*(4), 122:1–122:7. http://dx.doi.org/10.1145/1778765.1778859.

Levine, S., & Koltun, V. (2012). Continuous inverse optimal control with locally optimal examples. In *International coference on international conference on machine learning* (pp. 475–482).

Levine, S., Popovic, Z., & Koltun, V. (2010). Feature construction for inverse reinforcement learning. In *Advances in neural information processing systems (vol. 23)* (pp. 1342–1350).

Levine, S., Popovic, Z., & Koltun, V. (2011). Nonlinear inverse reinforcement learning with Gaussian processes. In *Advances in neural information processing systems (vol. 24)* (pp. 19–27).

Li, K., & Burdick, J. W. (2017a). Inverse reinforcement learning in large state spaces via function approximation. arXiv:1707.09394.

Li, K., & Burdick, J. W. (2017b). Meta inverse reinforcement learning via maximum reward sharing for human motion analysis. arXiv:1710.03592, arXiv:1710.03592 [cs.AI].

Li, W., Todorov, E., & Liu, D. (2011). Inverse optimality design for biological movement systems. *IFAC Proceedings Volumes*, *44*(1), 9662–9667. http://dx.doi.org/10.3182/20110828-6-IT-1002.00877.

Lin, J. F.-S., Bonnet, V., Panchea, A. M., Ramdani, N., Venture, G., & Kulić, D. (2016). Human motion segmentation using cost weights recovered from inverse optimal control. In *IEEE/RAS international conference on humanoid robots* (pp. 1107–1113). http://dx.doi.org/10.1109/HUMANOIDS.2016.7803409.

Lin, H.-I., Nguyen, X.-A., & Chen, W.-K. (2018). Active intention inference for robot-human collaboration. *International Journal of Computational Methods and Experimental Measurements*, *6*(4), 772–784. http://dx.doi.org/10.2495/CMEM-V6-N4-772-784.

Liu, C. K., Hertzmann, A., & Popović, Z. (2005). Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics*, *24*(3), 1071–1081. http://dx.doi.org/10.1145/1073204.1073314.

Liu, Y., Li, Z., Liu, H., & Kan, Z. (2020). Skill transfer learning for autonomous robots and human-robot cooperation: A survey. *Robotics and Autonomous Systems*, 103515:1–11. http://dx.doi.org/10.1016/j.robot.2020.103515.

Llorens, D., Prat, F., Marzal, A., Vilar, J. M., Castro, M. J., Amengual, J.-C., Barrachina, S., Castellanos, A., Boquera, S. E., Gomez, J. A., Gorbe, J., Gordo, A., Palazon, V., Peris, G., Ramos-Garijo, R., & Zamora, F. (2008). The UJIpenchars database: A pen-based database of isolated handwritten characters. In *International conference on language resources and evaluation* (pp. 2647–2651).

Mainprice, J., & Berenson, D. (2014). Learning cost functions for motion planning of human-robot collaborative manipulation tasks from human-human demonstration. In *AAAI fall symposium series* (pp. 107–109).

Majumdar, A., Singh, S., Mandlekar, A., & Pavone, M. (2017). Risk-sensitive inverse reinforcement learning via coherent risk models. In *Robotics: Science and systems (vol. 13)*. http://dx.doi.org/10.15607/RSS.2017.XIII.069.

Melo, F. S., Lopes, M., Santos-Victor, J., & Ribeiro, M. I. (2007). A unified framework for imitation-like behaviors. In *International symposium on imitation in animals and artifacts* (pp. 28–38).

Menner, M., Worsnop, P., & Zeilinger, M. N. (2019). Constrained inverse optimal control with application to a human manipulation task. *IEEE Transactions on Control Systems Technology*, http://dx.doi.org/10.1109/TCST.2019.2955663, (in print).

Menner, M., & Zeilinger, M. N. (2018). Convex formulations and algebraic solutions for linear quadratic inverse optimal control problems. In *European control conference* (pp. 2107–2112). http://dx.doi.org/10.23919/ECC.2018.8550090.

Michini, B., Cutler, M., & How, J. P. (2013). Scalable reward learning from demonstration. In *IEEE international conference on robotics and automation* (pp. 303–308). http://dx.doi.org/10.1109/ICRA.2013.6630592.

Michini, B., & How, J. P. (2012). Bayesian nonparametric inverse reinforcement learning. In *Lecture notes in computer science*, *Machine learning and knowledge discovery in databases* (pp. 148–163). http://dx.doi.org/10.1007/978-3-642-33486-3_10.

Molloy, T. L., Ford, J. J., & Perez, T. (2018). Finite-horizon inverse optimal control for discrete-time nonlinear systems. *Automatica*, *87*, 442–446. http://dx.doi.org/10.1016/j.automatica.2017.09.023.

Mombaur, K., & Clever, D. (2017). Inverse optimal control as a tool to understand human movement. In *Geometric and numerical foundations of movements* (pp. 163–186).

Mombaur, K. D., Olivier, A.-H., & Crétual, A. (2013). Forward and inverse optimal control of bipedal running. In *Modeling, simulation and optimization of bipedal walking* (pp. 165–179). http://dx.doi.org/10.1007/978-3-642-36368-9_13.

Mombaur, K., Truong, A., & Laumond, J.-P. (2010). From human to humanoid locomotion—an inverse optimal control approach. *Autonomous Robots*, *28*(3), 369–383. http://dx.doi.org/10.1007/s10514-009-9170-7.

Mori, T., Howard, M., & Vijayakumar, S. (2011). Model-free apprenticeship learning for transfer of human impedance behaviour. In *IEEE/RAS international conference on humanoid robots* (pp. 239–246).

Moylan, P., & Anderson, B. (1973). Nonlinear regulator theory and an inverse optimal control problem. *IEEE Transactions on Automatic Control*, *18*(5), 460–465. http://dx.doi.org/10.1109/TAC.1973.1100365.

Muelling, K., Boularias, A., Mohler, B., Schölkopf, B., & Peters, J. (2014). Learning strategies in table tennis using inverse reinforcement learning. *Biological Cybernetics*, *108*(5), 603–619. http://dx.doi.org/10.1007/s00422-014-0599-1.

Munzer, T., Piot, B., Geist, M., Pietquin, O., & Lopes, M. (2015). Inverse reinforcement learning in relational domains. In *International joint conferences on artificial intelligence*.

Neu, G., & Szepesvári, C. (2009). Training parsers by inverse reinforcement learning. *Machine Learning*, *77*(2–3), 303–337.

Neu, G., & Szepesvari, C. (2012). Apprenticeship learning using inverse reinforcement learning and gradient methods. arXiv:1206.5264, arXiv:1206.5264 [cs.LG].

Ng, A. Y., & Russell, S. J. (2000). Algorithms for inverse reinforcement learning. In *International conference on machine learning* (pp. 663–670).

Nguyen, Q. P., Low, B. K. H., & Jaillet, P. (2015). Inverse reinforcement learning with locally consistent reward functions. *Advances in Neural Information Processing Systems*, *28*, 1747–1755.

Oguz, O. S., Zhou, Z., Glasauer, S., & Wollherr, D. (2018). An inverse optimal control approach to explain human arm reaching control based on multiple internal models. *Scientific Reports*, *8*(1), 5583. http://dx.doi.org/10.1038/s41598-018-23792-7.

Okal, B., & Arras, K. O. (2016). Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning. In *IEEE international conference on robotics and automation* (pp. 2889–2895). http://dx.doi.org/10.1109/ICRA.2016.7487452.

Panchea, A. M. (2015). *Inverse optimal control for redundant systems of biological motion* (Ph.D. thesis), Orléans University.

Panchea, A. M., Miossec, S., Buttelli, O., Fraisse, P., Van Hamme, A., Welter, M.-L., & Ramdani, N. (2017). Gait analysis using optimality criteria imputed from human data. In *IFAC world congress (vol. 50)* (pp. 13510–13515). http://dx.doi.org/10.1016/j.ifacol.2017.08.2340.

Panchea, A. M., Ramdani, N., Bonnet, V., & Fraisse, P. (2018). Human arm motion analysis based on the inverse optimization approach. In *IEEE international conference on biomedical robotics and biomechatronics* (pp. 1005–1010). http://dx.doi.org/10.1109/BIOROB.2018.8488045.

Papadopoulos, A. V., Bascetta, L., & Ferretti, G. (2016). Generation of human walking paths. *Autonomous Robots*, *40*(1), 59–75. http://dx.doi.org/10.1007/s10514-015-9443-2.

Park, T., & Levine, S. (2013). Inverse optimal control for humanoid locomotion. In *Robotics science and systems workshop on inverse optimal control and robotic learning from demonstration* (pp. 4887–4892).

Park, D., Noseworthy, M., Paul, R., Roy, S., & Roy, N. (2020). Inferring task goals and constraints using Bayesian nonparametric inverse reinforcement learning. In *Conference on robot learning (vol. 100)* (pp. 1005–1014).

Park, J., Zatsiorsky, V. M., & Latash, M. L. (2011). Finger coordination under artificial changes in finger strength feedback: A study using analytical inverse optimization. *Journal of Motor Behavior*.

Pfeiffer, M., Schwesinger, U., Sommer, H., Galceran, E., & Siegwart, R. (2016). Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 2096–2101). http://dx.doi.org/10.1109/IROS.2016.7759329.

Priess, M. C., Conway, R., Choi, J., Popovich, J. M., & Radcliffe, C. (2014). Solutions to the inverse LQR problem with application to biological systems analysis. *IEEE Transactions on Control Systems Technology*, *23*(2), 770–777. http://dx.doi.org/10.1109/TCST.2014.2343935.

Puydupin-Jamin, A.-S., Johnson, M., & Bretl, T. (2012). A convex approach to inverse optimal control and its application to modeling human locomotion. In *IEEE international conference on robotics and automation* (pp. 531–536). http://dx.doi.org/10.1109/ICRA.2012.6225317.

Qiao, Q., & Beling, P. A. (2011). Inverse reinforcement learning with Gaussian process. In *American control conference* (pp. 113–118). http://dx.doi.org/10.1109/ACC.2011.5990948.

Ramachandran, D. (2007). Bayesian inverse reinforcement learning. In *International joint conference on artificial intelligence* (pp. 2586–2591).

Ratliff, N., Bagnell, J. A., & Srinivasa, S. S. (2007). Imitation learning for locomotion and manipulation. In *IEEE/RAS international conference on humanoid robots* (pp. 392–397). http://dx.doi.org/10.1109/ICHR.2007.4813899.

Ratliff, N. D., Bagnell, J. A., & Zinkevich, M. A. (2006). Maximum margin planning. In *International conference on machine learning* (pp. 729–736). http://dx.doi.org/10.1145/1143844.1143936.

Ratliff, N., Bradley, D., Bagnell, J. A., & Chestnutt, J. (2006). Boosting structured prediction for imitation learning. In *International conference on neural information processing systems* (pp. 1153–1160).

Ravichandar, H., Polydoros, A. S., Chernova, S., & Billard, A. (2020). Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*.

Rebula, J. R., Schaal, S., Finley, J., & Righetti, L. (2019). A robustness analysis of inverse optimal control of bipedal walking. *IEEE Robotics and Automation Letters*, *4*(4), 4531–4538. http://dx.doi.org/10.1109/LRA.2019.2933766.

Reddy, T. S., Gopikrishna, V., Zaruba, G., & Huber, M. (2012). Inverse reinforcement learning for decentralized non-cooperative multiagent systems. In *2012 IEEE international conference on systems, man, and cybernetics* (pp. 1930–1935). http://dx.doi.org/10.1109/ICSMC.2012.6378020.

Rothkopf, C. A., & Ballard, D. H. (2013). Modular inverse reinforcement learning for visuomotor behavior. *Biological Cybernetics*, *107*(4), 477–490. http://dx.doi.org/10.1007/s00422-013-0562-6.

Rothkopf, C. A., & Dimitrakakis, C. (2011). Preference elicitation and inverse reinforcement learning. In *Lecture notes in computer science*, *Machine learning and knowledge discovery in databases* (pp. 34–48). http://dx.doi.org/10.1007/978-3-642-23808-6_3.

Schaal, S. (1997). Learning from demonstration. In *Advances in neural information processing systems (vol. 9)* (pp. 1040–1046).

Sermanet, P., Xu, K., & Levine, S. (2016). Unsupervised perceptual rewards for imitation learning. arXiv:1612.06699, arXiv:1612.06699 [cs.CV].

Shiarlis, K., Messias, J., & Whiteson, S. A. (2016). Inverse reinforcement learning from failure.

Shukla, N., He, Y., Chen, F., & Zhu, S.-C. (2017). Learning human utility from video demonstrations for deductive planning in robotics. In *Conference on robot learning* (pp. 448–457).

Silver, D., Bagnell, J. A., & Stentz, A. (2010). Learning from demonstration for autonomous navigation in complex unstructured terrain. *International Journal of Robotics Research*, *29*(12), 1565–1592. http://dx.doi.org/10.1177/0278364910369715.

Syed, U., Bowling, M., & Schapire, R. E. (2008). Apprenticeship learning using linear programming. In *ACM international conference on machine learning* (pp. 1032–1039). http://dx.doi.org/10.1145/1390156.1390286.

Syed, U., & Schapire, R. E. (2007). A game-theoretic approach to apprenticeship learning. In *ACM international conference on neural information processing systems* (pp. 1449–1456).

Sylla, N., Bonnet, V., Venture, G., Armande, N., & Fraisse, P. (2014). Human arm optimal motion analysis in industrial screwing task. In *IEEE EMBS/RAS international conference on biomedical robotics and biomechatronics* (pp. 964–969). http://dx.doi.org/10.1109/BIOROB.2014.6913905.

Terekhov, A. V., Pesin, Y. B., Niu, X., Latash, M. L., & Zatsiorsky, V. M. (2010). An analytical approach to the problem of inverse optimization with additive objective functions: An application to human prehension. *Journal of Mathematical Biology*, *61*(3), 423–453. http://dx.doi.org/10.1007/s00285-009-0306-3.

Terekhov, A. V., & Zatsiorsky, V. M. (2011). Analytical and numerical analysis of inverse optimization problems: Conditions of uniqueness and computational methods. *Biological Cybernetics*, *104*(1–2), 75–93. http://dx.doi.org/10.1007/s00422-011-0421-2.

Theodorou, E., Buchli, J., & Schaal, S. (2010). A generalized path integral control approach to reinforcement learning. *Journal of Machine Learning Research*, *11*, 3137–3181. http://dx.doi.org/10.5555/1756006.1953033.

Unni, M. P., Sinha, A., Chakravarty, K., Chatterjee, D., & Das, A. (2017). Neuromechanical cost functionals governing motor control for early screening of motor disorders. *Frontiers in Bioengineering and Biotechnology*, *5*, http://dx.doi.org/10.3389/fbioe.2017.00078.

Vogel, A., Ramachandran, D., Gupta, R., & Raux, A. (2012). Improving hybrid vehicle fuel efficiency using inverse reinforcement learning. In *AAAI conference on artificial intelligence*.

Westermann, K., Lin, J. F.-S., & Kulić, D. (2020). Inverse optimal control with time-varying objectives: Application to human jumping movement analysis. *Scientific Reports*, *10*(1), 11174. http://dx.doi.org/10.1038/s41598-020-67901-x.

Wulfmeier, M., Ondruska, P., & Posner, I. (2015). Maximum entropy deep inverse reinforcement learning. arXiv:1507.04888, arXiv:1507.04888 [cs].

Xia, C., & El Kamel, A. (2016). Neural inverse reinforcement learning in autonomous navigation. *Robotics and Autonomous Systems*, *84*, 1–14. http://dx.doi.org/10.1016/j.robot.2016.06.003.

Yao, K., & Billard, A. (2020). An inverse optimization approach to understand human acquisition of kinematic coordination in bimanual fine manipulation tasks. *Biological Cybernetics*, *114*(1), 63–82.

Yin, H., Alves-Oliveira, P., Melo, F. S., Billard, A., & Paiva, A. (2016). Synthesizing robotic handwriting motion by learning from human demonstrations. In *International joint conference on artificial intelligence, no. CONF*. (pp. 3530–3537).

Zhang, H., Li, Y., & Hu, X. (2019). Inverse optimal control for finite-horizon discrete-time linear quadratic regulator under noisy output. In *IEEE conference on decision and control* (pp. 6663–6668). http://dx.doi.org/10.1109/CDC40024.2019.9029795.

Zhang, H., Umenberger, J., & Hu, X. (2018). Inverse quadratic optimal control for discrete-time linear systems. arXiv:1810.12590, arXiv:1810.12590 [math].

Zheng, J., Liu, S., & Ni, L. M. (2014). Robust Bayesian inverse reinforcement learning with sparse behavior noise. In *AAAI conference on artificial intelligence* (pp. 2198–2205).

Zhifei, S., & Er, M. J. (2012). A survey of inverse reinforcement learning techniques. In Y. Gao, J. Peters, & A. Tsourdos (Eds.), *International Journal of Intelligent Computing and Cybernetics*, *5*(3), 293–311. http://dx.doi.org/10.1108/17563781211255862.

Ziebart, B. D., Maas, A., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *National conference on artificial intelligence* (p. 6).

Ziebart, B. D., Maas, A. L., Dey, A. K., & Bagnell, J. A. (2008). Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *ACM international conference on ubiquitous computing* (pp. 322–331).

Ziebart, B. D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., & Srinivasa, S. (2009). Planning-based prediction for pedestrians. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 3931–3936). http://dx.doi.org/10.1109/IROS.2009.5354147.