Artificial Intelligence Review (2022) 55:4307–4346
https://doi.org/10.1007/s10462-021-10108-x

A survey of inverse reinforcement learning

Stephen Adams[1] · **Tyler Cody**[1] · **Peter A. Beling**[1]

Published online: 8 February 2022

Abstract
Learning from demonstration, or imitation learning, is the process of learning to act in an environment from examples provided by a teacher. Inverse reinforcement learning (IRL) is a specific form of learning from demonstration that attempts to estimate the reward function of a Markov decision process from examples provided by the teacher. The reward function is often considered the most succinct description of a task. In simple applications, the reward function may be known or easily derived from properties of the system and hard coded into the learning process. However, in complex applications, this may not be possible, and it may be easier to learn the reward function by observing the actions of the teacher. This paper provides a comprehensive survey of the literature on IRL. This survey outlines the differences between IRL and two similar methods - apprenticeship learning and inverse optimal control. Further, this survey organizes the IRL literature based on the principal method, describes applications of IRL algorithms, and provides areas of future research.

Keywords Reinforcement learning · Inverse reinforcement learning · Inverse optimal control · Apprenticeship learning · Learning from demonstration

1 Introduction

Learning from demonstration, or imitation learning, is the process of learning to act in an environment from examples provided by a teacher. In computer science and robotics, learning from demonstration is often utilized to teach a computer or robot a control policy (Argall et al. 2009; Hussein et al. 2017), which maps a state to a particular action. In simple applications, a policy may be able to be "hard coded" by a developer. However, this method has the significant limitation that the developer must anticipate all possible future states and know the correct action. A considerable amount of time and resources must be invested by the developer to completely understand the system and to be able to provide this information. Policies can also be learned by interacting with a system or environment. Reinforcement learning (RL) (Sutton and Barto 1998) is one method for learning a control policy through interaction and has shown considerable ability to learn optimal policies in

✉ Stephen Adams
scadams21@vt.edu

[1] Hume Center for National Security and Technology, Virginia Tech, Arlington, VA 22203, USA

complex environments (Duan et al. 2016; Mnih et al. 2015; Zhu et al. 2017). However, RL often requires a large number of interactions with the environment in order to learn the optimal policy. Learning from demonstration alleviates these drawbacks by learning from a teacher.

Inverse reinforcement learning (IRL) is a specific form of learning from demonstration that attempts to learn the reward function of the teacher providing the examples. The underlying sequential control model when utilizing IRL is a Markov decision process (MDP). An MDP is composed of states, actions, rewards, a transition function, and a discount rate. When all of the components of the MDP are known and the number of states and actions is relatively small, dynamic programming can be used to find the optimal policy. When components of the MDP are unknown or the state and action spaces are so large that dynamic programming is unfeasible, RL can be applied to learn the optimal policy. IRL attempts to learn the structure or parameters of the reward function provided expert demonstrations.

The reward function is often considered the most succinct description of a task. If the reward function is known, then methods such as dynamic programming or RL can be used to learn the policy. In applications where the ultimate goal is to win, such as games, the reward function may be obvious and easy to encode into a learning algorithm. In other applications, such as robotic control, the reward function may not be obvious. For example, consider the task of developing a robotic arm to pick up an object. If the control policy were to be hard coded by the designer, each actuator action of the arm must be designed and coded. If the object were able to be moved or not always placed in the same spot, this process would need to be repeated for all possible locations, and the robot would need to recognize the position of the object. RL could be implemented to teach the robotic arm to pick up the object but the reward function would be sparse, i.e., the robot would receive a reward signal if the object was successfully grasped and nothing otherwise. Sparse reward functions can often lead to long training times due to the lack of feedback. IRL would allow an expert to demonstrate the successful task and then a highly parameterized reward function could be transferred to the robot for optimization.

IRL is one method of learning from demonstration. Apprenticeship learning and inverse optimal control (IOC) are two similar forms of learning from demonstration. While IRL attempts to learn the reward function, apprenticeship learning attempts to directly learn the policy. In some apprenticeship learning algorithms, IRL is used as an intermediate step, i.e., the reward function is learned and then the optimal policy is learned using the reward function. This survey differentiates the two by the ultimate goal of the algorithm. IRL has the added advantage that the reward function can be transferred to new environments with different dynamics while the policy learned by apprenticeship learning is dependent upon a stable environment in order to maintain optimality. IOC is essentially the continuous version of IRL where the primary difference between the two methods is the form of the underlying sequential control model. IRL assumes an MDP while IOC assumes a state-space model. Other differences between these methods and IRL are discussed in a later section.

This survey provides an overview of existing IRL methods. A previous survey of IRL techniques was published in 2012 by Zhifei and Meng Joo (2012). In the time since then, there have been several advances in the field of IRL that justify a new and updated survey. In addition, the primary contribution of this survey is to organize and categorize the rapidly growing literature on IRL. This survey divides the literature on IRL methods into two broad categories: algorithms and extensions of the basic IRL paradigm. A second contribution is identifying that while the former category is extremely well developed, the latter category needs more development in order for IRL to be widely applicable. Furthermore, this survey divides the

literature on IRL applications into three broad categories based on the intended use of the learned reward function: learn to mimic the expert, learn to better interact with other systems, and learn about the system under study. A third contribution of this survey is relating the IRL concept to other similar concepts and identifying that the fields of IRL and IOC have essentially merged and become interchangeable. Overall, IRL has undergone significant foundational and applied development since previous surveys have been published, and, with these developments, other areas of machine learning like deep learning and transfer learning are primed for synthesis with IRL. This survey provides an up-to-date technical overview of IRL that can readily serve as a basis for its extension by practitioners and researchers alike into these new domains.

Section 2 provides background information on MDPs, the forward RL problem, and other topics related to IRL. Section 3 outlines IRL methods, and Sect. 4 outlines applications of IRL methods. Section 5 provides provides conclusions, limitations of current IRL techniques, and future research directions.

2 Background

This section outlines background information on Markov decision processes (MDPs) and topics related to IRL – RL, apprenticeship learning, and IOC.

2.1 Markov decision processes and reinforcement learning

An MDP is a model for sequential decision making (Puterman 2014; Sutton and Barto 1998). The canonical MDP is defined by the tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P^a_{s,s'}, \gamma, R\}$ where:

- $\mathcal{S}$ represents a finite set of I states. A particular state at time t is denoted using S_t.
- $\mathcal{A}$ represents a finite set of K actions. A particular action at time t is denoted using A_t.
- $P^a_{s,s'} = \mathbb{P}(s'|s, a)$ represents the transition function.
- γ represents the discount factor and can be interpreted as quantifying the relative importance of short-term and long-term rewards.
- R represents the reward function.

The reward received for taking action a in state s is denoted as $r(s, a)$. If the reward is solely dependent upon the state, the reward function can be written as $r(s)$. Figure 1 displays a graphical model of an MDP. The sum of discounted future rewards at time t is written as $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$. A policy π maps states to actions. The probability of taking action a in state s under policy π is denoted as $\pi(a|s)$. The objective of an agent using an MDP for their sequential decision making process is to find a policy that maximizes expected discounted future reward.

A major component of MDPs is the value of being in a particular state and following a given policy. This is defined as

$$\begin{aligned} V^\pi(s) &= \mathbb{E}[G_t|S_t = s] \\ &= \sum_{a\in\mathcal{A}} \pi(a|s)\left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P^a_{s,s'} V^\pi(s')\right), \end{aligned} \tag{1}$$

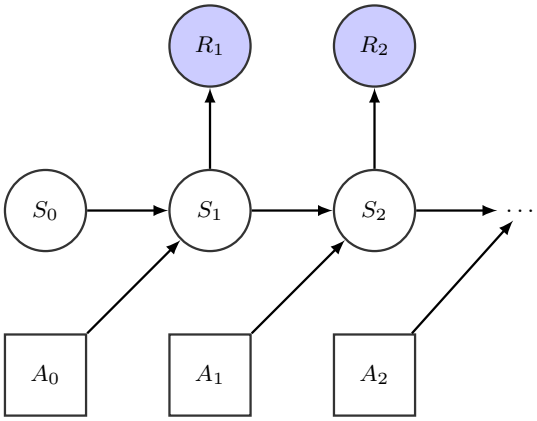

Fig. 1 Graphical model for an MDP. Open circles represent states, squares represent actions, and filled circles represent rewards

where $\mathbb{E}[\cdot]$ represents the expectation. Similarly, the value of a state-action pair is defined as

$$\begin{aligned} Q^{\pi}(s,a) &= \mathbb{E}[G_t | S_t = s, A_t = a] \\ &= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^{a}_{s,s'} \sum_{a' \in \mathcal{A}} \pi(a'|s') Q^{\pi}(s',a'). \end{aligned} \tag{2}$$

An optimal policy π^* can be found by maximizing either V or Q. The optimal state value function is defined as $V^*(s) \doteq \max_{\pi} V^{\pi}(s)$, and the optimal state-action value is defined as $Q^*(s,a) \doteq \max_{\pi} Q^{\pi}(s,a)$. The optimal state value function and state-action value function can be found using

$$V^*(s) = \max_{a} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^{a}_{s,s'} V^*(s') \right), \tag{3}$$

and

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^{a}_{s,s'} \max_{a'} Q^*(s',a'). \tag{4}$$

If the size of the MDP is relatively small and the components are known, then dynamic programming can be used to find π^*. However, if this is not the case, other methods, such as RL, are used to find π^*.

RL (Sutton and Barto 1998) estimates an optimal policy for an MDP by having an agent interact with an environment. Generally, RL is utilized when components of the MDP are not available or difficult to estimate. For example, the transition function may be difficult to estimate due to the size of the state space or limits to the amount of available transition data. However, if the state space is large, iterative methods such as dynamic programming cannot efficiently solve for an optimal policy even if all the components of the MDP are available.

During the training phase of RL, the agent selects an action for the current state using an exploration policy. A reward is received based on the state-action pair, and then the system transitions to the next state. The agent uses the collected reward to update its expectation of future reward, e.g. the Q-value. There are several algorithms for updating the Q-values including Monte Carlo learning, SARSA, and Q-learning. For

more information on RL, see (Arulkumaran et al. 2017; Garcıa and Fernández 2015; Ghavamzadeh et al. 2015).

2.2 Apprenticeship learning and inverse optimal control

The objective of apprenticeship learning is to learn a policy that mimics the unknown policy being executed by the expert providing the demonstrations. The primary difference between apprenticeship learning and IRL is the objective of the algorithm. IRL's objective is to recover the unknown reward function of the expert providing the demonstrations. IRL could be used as a subroutine in an apprenticeship learning algorithm. In this case, IRL would be used to recover the reward function and then the reward would be used to estimate a policy that mimics the expert's. In this sense, all IRL algorithms could be considered an apprenticeship learning algorithm because any of them could be used to estimate the expert's policy. However, not all apprenticeship learning algorithms can be considered IRL.

Pure apprenticeship learning algorithms omit the step of recovering the expert's reward function. One of the most popular techniques is to cast the apprenticeship learning problem as a supervised classification problem and learn a classifier that is used as the policy (Melo and Lopes 2010; Piot et al. 2014; Ratliff et al. 2007; Syed and Schapire 2010). Another popular approach is to learn probabilistic models, such as Gaussian mixture models or hidden Markov models, that produce trajectories that are similar to those provided by the expert (Calinon et al. 2010, 2007; Hussein et al. 2015). Active learning has also been incorporated into the apprenticeship learning framework (Cakmak and Thomaz 2011).

Apprenticeship learning has several advantages over IRL. First, by omitting the IRL step which often contains a subroutine that solves the forward RL problem, the algorithm generally converges to an optimal policy faster than IRL algorithms. Second, IRL is an ill-posed problem because there are numerous reward functions that will produce similar behavior. This makes learning the "correct" reward function of the expert difficult if not impossible. The objective of apprenticeship learning is to find a policy that mimics the behavior of the expert and is thus a less restrictive problem. The main advantage of IRL over apprenticeship learning is that the reward function is the most succinct representation of the task—more succinct than the policy. Furthermore, the learned policy in apprenticeship learning is dependent upon the dynamics of the environment. If the dynamics of the environment change, the learned policy may no longer be optimal and the apprenticeship learning algorithm would need to be rerun with new demonstrations. The reward function learned in IRL is independent of the environment's dynamics and a new policy could be learned after the environment changes using dynamic programming or RL.

Piot et al. (2013) compare the empirical performance of a policy following IRL's learned reward structure with a policy learned using apprenticeship learning. They note that estimating the reward is difficult and found that it often serves to add error in cases where apprenticeship learning performs well. However, for simpler reward structures where the reward is less informative, e.g. the reward is only state-dependent or sparse, IRL outperforms apprenticeship learning. Moreover, when the dynamics of the underlying MDP are perturbed, the advantage of IRL is made clear; apprenticeship learning is highly dependent on the dynamics used in training while IRL is not. In short, for MDPs with complicated reward structures and static dynamics, Piot, Geist, and Pietquin found that estimating the reward deteriorates performance, however, when the reward is simple or the dynamics change, IRL clearly outperforms apprenticeship learning. Later, they showed that even for

complicated reward structures, IRL can outperform apprenticeship learning when a good representation of the reward and transition function are provided (Piot et al. 2017).

IOC is another area of learning from demonstration that is similar to IRL. While there is a clear distinction between apprenticeship learning and IRL, the divide between IOC and IRL is less clear. In fact, we posit that while the two (IOC and IRL) began as separate subjects, as the research progressed they merged and are today relatively indistinguishable. The early work on control-theoretic IOC was inspired by the the concept of inverse optimality first studied by Kalman (1964). Similar to IRL, control-theoretic IOC estimates a cost function for a sequential control problem. There are two primary differences between control-theoretic IOC and IRL. The first is a focus on stability as opposed to optimality. The second is the use of a control-Lyapunov function, which is treated as an optimal value function, in place of demonstrations provided by an expert. The latter is the most striking difference between the two subjects, and the primary delineation as we see it. The first control-theoretic IOC algorithms were implemented due to difficulties in solving the Hamilton-Jacobi-Bellman equations associated with control problems (Krstic and Li 1998; Krstic and Tsiotras 1999). Several follow up studies outline the field (Cai and Han 2005; Kanazawa et al. 2009; Luo et al. 2005; Nakamura et al. 2011; Ornelas et al. 2010, 2011; Ornelas-Tellez et al. 2012; Ruiz-Cruz et al. 2013). Aghasadeghi et al. (2012) take the control-theoretic IOC concepts and use observed trajectories as a surrogate for the control policy. From this point forward in the literature, IOC and IRL are essentially interchangeable.

3 Survey of IRL methods

Let $\mathcal{M} \setminus R$ represent an MDP with an unknown reward function. The objective of IRL is to learn or estimate the reward function given demonstrations of behavior in the environment. In the early IRL work, the demonstrations are usually assumed to be provided by an expert. However in later work, this assumption is relaxed. The demonstrations come in the form of trajectories composed of state-action pairs $\mathcal{T}_m = \{(s_1, a_1), ..., (s_{T_m}, a_{T_m})\}$, where $m = 1, ..., M$ represents the m^{th} expert trajectory, and T_m represents the number of time steps in the m^{th} trajectory. The set of all trajectories is denoted as $\mathcal{T}$.

Another assumption of IRL is that the expert is following an optimal policy with respect to their reward function R_E (this assumption also is relaxed in some of the more recent literature). An IRL algorithm attempts to recover the expert's reward function. The concept of IRL was first proposed by Russell (1998). He defines the IRL problem as determining the reward function optimized by an agent given 1) the agent's behavior over time and in varying circumstances, 2) measurements of the information given to the agent, and 3) a model of the environment. Russel does not propose an IRL algorithm but outlines several areas of future research for the field.

The IRL methods outlined in this section are divided into two categories. The first category is algorithms and represents the bulk of the IRL literature. The second category represents extensions of IRL. This category represents areas of research that go beyond the basic IRL paradigm and generally relax some of the assumptions of the problem formulation, e.g., multi-agent IRL. The former category is a well developed area of research while the latter represents areas that must be further developed in order for IRL to be widely applicable.

3.1 Algorithms

This section categorizes IRL methods based on the algorithm used to estimate the reward function (summarized in Table 1). The first category is max margin methods. The max-margin method was the first class of IRL algorithms and attempts to estimate rewards that maximize the margin between the optimal policy or value function and all other policies or value functions. IRL methods that match feature expectation are also categorized as max-margin methods because they share two traits. First, most of these methods assume that the reward is a linear combination of features and attempt to estimate the weights associated with the features. Second, feature expectation methods often maximize a slack variable in order to drive the estimated feature expectation to the empirical feature expectation.

The second category of IRL algorithms is Bayesian methods, which have two defining characteristics. The first is that the optimization routine maximizes the posterior distribution of the reward. The second is the use of prior distributions on the reward in order to define the posterior distribution. One advantage of Bayesian methods is the ability to convey prior information about the reward to the learning algorithm through the prior distribution. A second advantage is the ability for the Bayesian methods to account for complex behavior by modeling the reward probabilistically as a mixture of multiple reward functions.

The third category of IRL algorithms is maximum entropy methods. These methods are defined by using maximum entropy in the optimization routine to estimate the reward function. Maximum entropy methods are easily extended to the continuous space and can address sub-optimal expert demonstrations.

There are several IRL methods that do not fit nicely into these categories. Miscellaneous algorithms are outlined in the final subsection of this section. While there are some broad advantages to using a particular category of algorithm over another, as outlined, because of each method's nuance, intra-category differences must be considered, i.e., the strength of broad, category-wise comparisons beyond scholarly distinctions is limited. Instead, advantages and limitations of various methods are noted along the way, typically in following with historical development, in reference to IRL generally or other IRL methods in particular.

3.1.1 Maximum margin methods

The first paper published on IRL that presents methods for estimating the reward was written by Ng and Russell (2000) and outlines three max margin IRL algorithms. The first two assume that the expert's policy and the transition function are known. The third algorithm relaxes the assumption that the expert's policy is known but still requires a transition function. Ng and Russel propose that the reward function provides the most succinct description of an agent's behavior. More general methods, such as imitation learning or apprenticeship learning, attempt to learn the agent's policy from the set of demonstrations. However, the policy is dependent upon the system dynamics. If the dynamics change, the policy must change. By learning the reward function directly, an optimal policy for any transition function is learned.

Table 1 Inverse reinforcement learning algorithms

Max margin	Bayesian	Max entropy	Misc.
Ng and Russell (2000)	Ramachandran and Amir (2007)	Ziebart et al. (2008)	Ho and Ermon (2016)
Abbeel and Ng (2004)	Rothkopf and Ballard (2013)	Ziebart et al. (2009)	Hausman et al. (2017)
Abbeel et al. (2008)	Michini and How (2012)	Ziebart et al. (2012)	Henderson et al. (2018)
Syed and Schapire (2008)	Levine et al. (2011)	Aghasadeghi and Bretl (2011)	Hahn and Zoubir (2015)
Syed et al. (2008)	Choi and Kim (2011)	Kalakrishnan et al. (2013)	Doerr et al. (2015)
Valko et al. (2013)	Qiao and Beling (2011)	Kalakrishnan et al. (2010)	Chen et al. (2016)
Zhou and Li (2018)	Qiao and Beling (2013)	Levine and Koltun (2012)	Babes et al. (2011)
Neu and Szepesvári (2007)	Choi and Kim (2012)	Ziebart et al. (2013)	Nguyen et al. (201)
Ratliff et al. (2006)	Choi and Kim (2013)	Bloem and Bambos (2014)	Levine et al. (2010)
Ratliff et al. (2006)	Budhraja and Oates (2017)	Zhou et al. (2018)	Klein et al. (2011)
Ratliff et al. (2009)	Michini and How (2012)	Shiarlis et al. (2016)	Majumdar et al. (2017)
Zou et al. (2018)	Michini et al. (2013)	Audiffren et al. (2015)	Singh et al. (2018)
Boularias and Chaib-Draa (2013)	Michini et al. (2015)	Bogert et al. (2016)	Ratliff et al. (2009)
Choi and Kim (2011)	Šošic et al. (2018)	Bogert and Doshi (2017)	Dvijotham and Todorov (2010)
Chinaei and Chaib-Draa (2012)	Šošić et al. (2018)	Byravan et al. (2015)	
	Lee et al. (2016)	Shimosaka et al. (2017)	
	Zheng et al. (2014)	Wulfmeier et al. (2016)	
	Rothkopf and Dimitrakakis (2011)	Wulfmeier et al. (2017)	
	Dimitrakakis and Rothkopf (2011)	Chen et al. (2019)	
	Surana (2014)	Finn et al. (2016)	
	Brown and Niekum (2018)	Mendez et al. (2018)	
		Yu et al. (2019)	
		Ranchod et al. (2015)	
		Boularias et al. (2012)	

The first IRL algorithm proposed by Ng and Russell (2000) makes three strong assumptions: 1) the policy is known, 2) the state transition dynamics are known, and 3) the state space is finite. Their primary result for this algorithm is the constraint

$$(\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1}\boldsymbol{R} \succeq 0, \tag{5}$$

where $\boldsymbol{P}_a$ is the transition matrix for action a, a_1 is the action defined by the policy $\pi(s) \equiv a_1$, and $\boldsymbol{R}$ is a vector of rewards. This constraint defines all reward functions that are possible solutions to the IRL problem. However, the solution set contains $\boldsymbol{R} = 0$ as well as numerous other solutions. To overcome this limitation, Ng and Russel propose a linear programming solution with a penalty term. This optimization problem is

$$\begin{aligned} \max \sum_{i=1}^{I} \min_{a \in \{a_2,\ldots,a_K\}} \quad & \{(\boldsymbol{P}_{a_1}(i) - \boldsymbol{P}_a(i))(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1}\boldsymbol{R}\} - \lambda ||\boldsymbol{R}||_1 \\ \text{s.t.} \quad & (\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1}\boldsymbol{R} \succeq 0 \quad \forall a \in \mathcal{A} \setminus a_1 \\ & |\boldsymbol{R}_i| \leq R_{\max}, \quad i = 1...I, \end{aligned} \tag{6}$$

where $\boldsymbol{P}_a(i)$ denotes the i^{th} row of $\boldsymbol{P}_a$, and $\mathcal{A} \setminus a_1$ represents all the actions in $\mathcal{A}$ except a_1.

The second algorithm proposed by Ng and Russel assumes an infinite state space. A linear approximation is used for the reward function

$$R(s) = \alpha_1 \phi_1(s) + \alpha_2 \phi_2(s) + ... + \alpha_D \phi_D(s), \tag{7}$$

where $\phi_1, ..., \phi_D$ are basis functions that are fixed and known. In this formulation, the weight parameters α are assumed to be unknown and estimated by the IRL algorithm. The value function for a given policy can now be written as

$$V^{\pi}(s) = \alpha_1 V_1^{\pi}(s) + \alpha_2 V_2^{\pi}(s) + ... + \alpha_D V_D^{\pi}(s). \tag{8}$$

For a policy to be optimal under the infinite state assumption, the following constraint must hold

$$\mathbb{E}_{s' \sim P_{sa_1}}[V^{\pi}(s')] \geq \mathbb{E}_{s' \sim P_{sa}}[V^{\pi}(s')] \quad \forall s, \text{ and } \forall a \in \mathcal{A} \setminus a_1. \tag{9}$$

The linear program for this algorithm is

$$\begin{aligned} \max \sum_{s \in S_0} \min_{a \in \{a_2,\ldots,a_K\}} \quad & \{p(\mathbb{E}_{s' \sim P_{sa_1}}[V^{\pi}(s')] - \mathbb{E}_{s' \sim P_{sa}}[V^{\pi}(s')])\} \\ \text{s.t.} \quad & |\alpha_d| \leq 1, \quad d = 1, ..., D, \end{aligned} \tag{10}$$

where S_0 is a subsample of states in the infinite state space, and $p(x) = x$ if $x \geq 0$ and $p(x) = 2x$ otherwise.

The third algorithm proposed by Ng and Russel addresses the more realistic case where the reward function is estimated from sample trajectories. Let $\mathcal{D}$ be a fixed initial state distribution. The objective of the IRL algorithm is to find the reward such that the unknown policy π maximizes $\mathbb{E}_{s_0 \sim \mathcal{D}}[V^{\pi}(s_0)]$. There are two assumptions for this algorithm. The first is that the optimal policy can be estimated. The second is the ability to simulate trajectories from the MDP. The second assumption allows for the transition function to be estimated from the sampled trajectories and does not need to be known a

priori. The sampled trajectories are used to estimate $\hat{V}_d^{\pi}(s_0)$ for $d = 1, ..., D$. Given a set of policies $\{\pi_1, ..., \pi_I\}$, the optimization problem becomes

$$\begin{aligned} &\text{maximize} \sum_{i=1}^{I} p\left(\hat{V}^{\pi^*}(s_0) - \hat{V}^{\pi_i}(s_0)\right) \\ &\text{s.t.} |\alpha_d| \leq 1, \quad d = 1, ..., D. \end{aligned} \tag{11}$$

This optimization problem returns a new set of α's and thus a new reward R. A new optimal policy is found using R, and the new policy is added to the set of policies. The algorithm is then repeated for a set number of iterations or until some measure of convergence is met.

Abbeel and Ng (2004) develop an IRL algorithm for apprenticeship learning that finds a policy that mimics that of the expert's by first estimating the reward function and then finding an optimal policy for the estimated reward function. The proposed IRL algorithm assumes that the reward can be expressed as a linear combination of known features and that the objective of the algorithm is to find the weights corresponding to each feature. Given this assumption, the value of a policy can be written as $\mathbb{E}_{s_0 \sim \mathcal{D}}[V^{\pi}(s_0)] = \alpha\mu(\pi)$. The apprenticeship learning algorithm attempts to find a policy $\widetilde{\pi}$ that minimizes $||\mu(\widetilde{\pi}) - \mu_E||$, where μ_E is the feature expectation of the expert. The expert's feature expectation can be estimated from the supplied trajectories. This optimization problem is written as

$$\begin{aligned} \max_{\alpha,t} \quad & t \\ \text{s.t.} \quad & \alpha^{\intercal}\mu_E \geq \alpha^{\intercal}\mu^{(j)} + t, \quad j = 0, ..., i-1 \\ & ||\alpha||_2 \leq 1, \end{aligned} \tag{12}$$

where $\mu^{(j)}$ represents the feature expectation under the j^{th} candidate policy. Abbeel and Ng also provide a simpler algorithm which they call the projection method. In a following study, Abbeel et al. (2008) adapt the max margin formulation to a path planning setting.

Syed and Schapire (2008) point out that a drawback to the method proposed by Abbeel and Ng (2004), and a limitation of many IRL algorithms, is that the performance of the agent learning the policy is limited to the quality of the expert. If the expert is not optimal, the learned policy will also be sub-optimal. In response, Syed and Schapire propose a game-theoretic approach to apprenticeship learning and IRL called multiplicative weights for apprenticeship learning (MWAL). The apprenticeship learning problem is posed as a two-player zero sum game. The goal of the learning agent (apprentice) is to maximize performance with respect to the expert even if the expert is playing sub-optimally. A key aspect of MWAL is the ability to impart prior knowledge to the learning agent about the importance of features. Syed, Bowling, and Schapire later use a linear programming approach to modify the MWAL algorithm so that it outputs a stationary policy (Syed et al. 2008).

As pointed out by Syed and Schapire, the reliance on expert demonstrations is a significant limitation in early IRL research. Following Syed and Schapire, several studies have addressed this limitation. Valko et al. (2013) extend the max margin formulation to the semi-supervised case where trajectories provided by an expert are assumed to be the labeled examples and trajectories provided by a non-expert are assumed to be the unlabeled examples. Zhou and Li (2018) develop a safety-aware version of max margin apprenticeship learning. A policy learned through apprenticeship learning can lead an agent into an unsafe state because an expert only supplies positive examples. Formal verification is incorporated into the learning process. If an estimated policy violates a

safety specification, a negative example is generated. The negative examples are incorporated into the objective function, and a safety-aware policy is generated.

Another limitation of the algorithm proposed by Abbeel and Ng (2004) is sensitivity to the scaling of the features. In response, Neu and Szepesvári (2007) use gradient optimization techniques and IRL concepts to tackle the apprenticeship learning problem. They formulate the following objective function

$$J(\alpha) = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \mu_E(s)(\pi(a|s) - \pi_E(a|s))^2, \tag{13}$$

and minimize $J(\alpha)$ using the gradient

$$\frac{\partial \pi_\alpha}{\partial \alpha_k} = \pi_\alpha(a|s)\beta\left(\frac{Q^*_\alpha(s,a)}{\partial \alpha_k} - \sum_{a' \in \mathcal{A}} \pi_\alpha(a'|s)\frac{Q^*_\alpha(s,a')}{\partial \alpha_k}\right). \tag{14}$$

Ratliff et al. (2006) propose maximum margin planning where the objective is to find a linear mapping of features to rewards so that the policy estimated from the reward function is "close" to the behavior in the demonstrations. They outline a quadratic program that incorporates a slack variable ζ that allows for violations of the constraint similar to the solution to support vector machines. The quadratic program is

$$\begin{aligned} &\min_{\alpha, \zeta_m} && \frac{1}{2}||\alpha||^2 + \frac{\gamma}{M}\sum_{m=1}^{M} \beta_m \zeta_m \\ &\text{s.t.} && \alpha^\intercal f_m(y_m) + \zeta_m \geq \max_{y \in \mathcal{Y}_m} \alpha^\intercal f_m(y) + \mathcal{L}_m(y), \quad \forall m, \end{aligned} \tag{15}$$

where y is the desired trajectory, $f(\cdot)$ is the expected feature count, $\mathcal{L}(\cdot)$ is the loss function, and γ and β are hyperparameters. If both the expected feature count and loss function are linear in the state-action space, the quadratic program can be transformed into a compact quadratic program. Ratliff *et al.* outline an efficient optimization algorithm and provide an algorithm for an online setting. The concept of maximum margin planning is later extended and generalized (Ratliff et al. 2006, 2009). Zou et al. (2018) build upon maximum margin planning and utilize deep learning to model human driving behavior in two capacities. The first is to extract relevant feature information from raw sensor input to represent the state information. The second is to estimate a policy given the current estimate of the reward function.

One of the limitations of max margin planning, and other IRL methods that rely on feature expectation, is that they require enough data (expert trajectories) to adequately estimate the expectation. Boularias and Chaib-Draa (2013) propose two bootstrapping methods that can improve the calculation of the feature expectation when only a small number of trajectories are provided by the expert. They demonstrate these methods on max margin planning and linear programming apprenticeship learning.

Most IRL algorithms assume that the state information is readily available to the algorithm. However, in many real-world applications, state information is only partially observable. For example, the state of the system may not be directly observable but a noisy sensor measurement that conveys some information about the state is available. In these instances, a partially observable MDP (POMDP) is used to model the sequential decision making process. Choi and Kim (2011) extend the algorithms presented by Ng and Russell (2000) and Abbeel and Ng (2004) to the partially observable case. Later,

Chinaei and Chaib-Draa (2012) further extended the concept of IRL using POMDPs and demonstrate their proposed algorithm on a healthcare dialogue system.

3.1.2 Bayesian IRL

In general, Bayesian methods combine prior knowledge with data or evidence. Bayesian IRL techniques consider the expert's actions as the evidence used to update the estimate of the reward function. The first Bayesian IRL method uses a modified Markov Chain Monte Carlo (MCMC) algorithm to infer the reward function (Ramachandran and Amir 2007). The model for this Bayesian IRL formulation is diplayed in Fig. 2. Ramachandran and Amir state several advantages to their proposed Bayesian IRL algorithm. First, an optimal policy is not required. Second, it is not assumed that the expert has infallible decision making. Third, as in Bayesian IRL generally, external information about the problem can be incorporated into the reward function through the prior distribution. The probability of each state-action pair is assumed to be independent so the likelihood can be written as

$$\mathbb{P}(\mathcal{T}|R) = \mathbb{P}((s_1, a_1)|R)\mathbb{P}((s_2, a_2)|R)...\mathbb{P}((s_T, a_T)|R). \tag{16}$$

The likelihood of each state-action pair is modeled using an exponential distribution and a Q function conditioned on R

$$\mathbb{P}((s_t, a_t)|R) = \frac{1}{Z_t} e^{\alpha Q^*_R(s_t, a_t)}, \tag{17}$$

where α is a parameter that represents the confidence, and Z_t is the normalizing constant. The likelihood of the entire trajectory is written as

$$\mathbb{P}(\mathcal{T}|R) = \frac{1}{Z} e^{\alpha E_R(\mathcal{T})}, \tag{18}$$

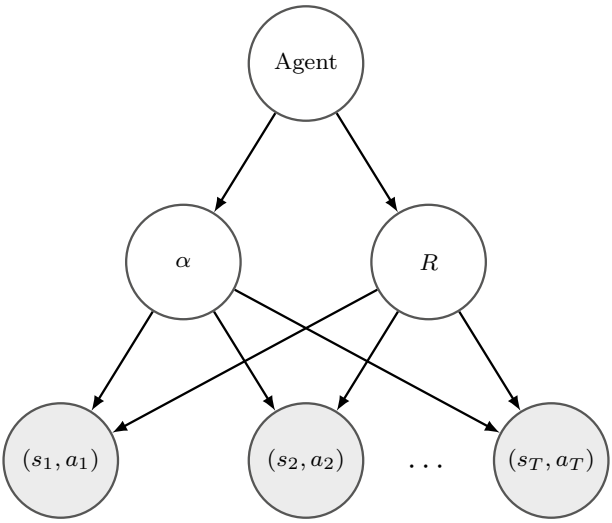

Fig. 2 The Bayesian IRL model from Ramachandran and Amir (2007)

where $E_R(\mathcal{T}) = \sum_{t=1}^{T} Q_R^*(s_t, a_t)$. Using Bayes theorem, the posterior for R can now be rewritten as

$$\mathbb{P}(R|\mathcal{T}) = \frac{1}{Z'} e^{\alpha E_R(\mathcal{T})} \mathbb{P}(R). \tag{19}$$

The normalizing constant Z' can be difficult to calculate. The proposed sampling algorithm Policy Walk uses the ratio of posterior probabilities so Z' does not need to be calculated. Rothkopf and Ballard (2013) extend this Bayesian formulation and the Policy Walk algorithm to modular MDPs (see (Rothkopf and Ballard 2010) for more information on modular MDPs).

The original Bayesian IRL formulation is limited because it cannot easily estimate reward structures for states that are not visited by the expert. Michini and How (2012) propose an extension to the Bayesian IRL method in Ramachandran and Amir (2007) by using a kernel function to estimate the similarity between states. When new candidate rewards are selected at random, more weight is given to states that are similar to those visited by the expert. This Bayesian method improves time to convergence and allows the user to tradeoff computation time for solution accuracy. Gaussian process IRL (Levine et al. 2011) is an extension of Bayesian IRL that also uses a kernel function to help estimate the reward for unvisited or unseen states.

Choi and Kim (2011) propose a Bayesian IRL framework that finds the maximum-a-posteriori (MAP) reward using a gradient method. They select the MAP estimate for the reward over the mean posterior reward commonly used in Bayesian IRL methods because the mean posterior reward can produce a policy inconsistent with the observed expert's behavior. This property is due to integrating over the entire reward space. Conversely, the MAP reward is simply a point that maximizes the posterior and is thus robust to rewards in the tail of the distribution.

The MAP solution to the IRL problem is $R_{MAP} = \text{argmax}_R[\log \mathbb{P}(\mathcal{T}|R) + \log \mathbb{P}(R)]$. It is generally assumed that the likelihood is a differentiable function of the experts' trajectories and either V^* or Q^*. Choi and Kim provide theorems that demonstrate that V^* and Q^* are convex and differentiable almost everywhere. If the prior on the reward is selected so that it is also differentiable, gradient descent can be used to find the MAP estimate $R_{new} = R + \delta \nabla_R \mathbb{P}(R|\mathcal{T})$, where δ is the step size.

Qiao and Beling (2011) provide MAP estimates for the reward function by formulating it as a quadratic convex programming problem. The reward is written as the vector $\mathbf{r} \in \mathbb{R}^n$ and Gaussian priors are assigned to $\mathbf{r} \sim \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}, \Sigma)$. The posterior distribution is derived using Bayes rule

$$\mathbb{P}(\mathbf{r}|\mathcal{T}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{r} - \boldsymbol{\mu})^\intercal \Sigma^{-1}(\mathbf{r} - \boldsymbol{\mu})\right). \tag{20}$$

Using the constraint proposed by Ng and Russell (2000), the IRL problem is written as the following quadratic program

$$\begin{aligned} \min_{\mathbf{r}} \quad & \frac{1}{2}\left((\mathbf{r} - \boldsymbol{\mu})^\intercal \Sigma^{-1}(\mathbf{r} - \boldsymbol{\mu})\right), \\ \text{s.t.} \quad & (\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)(\boldsymbol{I}_n - \gamma \boldsymbol{P}_{a^*})^{-1}\mathbf{r} \geq 0 \quad \forall a \in \mathcal{A}, \\ & \mathbf{r}_{min} \leq \mathbf{r} \leq \mathbf{r}_{max}. \end{aligned} \tag{21}$$

Qiao and Beling also use a preference graph and Gaussian processes to deal with problems in large or infinite state spaces. In a later work, Qiao and Beling (2013) use their Gaussian process method to recognize different agents based on their sequence of actions.

In many IRL implementations, expert trajectories are collected from multiple agents and it is assumed that all agents are maximizing the same reward function. Choi and Kim (2012) propose a nonparametric Bayesian formulation that does not assume that a single expert is maximizing a single reward function. A Dirichlet process mixture model is utilized to relax this assumption and estimate multiple rewards. Trajectories are assigned to clusters where each cluster is assumed to have a different reward structure. Under this formulation, an expert's behavior is generated from the following process:

1. Draw a cluster assignment using the Dirichlet process.
2. Draw the reward from the prior $\prod_{d=1}^{D} \mathcal{N}(r_{k,d}|\mu, \sigma)$ where k represents the cluster assignment, and d is the component of the reward function.
3. Draw the trajectory from the likelihood function given the cluster-specific reward and α.

A Metropilis-Hasting algorithm (Chib and Greenberg 1995) is utilized for inference. Figure 3 displays the graphical model for this IRL method. Later, Choi and Kim (2013) propose an extension that can learn composite features called Bayesian non-parametric feature construction IRL (BNP-FIRL). Budhraja and Oates (2017) build upon the BNP-FIRL framework and utilize neural networks and neuroevolution to map state features to estimates of state values. Neuroevolution of augmenting topologies (NEAT) (Stanley and Miikkulainen 2002) utilizes a genetic algorithm to find the optimal number of layers and nodes for a neural network. Budhraja and Oates combine IRL and NEAT.

Michini and How (2012) approach the single reward function assumption in a similar fashion by using a Chinese restaurant process (CRP) prior (Griffiths et al. 2004) to partition the reward into *subgoals*. The CRP is similar to the Dirichlet process used by Choi and Kim (2012), but is more flexible. The CRP can allow for data-dependent selection of parameters and functional dependence on the current partition. The primary advantage of the proposed Bayesian nonparametric IRL (BNIRL) method is that the partitioned rewards can be very simple and thus easier to learn. Complex reward functions can be expressed as multiple simple subgoals.

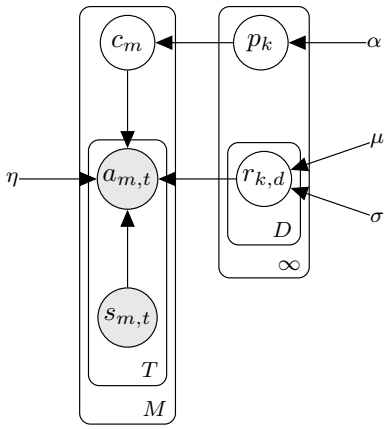

Fig. 3 Graphical model nonparametric Bayesian IRL formulation from Choi and Kim (2012)

The BNIRL method does not scale well to large state spaces because it requires calculating the optimal value function in order to evaluate the proposed reward. Michini et al. (2013) propose two solutions that do not require solving the full MDP. The first uses real-time dynamic programming, and the second modifies BNIRL to use a closed-loop controller instead of solving the MDP. Later, Michini et al. (2015) combine their work in Michini and How (2012) and Michini et al. (2013) and extend it to the continuous space using Gaussian processes.

Šošic et al. (2018) outline a significant limitation to the BNIRL method proposed by Michini and How. The BNIRL method is restricted to pure subgoal extraction and does not provide a mechanism to forecast expert behavior. This is due to the exchangeability in subgoal assignment in the original formulation. Thus, the recovered subgoals do not exhibit behavior similar to the expert's behavior. To address this limitation, Šošic *et al.* propose distance-dependent Bayesian nonparametric IRL (ddBNIRL) which replaces the exchangeable prior with a non-exchangeable prior and allows for information from visited states to be conveyed to non-visited states through a distance metric. This IRL formulation is later extended to address spatio-temporal relationships and time invariance (Šošić et al. 2018).

Lee et al. (2016) propose using a leveraged Gaussian process to incorporate negative examples into the reward learning process. One of the main limitations of IRL is the concentration of demonstrations around high reward states. The proposed method models a demonstrator that provides both positive and negative demonstrations characterized by a continuous proficiency parameter between -1 and 1. A proficiency of 1 represents a positive example, or what the agent should do, and a proficiency of -1 represents a negative example, or what the agent should not do.

Most IRL algorithms assume that the agents providing the sample trajectories are trustworthy, i.e., they are attempting to maximize the reward function. Zheng et al. (2014) propose robust Bayesian IRL to address the issue of sparse noise in the examples. Sparse noise describes the case where a majority of the demonstrations are trustworthy but a small number are untrustworthy. They motivate their work with a cab driver example. Most cab drivers are optimizing a single reward function. However, new cab drivers may not yet be experts and not follow the optimal route. Dishonest cab drivers may intentionally select a longer route to increase fairs. Robust Bayesian IRL includes a parameter that estimates the reliability of the demonstrations and uses the expectation-maximization (EM) algorithm (Dempster et al. 1977) to estimate the parameters.

Rothkopf and Dimitrakakis (2011) pose IRL as a preference elicitation problem where the goal is to infer if the decision maker prefers one set of events over another. For this formulation, it is assumed that there is some ordering to events and that each event has a utility where the decision maker is trying to maximize expected utility. Using these assumptions, the objective in the IRL problem is to assign numerical values to the utility. Rothkopf and Dimitrakakis define a structured prior on both the reward and the policy and then derive two Markov chain procedures for inferring the reward. Dimitrakakis and Rothkopf (2011) later expand this formulation to the multitask setting through the selection of priors on the reward and policy.

Hidden MDPs (hMDPs) (Kitani et al. 2012) were developed for the situation where the observer receives noisy state information. hMDPs are similar to partially observable MDPs but in hMDPs the state is known to the agent interacting with the environment and it is the observer collecting the trajectories that receives noisy state information. Surana (2014) proposes a Bayesian framework for the IRL problem for hMDPs where the observer is provided noisy observations $y_{1:T} = \{y_1, ..., y_T\}$ in place of the true state sequence. Surana

notes that given π, a hMDP reduces to a hidden Markov model (Rabiner 1989). MCMC methods can be used to sample the posterior for both the hidden states and the reward.

Brown and Niekum (2018) use the Bayesian IRL formulation to determine bounds on the performance of a policy. In this method, MCMC sampling is used to draw samples of an expert's reward function from the posterior distribution estimated using IRL, which is assumed to be an optimal policy. The loss is calculated between the evaluation policy and the policy estimated by IRL. A worst-case bound for the evaluation policy is also derived.

3.1.3 Maximum entropy IRL

Ziebart et al. (2008) propose the first maximum entropy IRL technique and claim that the proposed method addresses both noise in the trajectory and imperfect behavior. In the proposed maximum entropy IRL method, it is assumed that a learner is observing an agent, and that the agent is maximizing a function that linearly maps the state features to rewards. Ziebart *et al.* utilize the same concept as Ng and Russell (2000) and attempt to match expected feature counts with observed feature counts from the expert's trajectories. However, Ziebart *et al.* take a probabilistic perspective and model the probability of observing any single trajectory as

$$\mathbb{P}(\mathcal{T}_m|\theta) = \frac{1}{Z(\theta)} e^{\theta^\intercal \mathbf{f}_{\mathcal{T}_m}}, \tag{22}$$

where θ is a vector of feature weights, $\mathbf{f}_{\mathcal{T}_m} = \sum_{s_j \in \mathcal{T}_m} \mathbf{f}_{s_j}$ is the path feature count, and $Z(\theta)$ is the partition function given the feature weights. Equation 22 models the trajectory for deterministic behavior. For non-deterministic behavior, the transition probability is included and the probability of observing a set of trajectories is

$$\mathbb{P}(\mathcal{T}|\theta, P) \approx \frac{e^{\theta^\intercal \mathbf{f}_{\mathcal{T}}}}{Z(\theta, P)} \prod_{s_{t+1}, a_t, s_t} P^{a_t}_{s_t, s_{t+1}}. \tag{23}$$

The reward weights are learned from the demonstrated behavior by maximizing the log likelihood $L(\theta)$ under the entropy distribution (Eq. 23)

$$\theta^* = \arg\max_\theta \sum_{m=1}^{M} \log \mathbb{P}(\tilde{\mathcal{T}}_m|\theta, P), \tag{24}$$

where $\tilde{\mathcal{T}}_m$ is the m^{th} observed trajectory. Gradient descent is used to maximize the likelihood and the gradient is

$$\begin{aligned} \nabla L(\theta) &= \tilde{\mathbf{f}} - \sum_{\mathcal{T}} \mathbb{P}(\mathcal{T}|\theta, P)\mathbf{f}_{\mathcal{T}} \\ &= \tilde{\mathbf{f}} - \sum_{s_i} D_{s_i} \mathbf{f}_{s_i}, \end{aligned} \tag{25}$$

where $\tilde{\mathbf{f}}$ is the empirical feature count, and D_{s_i} is the expected state visitation frequency, which Ziebart *et al.* provide an efficient algorithm to calculate.

Later works characterize maximum entropy IRL as IOC and utilize it for human behavior modeling (Ziebart et al. 2009) and pointing target prediction (Ziebart et al. 2012). Other works expand into the continuous space using the maximum entropy concept (Aghasadeghi and Bretl 2011; Kalakrishnan et al. 2013, 2010; Levine and Koltun 2012). Furthermore,

Ziebart et al. (2013) propose the concept of maximum casual entropy for interacting processes and demonstrate how this concept can be applied to the IOC problem. Bloem and Bambos (2014) and later Zhou et al. (2018) extend maximum casual entropy to the infinite time horizon problem.

One of the major limitations of IRL is the reliance on successful demonstrations provided by the expert. To address this limitation, Shiarlis et al. (2016) modify the maximum casual entropy formulation to incorporate unsuccessful demonstrations. A constraint is added to the optimization problem that forces the algorithm to find reward weights that not only minimize the difference between the empirical feature expectation for the successful examples and the learned feature expectation, but also maximize the difference between the empirical feature expectation for the unsuccessful examples and the learned feature expectation.

Another approach to overcoming the limitation of the demonstrations being provided solely by an expert combines maximum entropy IRL with semi-supervised learning (Audiffren et al. 2015). In this formulation, the supervised examples $\Sigma^* = \{\mathcal{T}_i^*\}_{i=1}^l$ are those provided by the expert, and the unsupervised examples $\tilde{\Sigma} = \{\tilde{\mathcal{T}}_j\}_{j=1}^u$ are those provided by a non-expert. It is assumed that the function $s(\mathcal{T}, \mathcal{T}')$ is provided and measures the similarity between two trajectories. A *pairwise penalty* is defined

$$\Psi(\theta|\Sigma) = \frac{1}{2(l+u)} \sum_{\mathcal{T},\mathcal{T}' \in \Sigma} s(\mathcal{T}, \mathcal{T}')(\theta(\mathbf{f}_{\mathcal{T}} - \mathbf{f}_{\mathcal{T}'}))^2, \tag{26}$$

where $\Sigma = \Sigma^* \cup \tilde{\Sigma}$. This penalty term is added to Eq. 24 to estimate the reward weights

$$\theta^* = \text{argmax}_\theta(L(\theta|\Sigma^*) - \lambda\Psi(\theta|\Sigma)), \tag{27}$$

where λ is the tradeoff parameter between the log likelihood and the similarity between the supervised and unsupervised trajectories. There are numerous possible choices for the similarity function, but while a handcrafted function would perform the best, a radial basis function is a suitable selection.

Bogert et al. (2016) adapt maximum entropy IRL to partially observed trajectories. In this context, both states and actions can be occluded. Bogert *et al.* propose using the EM algorithm (Dempster et al. 1977) to estimate the missing information in the provided trajectory. However, computing the expectation becomes intractable as the size of the trajectory grows or multiple trajectories are provided by multiple experts. Therefore, Bogert and Doshi (2017) propose modeling the process generating the trajectories as a dynamic Bayesian network and demonstrate that Gibbs sampling can be used to perform fast inference on the latent variables.

Byravan et al. (2015) address scaling issues with maximum entropy IRL and continuous state spaces by discretizing the continuous trajectories into a graph. Once a sufficiently coarse graph is created, maximum entropy IRL is used to estimate the reward function. Shimosaka et al. (2017) build on this graph-based concept and adapt the creation of the graph to situations where the state space contains temporal information and transitions can vary depending on this information.

Wulfmeier et al. (2016) use deep architectures to map sensory information to reward function parameters in a path planning problem. This type of application, especially in a dense urban environment, requires a complex non-linear reward function. A maximum entropy deep IRL algorithm is proposed that can scale to large complex environments but requires a large number of expert trajectories. For the demonstration on urban driving, the inputs into the neural network are visual sensors (including 2D and 3D LIDARs),

and over 25,000 trajectories where used as expert demonstrations. Wulfmeier et al. (2017) later expand on this initial study and include more indepth discussion and experimentation. Chen et al. (2019) and Finn et al. (2016) extend maximum entropy deep IRL to continuous action and state spaces.

Mendez et al. (2018) propose efficient lifelong IRL where the maximum entropy formulation is extended to an agent continually learning multiple tasks. It is assumed that an expert can provide demonstrations to the agent for each task. The agent sequentially learns the parameters of the reward function for a task before moving on to the next task. In order to improve efficiency, a transfer learning framework is proposed so that the agent can learn parts of the reward function from previously learned similar tasks. Yu et al. (2019) develop probabilistic embeddings for meta-inverse reinforcement learning (PEMIRL), which combines aspects of maximum entropy IRL, meta-learning, and deep generative models to learn reward functions that can generalize to new tasks with only a single demonstration provided by the expert. Ranchod et al. (2015) propose non-parametric Bayesian reward segmentation (NPBRS) to identify multiple skills in the set of demonstrations. NPBRS uses maximum entropy IRL to estimate a set of unknown reward functions that maximizes the likelihood of the demonstrations.

Boularias et al. (2012) propose structured apprenticeship learning. This technique builds on the concept of maximum entropy IRL, but assumes a probability distribution over policies as opposed to over trajectories. The objective in structured apprenticeship learning is to find the distribution over policies that maximizes entropy and closely matches the feature expectation. The reward weights are learned during the maximization process.

3.1.4 Miscellaneous IRL methods

Generative adversarial networks (GANs) pair two neural networks, where a generator creates new data with the objective of confusing the discriminator, and the discriminator attempts to determine if an observation is a sample from the data set or a fake sample created by the generator (Goodfellow et al. 2014). Figure 4 displays the basic structure of a GAN. This concept has been extended to IRL. Generative adversarial imitation learning (GAIL) (Ho and Ermon 2016) extends the concepts proposed by Ho et al. (2016) but uses a GAN to learn the expert's policy. Hausman et al. (2017) build on GAIL and propose a framework that can accommodate unlabeled and unstructured data, i.e., data consisting of multiple tasks without corresponding labels distinguishing the

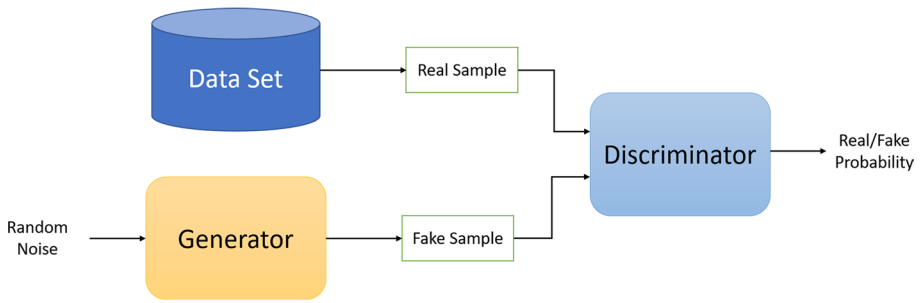

Fig. 4 Basic structure of generative adversarial network

tasks. OptionGAN (Henderson et al. 2018) further generalizes to the scenario where the demonstrations are produced by multiple reward functions.

Hahn and Zoubir (2015) frame the IRL problem as a generative probabilistic model. This work is based upon prior work (Toussaint and Storkey 2006) in RL that also formulates the problem as a generative probabilistic model. Toussaint and Storkey (2006) propose modeling the reward as a mixture with K components, where K is the length of the trajectory. This model assumes that the reward is only received at the end of each mixture component. This joint distribution is

$$\mathbb{P}(r, s_{1:k}, a_{1:k}|k) = \mathbb{P}(s_1)\mathbb{P}(a_1|s_1)\prod_{n=2}^{k}\mathbb{P}(s_n|s_{n-1}, a_n)\mathbb{P}(a_n|s_n)\mathbb{P}(r|s_k, a_k). \tag{28}$$

Under this model, the probability of the reward is equivalent to the reward function, and the marginal distribution is $\mathbb{P}(r, s_{1:K}, a_{1:K}) = \sum_{k=1}^{K}\mathbb{P}(r, s_{1:k}, a_{1:k}|k)\mathbb{P}(k)$. Toussaint and Storkey demonstrate that this marginal distribution can be calculated recursively for the forward RL problem. Hahn and Zoubir leverage this to derive an EM algorithm for the IRL problem.

The reward function is assumed to be a linear combination of basis functions $R(s, a) = \sum_{d=1}^{D}\alpha_d\phi_d$. The reward is also assumed to follow a Gaussian distribution $\mathbb{P}(r|s, a) = \mathcal{N}(r|R(s, a), \sigma_r^2)$. The complete data likelihood for M independent trajectories is

$$\begin{aligned}\mathbb{P}_\theta(\mathcal{T}, r, k) &= \prod_{m=1}^{M}\mathbb{P}_\theta(r, s_{1:k}^{(m)}, a_{1:k}^{(m)}|k)\mathbb{P}(k)\\ &= \prod_{m=1}^{M}\left(\prod_{n=1}^{k}\mathbb{P}_\theta(a_n^{(m)}|s_n^{(m)})\right)\mathbb{P}_\theta(r|s_n^{(m)}, a_n^{(m)}),\end{aligned} \tag{29}$$

where θ is the set of parameters that includes the reward weights and the parameters for the policy, and $\mathbb{P}_\theta(\cdot|\cdot)$ indicates that the distribution is dependent on θ. The Q function for the EM algorithm is

$$\begin{aligned}Q(\theta|\theta') = \int\sum_{k=1}^{\infty}\mathbb{P}_{\theta'}(k, r|\mathcal{T})\Bigg(\sum_{m=1}^{M}\Big(\log\mathbb{P}_\theta(r|s_k^{(m)}, a_k^{(m)})\\ + \sum_{n=1}^{k}\log\mathbb{P}_\theta\big(a_n^{(m)}|s_n^{(m)}\big)\Big)\Bigg)dr + C,\end{aligned} \tag{30}$$

where θ' is the set of parameters from the previous iteration, and C represent the constant terms. The E and M steps of the EM algorithm can be derived from the Q function.

Doerr et al. (2015) propose an algorithm that poses the IRL problem as an RL problem. An RL agent is tasked with finding reward weights using the policy search algorithm where the corresponding optimal policy produces trajectories similar to the expert's trajectories. Adversarial IOC (Chen et al. 2016) considers the situation where the demonstrator and the learning agent operate under different sequential decision models. For example, the demonstrator and the learning agent have different capabilities which leads to a difference in the action space, or they operate in different environments which leads to a difference in the transition dynamics. Adversarial IOC is posed as a zero-sum game where the learning agent is trying to learn a control policy that

minimizes a loss function, and an adversarial agent is trying to learn a control policy that mimics the demonstrations but also maximizes the learning agent's loss.

Babes et al. (2011) develop an apprenticeship learning algorithm that assumes that multiple agents with different reward functions provide the set of expert trajectories. They propose using the EM algorithm to cluster the trajectories and then use IRL or apprenticeship learning to estimate either the expert's reward or policy, respectively. Nguyen et al. (201) use the EM algorithm to estimate locally consistent rewards. Their IRL formulation is similar to that in Choi and Kim (2012) where it is assumed that the agent is optimizing multiple reward functions.

Many IRL methods require reward features or basis functions, which must be selected *a priori*. Levine et al. (2010) present a method that simultaneously constructs reward features and estimates the reward weights. The objective is to reduce the reward feature space from a set of possible features to a set of relevant features. This method iteratively adds features to the reward feature set Φ. Each iteration is composed of two steps. The first step finds the reward function that best fits the current feature set Φ^k and the provided demonstrations. The second step generates a new feature set Φ^{k+1} that improves on the reward function variation. Klein et al. (2011) demonstrate that a set of $|\mathcal{S}| - 2$ features exist in the reward space that are relevant to the IRL problem.

Most IRL formulations assume that the expert is risk neutral, i.e., the expert is attempting to maximize the expectation of future reward. However, some expert's or systems may not be risk neutral and may, for example, attempt to maximize the minimum possible future reward. To address this situation, Majumdar et al. (2017) propose risk-sensitive IRL. A risk metric maps an uncertain reward to a real number. Majumdar et al. outline procedures for estimating the risk metric when the reward function is known and the more general case where both are unknown. This work is extended in Singh et al. (2018), which includes improvements to the model and the algorithm and expanded validation experiments.

Behavior cloning, a form of imitation learning and apprenticeship learning, often uses a supervised classifier to learn a mapping of states to actions that replicate the behavior of the provided trajectories. Ratliff et al. (2009) propose inverse optimal heuristic control which combines a behavior cloning classifier with a traditional IRL paradigm. Gradient optimization is used to estimate the weights of the reward function.

Linearly-solvable MDPs (Kappen 2005; Todorov 2007) are a special case of the general MDP where the action is interpreted as a stochastic state selection process. In other words, the agent controls the dynamics of the system rather than taking an action by selecting a state to which they wish to move. A policy for linearly-solvable MDPs is generally written as $\pi(s'|s)$. Dvijotham and Todorov (2010) develop a maximum likelihood estimation IRL algorithm for linearly-solvable MDPs.

3.2 Extensions of IRL

This section outlines areas in the IRL literature that have been studied but need further development in order to make IRL commonly used in real-world systems. The first extension to IRL is model-free methods. While the model-based methods outlined in the previous section represent a significant advance in the field, the reliance on a model limits the real-world application of IRL. The second extension to IRL involves active learning and feedback. In the basic IRL paradigm, the demonstrations are provided to the IRL algorithm. This extension allows for feedback to be requested from the expert in order to

Table 2 Extensions of inverse reinforcement learning

Model-free	Active learning	Multi-agent
Boularias et al. (2011)	Lopes et al. (2009)	Natarajan et al. (2010)
Suay et al. (2016)	Sadigh et al. (2017)	Reddy et al. (2012)
Tossou and Dimitrakakis (2013)	Kunapuli et al. (2013)	Bogert and Doshi (2014)
Ho et al. (2016)	Ezzeddine et al. (2018)	Bogert and Doshi (2015)
Klein et al. (2012)	El Asri et al. (2016)	Lin et al. (2018)
Klein et al. (2013)	Odom and Natarajan (2016)	Lin et al. (2019)
Uchibe and Doya (2014)	Pan and Shen (2018)	Zhang et al. (2019)
Uchibe (2016)	Amin et al. (2017)	Hadfield-Menell et al. (2016)
Uchibe (2018)	Woodworth et al. (2018)	Šošic et al. (2017)
Herman et al. (2016)		
Herman et al. (2016)		
Kangasrääsiö and Kaski (2018)		
Makino and Takeuchi (2012)		

improve the efficiency of estimating the reward. The final extension is multi-agent IRL. The methods are listed in Table 2.

3.2.1 Model-free IRL

Most IRL methods assume that a model (more specifically the transition function) is known to the learner and that it can be used to solve the forward RL problem in order to evaluate the quality of the estimated reward function. However, in many scenarios, the transition function may be unknown and difficult or impossible to estimate. Some IRL methods only require access to a routine for finding the optimal policy, which could be a model-free RL approach. However, many of these methods need a large number of interactions with the environment in order to converge to an estimate of the optimal policy, which would significantly increase the computation time of the IRL algorithm. In this section, model-free IRL approaches are outlined.

Boularias et al. (2011) propose relative entropy IRL as the first model-free IRL method, which builds on the maximum entropy IRL algorithm (Ziebart et al. 2008). The proposed algorithm minimizes the relative entropy (or KL-divergence) between the empirical distribution of the demonstrations under a baseline policy and the distribution of the demonstrations under the learned policy. Let $p(\mathcal{T}_m)$ be the probability distribution of an observed trajectory in the set of trajectories $\mathcal{T}$, and let $q(\mathcal{T}_m)$ be the probability distribution under the baseline policy and the transition function. The IRL problem can be formulated as minimizing the relative entropy between these two distributions with constraints

$$\begin{aligned} &\min_p \sum_{\mathcal{T}_m \in \mathcal{T}} p(\mathcal{T}_m) \ln \left[\frac{p(\mathcal{T}_m)}{q(\mathcal{T}_m)} \right] \\ &\text{s.t.} \left| \sum_{\mathcal{T}_m \in \mathcal{T}} p(\mathcal{T}_m) f_d^\tau - \hat{f}_d \right| \leq \epsilon_d \quad d = 1, ..., D \\ &\sum_{\mathcal{T}_m \in \mathcal{T}} p(\mathcal{T}_m) = 1 \\ &p(\mathcal{T}_m) \geq 0 \quad \forall \mathcal{T}_m \in \mathcal{T}, \end{aligned} \tag{31}$$

where f_d^τ is the expected feature count under the learned policy for the d^{th} feature, and $\hat{f}_d$ is the empirical feature count from the observed trajectories. The reward weights α can be included in the first constraint. The Lagrangian and KKT conditions are used to define the dual problem, and the IRL problem now becomes maximizing the dual with respect to α. Suay et al. (2016) propose using relative entropy IRL to recover dense reward functions that are later used to solve the forward RL problem in a sparse reward environment.

Tossou and Dimitrakakis (2013) develop two model-free IRL algorithms, which are extensions of model-based Bayesian approaches (Dimitrakakis and Rothkopf 2011; Rothkopf and Dimitrakakis 2011). The first model, called the reward prior model, uses a different parameterization of the reward function and a model-free method for estimating the value function. The second model, called the policy optimality model, uses a prior distribution on the policy and estimates a posterior on the policy from the data. Several variations of these models using different prior distributions are outlined.

Ho et al. (2016) set the model-free IRL problem as an apprenticeship learning problem. The objective of the proposed method is to find an optimal policy that matches the expert's. However, there is a subroutine which estimates the cost function. This method assumes that the true cost function c is within a set of possible cost functions $\mathcal{C}$. First, the state visitation distribution is defined as $\rho_\pi(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_{p_0,\pi}(s_t = s)$. $\mathbb{P}_{p_0,\pi}(s_t = s)$ is the probability of being in s at t when the initial state is sampled from p_0 and then π is followed. The state-action visitation distribution is $\rho_\pi(s, a) = \pi(a|s)\rho_\pi(s)$. The expected cost can be calculated using $\eta^c(\pi) = \sum_{s,a} \rho_\pi(s, a)c(s, a)$. The objective function for the proposed apprenticeship learning algorithm is $\min_\pi \sup_{c \in c} \eta^c(\pi) - \eta^c(\pi_E)$.

Klein et al. (2012) suggest using structured classification as a model-free IRL method. The proposed idea learns a multi-class classifier that mimics the expert's policy. An estimate of the expert's feature expectation is used to learn the parameters of a linear scoring function. A classifier is trained that maps the reward basis functions to the reward considering the linear scoring function. The parameters of the learned classifier are the reward weights. A model-free algorithm can be used to estimate the expert's feature expectation, thus making the structured classification IRL method model-free. Klein et al. (2013) later build upon this method and utilize a regression algorithm to estimate the reward function.

Uchibe and Doya (2014) propose a model-free IRL approach that utilizes dynamic policy programming (Azar et al. 2012). In the proposed approach, the reward function is referred to as a cost function $l(s, a)$ and the objective for the forward problem is to find a policy that minimizes expected future cost. Further, the cost function is constrained and has the following form

$$l(s,a) = q(s) + \frac{1}{\beta}\text{KL}(\pi(\cdot|s)||\pi_0(\cdot|s)), \tag{32}$$

where $\text{KL}(\pi(\cdot|s)||\pi_0(\cdot|s))$ represents the KL-divergence between two policies, $q(s)$ is the state-dependent cost component, and β is the inverse temperature parameter. The optimal value function can now be written as

$$V^*(s) = \min_{\pi(\cdot|s)} \sum_{a\in\mathcal{A}} \pi(a|s)\left[q(s) + \frac{1}{\beta}\ln\left[\frac{\pi(a|s)}{\pi_0(a|s)}\right] + \gamma \sum_{s'\in\mathcal{S}} P^a_{s,s'} V^*(s')\right]. \tag{33}$$

If $q_\beta(s) = \beta q(s)$ and $V^*_\beta(s) = \beta V^*(s)$, then the log ratio of the policies is

$$\ln\left[\frac{\pi(a|s)}{\pi_0(a|s)}\right] = q_\beta(s) + \gamma \sum_{s'\in\mathcal{S}} P^a_{s,s'} V^*_\beta(s') - V^*_\beta(s). \tag{34}$$

The IRL problem is now broken into two parts. The first estimates the log ratio of the stochastic policies, and the second estimates the cost and value functions. Assume that two demonstration sets are provided, where $\mathcal{T}^\pi$ is produced under policy π, and $\mathcal{T}^\theta$ is produced under policy π_0. Least squares conditional density estimation (Sugiyama et al. 2010) is used to estimate the negative ratio of these policies. Least squares with regularization is used to estimate the linearly parameterized cost $\hat{q}_\beta(s;\boldsymbol{w}_q^\intercal) = \boldsymbol{w}_q^\intercal \boldsymbol{\psi}_q(s)$ and value functions $\hat{V}^*_\beta(s;\boldsymbol{w}_V^\intercal) = \boldsymbol{w}_V^\intercal \boldsymbol{\psi}_V(s)$, where $\boldsymbol{w}$ are the parameters and $\boldsymbol{\psi}$ are the basis functions. Theoretically, any basis function could be selected but Gaussian functions are recommended by Ucbibe and Doya. Uchibe later extends this work and utilizes deep learning architectures to estimate the cost and value functions instead of the least squares method (Uchibe 2016; Uchibe 2018).

Herman et al. (2016) propose maximizing the likelihood with respect to the reward function and the system dynamics simultaneously by learning both components of the MDP. Herman et al. suggest that it may be advantageous to learn the true transition function and the agent's belief about the transition function, designated $\mathbb{P}_A(s'|s,a)$, as these two functions may be different. Therefore, the proposed algorithm estimates three sets of parameters

- $\boldsymbol{\theta}_R$ - The reward feature weights,
- $\boldsymbol{\theta}_{T_A}$ - The parameters of the agent's transition function,
- $\boldsymbol{\theta}_T$ - The parameters of the true transition function.

The likelihood of the demonstrated trajectories is

$$\mathbb{P}(\mathcal{T}|\boldsymbol{\theta}) = \prod_{\mathcal{T}_m\in\mathcal{T}} \mathbb{P}\left(s_1^{\mathcal{T}_m}\right) \prod_{t=1}^{T_m-1}\left[\pi_\theta\left(s_t^{\mathcal{T}_m}|a_t^{\mathcal{T}_m}\right)\mathbb{P}_{\theta_T}\left(s_{t+1}^{\mathcal{T}_m}|s_t^{\mathcal{T}_m}, a_t^{\mathcal{T}_m}\right)\right], \tag{35}$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_R\ \boldsymbol{\theta}_{T_A}\ \boldsymbol{\theta}_T]$. Note that the the transition function is only dependent on $\boldsymbol{\theta}_T$, while the policy is dependent upon $\boldsymbol{\theta}_R$ and $\boldsymbol{\theta}_{T_A}$. The gradient of the log likelihood is used to update the parameters. Herman et al. (2016) expand this work to the situation where the agent's stochastic policy mapping is also unknown and needs to be estimated.

Kangasrääsiö and Kaski (2018) present a model-free IRL method where only partial expert trajectories are available. The authors present three methods to estimate the rewards under this assumption. The first computes the exact likelihood. The second utilizes

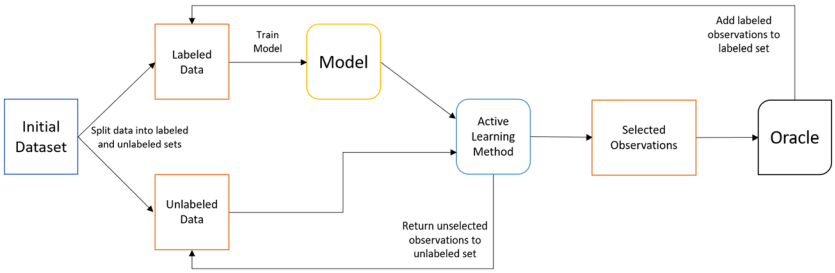

Fig. 5 Basic active learning framework

Monte-Carlo estimation of the likelihood. The third uses an approximate Bayesian computation approach (Sunnåker et al. 2013) to evaluate the likelihood. Makino and Takeuchi (2012) propose two methods for learning the components of a POMDP which include the transition function and the reward function. The first uses the COBYLA algorithm (Powell 1998) to find local MAP estimates for the parameters, and the second uses an MCMC sampler.

3.2.2 Active learning and feedback

In general, active learning is an area of machine learning where the learning agent can query an unlabeled data set and ask that an expert label specific examples (Settles 2012). By selecting specific examples, the learning agent hopes to learn with fewer labeled examples. The basic active learning framework is displayed in Fig. 5. This section outlines IRL algorithms that utilize active learning concepts and interact with the expert who is providing feedback.

Lopes et al. (2009) utilize the Bayesian IRL framework and active learning concepts to reduce the number of demonstrations needed to estimate the reward function. In this framework, the learner is able to query the expert with regard to their action. The learner selects the queried state based on the current posterior estimate of the reward function. Sadigh et al. (2017) also propose an active learning technique for IRL, but their work assumes that a human provides a preference between two sample trajectories. Kunapuli et al. (2013) utilize preference elicitation to incorporate information from the human expert about their preference for actions in a specific state. Ezzeddine et al. (2018) use feedback from a human trainer when the provided demonstrations are sub-optimal. The human trainer provides feedback in the form of positive and negative reinforcement with respect to the action in the demonstration set.

El Asri et al. (2016) propose score-based IRL (SBIRL) where each trajectory is provided a score from an outside expert who may not produce accurate scores. The scores are interpreted as noisy estimates of the sum of discounted future rewards. One advantage of SBIRL is that the collected trajectories can be produced by an optimal or sub-optimal policy. Reward weights are estimated by solving a linear least-squares problem

$$\begin{aligned}\alpha_m &= \text{argmin}_{\alpha_m} \frac{1}{M} \sum_{m=1}^{M} (v_m - \alpha^\intercal \mu(\mathcal{T}_m))^2 \\ &= \left(\frac{1}{M} \sum_{m=1}^{M} \mu(\mathcal{T}_m) \mu(\mathcal{T}_m)^\intercal \right)^{-1} \frac{1}{M} \sum_{m=1}^{M} \mu(\mathcal{T}_m) v_m,\end{aligned} \tag{36}$$

where v_m is the score for the m^{th} trajectory, and μ is the empirical discounted sum of the reward features.

Odom and Natarajan (2016) propose active advice seeking IRL, which is a more expressive form of interaction with a human operator and has two advantages over active learning: it can receive information over a large portion of the feature space, and it can receive a set of optimal labels as opposed to a single label. The set of provided labels convey information about actions that are preferred and actions that should be avoided. Pan and Shen (2018) propose human-interactive IRL. In this method, the human first supplies full demonstrations and also defines subgoals for the agent. The agent then learns from the provided demonstrations and attempts to complete the tasks. The human can then provide further feedback in the form of additional full demonstrations or specific feedback on the subgoals.

In order to incorporate a notion of safety when a robot is interacting with a human, Amin et al. (2017) propose repeated IRL. The human provides feedback to the robotic agent whenever the human is "surprised" by the agent's policy. The agent's objective is to find a policy that minimizes the number of times the human is surprised. Woodworth et al. (2018) relax some of the assumptions made by repeated IRL and propose observational repeated IRL. This algorithm does not assume that the human can provide feedback to the agent.

3.2.3 Multi-agent IRL

Most IRL algorithms address the problem of a single agent interacting with an environment. This section outlines multi-agent IRL (MIRL) where multiple agents are interacting with the environment and each other. One possible solution is to model each agent independently and assume that the actions of the other agents are part of the stochastic nature of the environment. However, this may produce suboptimal results in situations where the agents are coordinating their actions to maximize a joint reward function.

Natarajan et al. (2010) were the first to tackle the MIRL problem. Their goal is to learn the reward functions of multiple agents acting independently in an environment and then estimate a policy for all agents for a central controller. Natarajan *et al.* assume an average-reward MDP. The Bellman equation for this type of MDP is

$$V^\pi(s) = r_s(\pi(s)) + \sum_{s' \in \mathcal{S}} P_{s,s'}^{\pi(s)} V^\pi(s) - \rho, \tag{37}$$

where ρ is the average reward under π. H-learning (Tadepalli and Ok 1998) can be used to solve the forward RL problem for an average-reward MDP. In line with the work of Ng and Russell (2000), Natarajan et al. derive the following constraint

$$(\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \boldsymbol{P}_{a_1})^{-1}(\mathbf{R} - \boldsymbol{\rho}) \geq 0 \quad \forall a \in \mathcal{A} \backslash a_1. \tag{38}$$

In the case of multiple agents, the average adjusted reward vector $(\mathbf{R} - \boldsymbol{\rho})$ can be replaced with $\sum_{i=1}^{I} w_i(\mathbf{R}_i - \boldsymbol{\rho}_i) = \Theta \mathbf{w}$, where I is the number of agents, $\Theta = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_I\}$, and $\boldsymbol{\theta}_i$ is the

average adjusted reward for agent i. Using the updated constraint, rewards are estimated using the following optimization problem

$$\begin{aligned} \max_{\theta_1,...,\theta_T} \; & -\lambda||\boldsymbol{\theta}|| + \sum_{m=1}^{M} \min_{a \in \mathcal{A} \setminus a_1} (\boldsymbol{P}_{a_1}(m) - \boldsymbol{P}_a(m))(\boldsymbol{I} - \boldsymbol{P}_{a_1})^{-1}\boldsymbol{\theta} \\ \text{s.t.} \quad & (\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \boldsymbol{P}_{a_1})^{-1}\boldsymbol{\theta} \geq 0 \quad \forall a \in \mathcal{A} \setminus a_1 \\ & \boldsymbol{\theta} = \sum_{i=1}^{I} \boldsymbol{\theta}_i w_i \\ & |\theta_i^j| < \theta_{max}, \quad \text{for } i = 1, ..., I, \text{ and } m = 1, ..., M. \end{aligned} \tag{39}$$

Reddy et al. (2012) also expand the original IRL formulation in Ng and Russell (2000) to the multi-agent setting. Their work focuses on non-cooperative agents and general sum stochastic games. The Nash equilibrium (Nash 1950) is a fundamental concept in game theory, and Reddy *et al.* present a theorem that rewrites the Nash equilibrium in the multi-agent RL framework. Using this result, the optimal reward function must satisfy the following condition: $Q(s, \pi_i \pi_{-i}(s)) \geq Q(s, a_i \pi_{-i}(s))$, $\forall a_i \in \mathcal{A}_i$, where π_i is the strategy for agent i in response to the strategy profile π_{-i}, i.e. the strategies of all agents except i. Using this condition, the constraint from Ng and Russell (2000) can be rewritten as

$$\left(\boldsymbol{P}_{\pi_i^*, \pi_{-i}^*} - \boldsymbol{P}_{a_i, \pi_{-i}^*}\right)\left(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_i^*, \pi_{-i}^*}\right)^{-1} \mathbf{R}_i \geq 0. \tag{40}$$

Reddy et al. present algorithms for estimating the reward function when the policy is known and for when the policy is unknown but sample trajectories are provided.

Bogert and Doshi (2014) investigate the situation where multiple robots are interacting in an environment but trajectory information from other robots can be occluded. This multi-agent IRL approach builds on the maximum entropy IRL algorithm presented in Ziebart et al. (2008). In order to reduce the size of the joint state and action spaces, each robot is modeled using a separate MDP except when the robots must interact. When the robots must interact, their actions are modeled as a Nash equilibrium for a game. Bogert and Doshi (2015) extend this work to the model-free setting where they develop an algorithm that simultaneously estimates rewards and transition probabilities.

Lin et al. (2018) take a Bayesian approach to MIRL and build upon the work by Qiao and Beling (2011). The presented method is specific to two-player zero-sum games. In this type of game, the reward function is symmetric $r^1(s, a^1, a^2) = -r^2(s, a^1, a^2)$, where the superscripts designate the player. An optimization problem with constraints is formulated that is similar to the MAP estimation proposed by Qiao and Beling (2011) and assumes a Gaussian prior on the rewards. This approach is later built upon and expanded to general sum games (Lin et al. 2019). Zhang et al. (2019) develop non-cooperative IRL in the context of zero-sum games where the agents' reward functions are in conflict and only one agent knows the true reward, i.e., the joint reward function of both agents.

In certain situations, there could be a mixture of the type of agent in the system, for example, robotic agents and human agents. Hadfield-Menell et al. (2016) investigate the situation where a robotic agent and a human agent are cooperatively working to maximize the human's reward function. Hadfield-Menell et al. argue against the classic single-agent formulation for this setting for two reasons. First, the robot should not directly learn the humans reward function because this could generate unnecessary behavior. Second, the classic formulation does not allow for the human to provide

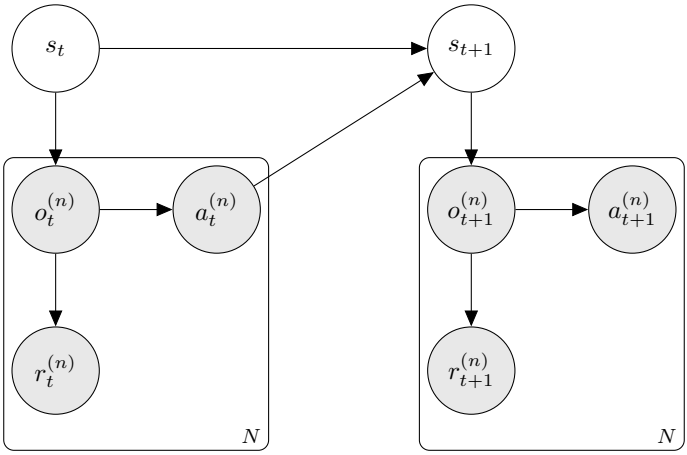

Fig. 6 Graphical model of a swarMDP (Šošic et al. 2017). *N* is the number of agents in the system

teaching demonstrations, i.e., demonstrations that are sub-optimal for the task but can provide vital information for teaching the robot. In order to address this problem, Hadfield-Manell *et al.* propose cooperative IRL (CIRL) and formulate it as a two-player game with partial information for the robot. The human player knows the reward function being maximized, but the robot must learn the reward function by playing the game. In the end, Hadfield-Manell *et al.* demonstrate that this problem can be reduced to a single-agent POMDP.

Šošic et al. (2017) study homogeneous or swarm systems and attempt to find a single local reward function that explains the global behavior of the system. In homogeneous systems, agents are assumed to have a similar architecture and are interchangeable. Agents are also assumed to be restricted to local areas of the system, meaning they only receive information that is within a predetermined region. A swarMDP (Fig. 6) is formulated for this type of system, and it is demonstrated that MIRL for this type of system can be reduced to the single-agent case.

4 Applications

IRL has been applied in numerous domains. This section categorizes the literature on the application of IRL into three branches defined by their goal. The first category of applications attempts to make the autonomous agent mimic the expert. The second category attempts to learn the reward function in order to improve the interaction between two systems, i.e., the reward function is learned so that one system can better understand the goals of another system. The third category interprets the learned reward function in order to learn about the system or subject. Table 3 lists the papers that fall into each category.

Table 3 Applications of inverse reinforcement learning

Mimic the expert	Interact with other systems	Learn about a system
Abbeel et al. (2007)	Ziebart et al. (2009)	Elnaggar and Bezzo (2018)
Coates et al. (2009)	Chung et al. (2010)	Yang et al. (2015)
Inga et al. (2017)	Kuderer et al. (2012)	Yang et al. (2018)
Howard et al. (2010)	Kuderer et al. (2013)	Lee et al. (2017)
Mori et al. (2011)	Kretzschmar et al. (2014)	Lee et al. (2018)
Lopes et al. (2007)	Kretzschmar et al. (2016)	Yamaguchi et al. (2018)
Okal et al. (2015)	Pfeiffer et al. (2016)	Hirakawa et al. (2018)
Okal and Arras (2016)	Shkurti et al. (2018)	
Lopes et al. (2011)	Scobee et al. (2018)	
Kim and Pineau (2016)	Chandramohan et al. (2011)	
Choi et al. (2018)	Boularias et al. (2010)	
Pflueger et al. (2019)	Chinaei and Chaib-Draa (2014)	
Krishnan et al. (2018)	Chinaei and Chaib-draa (2014)	
Muelling et al. (2013)	Neu and Szepesvári (2009)	
Muelling et al. (2014)	Rhinehart and Kitani (2018)	
Shimosaka et al. (2016)	Mainprice et al. (2015)	
Kuderer et al. (2015)		
Sun et al. (2018)		
Gao et al. (2018)		
You et al. (2019)		
Yu et al. (2019)		
Li et al. (2018)		
Lee and Popović (2010)		
Mathe and Sminchisescu (2013)		
Imani and Braga-Neto (2018)		

4.1 Mimic the expert

One of the first applications of IRL was to control an aerobatic helicopter (Abbeel et al. 2007). The maximum margin formulation from Abbeel and Ng (2004) is used to estimate the weights of the features from examples provided by a human pilot performing the desired maneuvers. Once the reward weights are estimated, differential dynamic programming is used to estimate a policy for remotely controlling the helicopter. The authors demonstrate that the proposed method can learn and perform four maneuvers: roll, flip, tail-in funnel, and nose-in funnel. Although not IRL, the concepts of this work are later extended to the case where the provided trajectories are sub-optimal and the dynamics of the model need to be learned (Abbeel et al. 2010; Coates et al. 2008). However, in Coates et al. (2009) an IRL step is included to learn the weights of the quadratic reward function.

As a generalization of the helicopter application, IRL is often used to learn the appropriate reward function for a robot. The literature argues that hard coding each action of a robot is inefficient and limits the robot to the hard-coded action set. IRL, on the other hand, allows for a robot to learn a reward function from human demonstration. This allows the robot to act in a more human fashion and adapt to changing environments. Inga et al. (2017) argue that a different reward function for each human controller should

be learned so that individual characteristic models can be constructed. In other early applications to robotics, IRL algorithms are used to learn controllers for the impedance signals of actuators from human demonstration (Howard et al. 2010; Mori et al. 2011).

Lopes et al. (2007) argue that a robot must be able to interpret the demonstrator's actions and states, i.e. there must be a translation between the demonstrator's action and state space and the robot's action and state space. This is a concept called affordance (Gibson 2014). The robot's state and action spaces may be limited by its capabilities. Once an affordance model is learned, Lopes et al. use the Bayesian IRL framework to learn a reward function. Adaptive-state graphs and Bayesian IRL are later used for online learning of the state and action spaces and the reward function (Okal and Arras 2016; Okal et al. 2015).

Lopes et al. (2011) investigate the scenario where a teacher provides feedback to the robot. The feedback is unstructured, meaning it can contain information about the correctness of the action selected by the robot, the name of the correct action, or a synonym for the correct action. The robot must learn a model for the feedback system as well as an estimate for the reward function. The Bayesian IRL framework is implemented for this application.

IRL is often used to learn reward functions for robots that will be interacting with a changing environment, such as navigating a path through human pedestrians. Kim and Pineau (2016) utilize the MAP-BIRL algorithm from Choi and Kim (2011) to estimate reward functions from human demonstrations for socially adaptive path planning in an unknown and dynamic environment. Choi et al. (2018) use maximum margin IRL to learn the reward function for path planning and use the learned reward function to predict future paths of agents. Pflueger et al. (2019) combine IRL with value iteration networks for path planning of planetary rovers.

Krishnan et al. (2018) present SWIRL, a sequential window IRL technique for controlling a robot when the rewards are delayed. In this scenario, the demonstrations are limited to only the states visited by the expert, and the corresponding actions are not provided to the learner. SWIRL is composed of three sub-algorithms. The first uses hierarchical clustering to partition the observed demonstrations into subtasks. The second uses IRL to estimate the reward function for each subtask. The third solves the forward problem and returns a policy. Kirshnan et al. present experiments that demonstrate that SWIRL requires less rollouts for learning a policy than standard RL algorithms.

IRL has also been used to teach computers and robots to play games that require sequential and complex decisions. In table tennis, a player cannot win a game by simply hitting the ball in the same spot each time. A player must develop an effective strategy to beat their opponent. IRL was successfully utilized to learn the reward function for playing table tennis from demonstrations (Muelling et al. 2013, 2014).

With the growing popularity of autonomous vehicles, IRL has also been applied to learn the reward functions for controlling cars from expert demonstration. Shimosaka et al. (2016) investigate defensive driving on residential roads. In their IRL implementation, the authors minimize the negative log likelihood of the demonstrated trajectories and use an L_1 regularization term to select features. Kuderer et al. (2015) use IRL to learn reward functions that reflect a driver's preferences, such as a relaxed driving style or high acceleration. Sun et al. (2018) use hierarchical IRL to learn reward functions for autonomous cars interacting with other agents (cars and pedestrians) on the road. Gao et al. (2018) use IRL to learn reward functions for car-following behavior. You et al. (2019) develop a new MDP formulation for representing driving in traffic and use deep maximum entropy IRL to learn the reward function for drivers.

Yu et al. (2019) use Bayesian IRL to learn the reward function of doctors managing mechanical ventilation of patients and their sedation while being ventilated. The purpose of their study is to develop an RL agent that can balance the risks of long-term mechanical ventilation with removing the ventilation too quickly from a patient. A reward function that encodes this information is difficult to hand craft and therefore should be learned from expert demonstrations. In another health-care application, Li et al. (2018) use function approximation and IRL to learn the reward function of paraplegic patients performing tasks following a physician's instructions.

Motion controllers are used in animation to control the movement of characters. Some of these motion controllers are RL-based, but rely on hand-crafting a reward function to produce the desired motion, which can be tedious. Lee and Popović (2010) propose using IRL to learn an appropriate reward function from demonstrated behavior. The proposed algorithm is applied to three different motion controller problems each containing three different styles. For example, on the simple navigation problem where the agent must move to a specified location, the three demonstrated styles are walking normally, walking backwards, and running. Similarly, Mathe and Sminchisescu (2013) use IRL to learn the reward function for eye movements.

Imani and Braga-Neto (2018) use the Bayesian IRL framework to control gene-regulatory networks in the situation where the only information about the gene network, i.e. the measurements and corresponding interventions, is provided by an expert in an experimental setting.

4.2 Interacting with other systems

The applications discussed in this section use IRL to estimate the reward of a system in order to improve an agent's interaction with that system. In one of the first applications of IRL in robotics, Ziebart et al. (2009) use maximum entropy IRL to predict pedestrians' future locations so that a robot can plan a path that avoids them. Similarly, Chung et al. (2010) implement the maximum margin framework to estimate the reward functions for pedestrians. The robot then uses these reward functions to estimate a pedestrian's path. Kuderer et al. (2012) and Kuderer et al. (2013) use the maximum entropy IRL concept to estimate the trajectories of pedestrians so that a robot navigating the environment can interact with the human in a socially compliant fashion. The maximum entropy concept is later extended to the situation where both the discrete actions of the robot and the continuous trajectories of the pedestrians are incorporated into the decision model (Kretzschmar et al. 2014, 2016). Pfeiffer et al. (2016) apply the maximum entropy method to a real-world scenario where the pedestrians are unaware of the experiment. Shkurti et al. (2018) use maximum entropy IRL to learn a reward function for following a target as opposed to avoiding pedestrians.

Haptic assistance provides feedback to a human controlling a system through the control interface in the form of torque or force. This concept is especially useful for teleoperations where the human is remotely controlling the system. In assistive teleoperations, the system is controlled by both a human operator and an autonomous agent. The agent uses estimates of the human's intent to influence the human's control over the system. Scobee et al. (2018) use max-margin IRL to develop a haptic assistance system for 2D driving.

Dialogue systems allow a human to interact with a machine through spoken word. The goal of a dialogue system is to respond correctly to a user's inquiry, therefore the policy to be learned selects the correct response to a series of questions or a series of verbal

interactions. Chandramohan et al. (2011) use the maximum margin IRL algorithm to learn reward functions for a dialogue system and demonstrate the method on a town information system. Other work on dialogue systems assume that the state is uncertain and formulate the model as a POMDP (Boularias et al. 2010; Chinaei and Chaib-Draa 2014; Chinaei and Chaib-draa 2014). In a similar natural language processing context, IRL has been used to train a grammar parser (Neu and Szepesvári 2009).

Rhinehart and Kitani (2018) developed an online maximum entropy IRL algorithm that learns the reward function of a human wearing a first-person camera and estimates the wearer's long-term goals. In a similar application, IRL is used to learn the reward function for human movement so that a robot can predict a human collaborator's motion (Mainprice et al. 2015).

4.3 Learn about the system

The applications described in this section use the estimated reward function to learn about a system. Elnaggar and Bezzo (2018) use IRL to identify the goal of sensor-spoofing cyber attacks on cyber-physical systems. They use the Bayesian IRL formulation and demonstrate the proposed method on a simulation involving a drone.

IRL has been applied to learn about the reward structure of traders in finance. Yang et al. (2015) use IRL as a feature extraction technique for classifying algorithmic trading strategies. In this application, the estimated rewards are used as features for either a supervised or unsupervised classification algorithm. The objective is to determine the class of trader based on the estimated rewards. The demonstrated method outperforms supervised learning problems that simply use trading summary statistics. Yang et al. (2018) use IRL to learn the reward of the market based on investor sentiment. The estimated reward is later used to predict market direction assuming that the MDP is stationary. IRL has also been used to inform behavior rules for agent-based models (Lee et al. 2017). This technique was later used to predict the price movement of Bitcoin (Lee et al. 2018).

Most of the IRL applications that have been discussed up to this point focus on extracting reward functions from human experts so that they may be applied in another setting. However, Yamaguchi et al. (2018) use IRL to identify animal behavior, and Hirakawa et al. (2018) use IRL to predict the movement of animals.

5 Conclusions and discussion of future research areas

This survey organizes and categorizes the IRL literature. After defining the IRL problem, it outlines the methods for IRL and segments them into two broad categories. The first is IRL algorithms, which represent the bulk of the literature and is extremely developed. The second category is extensions of the basic IRL paradigm and represents an area of the literature that needs more development in order for IRL to become widely applicable on a variety of real-world problems. In particular, we believe that it is essential to develop more model-free methods. This survey also organizes and categorizes the IRL literature on applications. These categories are defined by the objective of the method: learn to mimic an expert, learn to assist the human or expert, or learn about the system under study. Most of the literature on applications of IRL focuses on teaching robots to act more like the expert.

There are several areas of machine learning that could be integrated into the IRL literature that have yet to be explored. For example, IRL solution methods have yet to be well-studied in the context of transfer learning (Pan and Yang 2009) and multi-view learning (Xu et al. 2013; Zhao et al. 2017). Transfer learning is underexplored in the IRL literature, but it has been successful in reducing training time and cost elsewhere in machine learning. IRL inherently learns a transferable model in the sense that it abstracts away from particular dynamics. Bayesian IRL methods can make use of priors, and many IRL methods can make use of sub-optimal examples. Such aspects of the IRL literature address some aspects of transfer learning, but not all. Transfer learning for RL has a rich literature (Taylor and Stone 2009) in comparison to transfer learning for IRL. In particular, IRL methods for handling examples from different sample spaces are relatively underexplored.

Multi-view learning could be useful for IRL by allowing demonstrations to be represented by a more varied set of inputs. Multi-view learning considers learning from multiple sources of data at once, such as multiple camera views, or a combination of photos, videos, and internet traffic data. The learned algorithm *views* all sources simultaneously as inputs when generating predictions or actions. It has been studied in the context of RL indirectly as distributed RL (Weiß 1995), with recent multi-view applications in deep RL (Barati et al. 2019). Multi-view IRL literature is sparse (Dimitrakakis et al. 2017), and, in the large, the various multi-view solution methods (Xu et al. 2013) have not been related to IRL.

A key challenge to these extensions, and the progression of IRL more generally, is finding and creating appropriate data and test-beds for validation. This is true for both classical and deep learning methods. The RL community's approach to the use of board and video games can serve as a guidepost. Whether such games will be as useful for the development of IRL is unknown. However, it is clear that those test-beds allowed for community-wide goal-setting, validation of progress, and benchmarking. Such a suite of test-beds has not been widely adopted in IRL research, and currently represents a bottleneck.

References

Abbeel P, Coates A, Ng AY (2010) Autonomous helicopter aerobatics through apprenticeship learning. Int J Robot Res 29(13):1608–1639

Abbeel P, Coates A, Quigley M, Ng AY (2007) An application of reinforcement learning to aerobatic helicopter flight. In: Advances in neural information processing systems. pp 1–8

Abbeel P, Dolgov D, Ng AY, Thrun S (2008) Apprenticeship learning for motion planning with application to parking lot navigation. In: 2008 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 1083–1090

Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the twenty-first international conference on Machine learning. ACM, p 1

Aghasadeghi N, Bretl T (2011) Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals. In: Intelligent robots and systems (IROS), 2011 IEEE/RSJ international conference on. IEEE, pp 1561–1566
Aghasadeghi N, Long A, Bretl T (2012) Inverse optimal control for a hybrid dynamical system with impacts. In: 2012 IEEE international conference on robotics and automation. IEEE, pp 4962–4967
Amin K, Jiang N, Singh S (2017) Repeated inverse reinforcement learning. In: Advances in neural information processing systems. pp 1815–1824
Argall BD, Chernova S, Veloso M, Browning B (2009) A survey of robot learning from demonstration. Robot Auton Syst 57(5):469–483
Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA (2017) Deep reinforcement learning: a brief survey. IEEE Signal Process Mag 34(6):26–38
Audiffren J, Valko M, Lazaric A, Ghavamzadeh M (2015) Maximum entropy semi-supervised inverse reinforcement learning. In: IJCAI. pp 3315–3321
Azar MG, Gómez V, Kappen HJ (2012) Dynamic policy programming. J Mach Learn Res 13:3207–3245
Babes M, Marivate V, Subramanian K, Littman ML (2011) Apprenticeship learning about multiple intentions. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp 897–904
Barati E, Chen X, Zhong Z (2019) Attention-based deep reinforcement learning for multi-view environments. In: Proceedings of the 18th international conference on autonomous agents and multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 1805–1807
Bloem M, Bambos N (2014) Infinite time horizon maximum causal entropy inverse reinforcement learning. In: Decision and control (CDC), 2014 IEEE 53rd annual conference on. IEEE, pp 4911–4916
Bogert K, Doshi P (2014) Multi-robot inverse reinforcement learning under occlusion with interactions. In: Proceedings of the 2014 international conference on autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 173–180
Bogert K, Doshi P (2017) Scaling expectation-maximization for inverse reinforcement learning to multiple robots under occlusion. In: Proceedings of the 16th conference on autonomous agents and multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 522–529
Bogert K, Lin JFS, Doshi P, Kulic D (2016) Expectation-maximization for inverse reinforcement learning with hidden data. In: Proceedings of the 2016 international conference on autonomous agents & multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 1034–1042
Bogert KD, Doshi P (2015) Toward estimating others' transition models under occlusion for multi-robot IRL. In: IJCAI. pp 1867–1873
Boularias A, Chaib-Draa B (2013) Apprenticeship learning with few examples. Neurocomputing 104:83–96
Boularias A, Chinaei HR, Chaib-draa B (2010) Learning the reward model of dialogue POMDPs from data. In: NIPS workshop on machine learning for assistive techniques. Citeseer
Boularias A, Kober J, Peters J (2011) Relative entropy inverse reinforcement learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 182–189
Boularias A, Krömer O, Peters J (2012) Structured apprenticeship learning. Joint European conference on machine learning and knowledge discovery in databases. Springer, New York, pp 227–242
Brown DS, Niekum S (2018) Efficient probabilistic performance bounds for inverse reinforcement learning. In: Thirty-Second AAAI conference on artificial intelligence
Budhraja KK, Oates T (2017) Neuroevolution-based inverse reinforcement learning. In: Evolutionary computation (CEC), 2017 IEEE congress on. IEEE, pp 67–76
Byravan A, Monfort M, Ziebart BD, Boots B, Fox D (2015) Graph-based inverse optimal control for robot manipulation. Ijcai 15:1874–1890
Cai XS, Han ZZ (2005) Inverse optimal control of nonlinear systems with structural uncertainty. IEE Proc-Control Theory Appl 152(1):79–83
Cakmak M, Thomaz A (2011) Active learning with mixed query types in learning from demonstration. In: Proc. of the ICML workshop on new developments in imitation learning
Calinon S, D'halluin F, Sauser EL, Caldwell DG, Billard AG, (2010) Learning and reproduction of gestures by imitation. IEEE Robot Autom Mag 17(2):44–54
Calinon S, Guenter F, Billard A (2007) On learning, representing, and generalizing a task in a humanoid robot. IEEE Trans Syst Man Cybern Part B (Cybernetics) 37(2):286–298
Chandramohan S, Geist M, Lefevre F, Pietquin O (2011) User simulation in dialogue systems using inverse reinforcement learning. Interspeech 2011:1025–1028
Chen X, Monfort M, Ziebart BD, Carr P (2016) Adversarial inverse optimal control for general imitation learning losses and embodiment transfer. In: UAI

Chen Xl, Cao L, Xu Zx, Lai J, Li Cx (2019) A study of continuous maximum entropy deep inverse reinforcement learning. Math Probl Eng
Chib S, Greenberg E (1995) Understanding the Metropolis-Hastings algorithm. Am Stat 49(4):327–335
Chinaei H, Chaib-draa B (2014) Dialogue POMDP components (part II): learning the reward function. Int J Speech Technol 17(4):325–340
Chinaei HR, Chaib-Draa B (2012) An inverse reinforcement learning algorithm for partially observable domains with application on healthcare dialogue management. In: Machine learning and applications (ICMLA), 2012 11th international conference on, vol 1. IEEE, pp 144–149
Chinaei HR, Chaib-Draa B (2014) Dialogue POMDP components (part I): learning states and observations. Int J Speech Technol 17(4):309–323
Choi D, An TH, Ahn K, Choi J (2018) Future trajectory prediction via RNN and maximum margin inverse reinforcement learning. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA). IEEE, pp 125–130
Choi J, Kim KE (2011) Inverse reinforcement learning in partially observable environments. J Mach Learn Res 12:691–730
Choi J, Kim KE(2011) MAP inference for Bayesian inverse reinforcement learning. In: Advances in neural information processing systems. pp 1989–1997
Choi J, Kim KE (2012) Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In: Advances in neural information processing systems. pp 305–313
Choi J, Kim KE (2013) Bayesian nonparametric feature construction for inverse reinforcement learning. In: IJCAI. pp 1287–1293
Chung SY, Huang HP (2010) A mobile robot that understands pedestrian spatial behaviors. In: 2010 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 5861–5866
Coates A, Abbeel P, Ng AY (2008) Learning for control from multiple demonstrations. In: Proceedings of the 25th international conference on Machine learning. ACM, pp 144–151
Coates A, Abbeel P, Ng AY (2009) Apprenticeship learning for helicopter control. Commun ACM 52(7):97–105
Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodological). pp 1–38
Dimitrakakis C, Parkes DC, Radanovic G, Tylkin P (2017) Multi-view decision processes: the helper-AI problem. In: Advances in neural information processing systems. pp 5443–5452
Dimitrakakis C, Rothkopf CA (2011) Bayesian multitask inverse reinforcement learning. European workshop on reinforcement learning. Springer, pp 273–284
Doerr A, Ratliff ND, Bohg J, Toussaint M, Schaal S (2015) Direct loss minimization inverse optimal control. In: Robotics: science and systems
Duan Y, Chen X, Houthooft R, Schulman J, Abbeel P (2016) Benchmarking deep reinforcement learning for continuous control. In: International conference on machine learning. pp 1329–1338
Dvijotham K, Todorov E (2010) Inverse optimal control with linearly-solvable MDPs. In: Proceedings of the 27th International conference on machine learning (ICML-10). pp 335–342
El Asri L, Piot B, Geist M, Laroche R, Pietquin O (2016) Score-based inverse reinforcement learning. In: Proceedings of the 2016 international conference on autonomous agents & multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 457–465
Elnaggar M, Bezzo N (2018) An IRL approach for cyber-physical attack intention prediction and recovery. In: 2018 Annual American control conference (ACC). IEEE, pp. 222–227
Ezzeddine A, Mourad N, Araabi BN, Ahmadabadi MN (2018) Combination of learning from non-optimal demonstrations and feedbacks using inverse reinforcement learning and Bayesian policy improvement. Expert Syst Appl
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
Gao H, Shi G, Xie G, Cheng B (2018) Car-following method based on inverse reinforcement learning for autonomous vehicle decision-making. Int J Adv Rob Syst 15(6):1729881418817162
Garcıa J, Fernández F (2015) A comprehensive survey on safe reinforcement learning. J Mach Learn Res 16(1):1437–1480
Ghavamzadeh M, Mannor S, Pineau J, Tamar A (2015) Bayesian reinforcement learning: a survey. Found Trends® Mach Learn 8(5-6):359–483
Gibson JJ (2014) The ecological approach to visual perception, classic. Psychology Press, Hove
Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems. pp 2672–2680
Griffiths TL, Jordan MI, Tenenbaum JB, Blei DM (2004) Hierarchical topic models and the nested chinese restaurant process. In: Advances in neural information processing systems. pp 17–24

Hadfield-Menell D, Russell SJ, Abbeel P, Dragan A (2016) Cooperative inverse reinforcement learning. In: Advances in neural information processing systems. pp 3909–3917
Hahn J, Zoubir AM (2015) Inverse reinforcement learning using expectation maximization in mixture models. In: Acoustics, speech and signal processing (ICASSP), 2015 IEEE international conference on. IEEE, pp 3721–3725
Hausman K, Chebotar Y, Schaal S, Sukhatme G, Lim JJ (2017) Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In: Advances in neural information processing systems. pp 1235–1245
Henderson P, Chang WD, Bacon PL, Meger D, Pineau J, Precup D (2018) OptionGAN: Learning joint reward-policy options using generative adversarial inverse reinforcement learning. In: Thirty-second AAAI conference on artificial intelligence
Herman M, Gindele T, Wagner J, Schmitt F, Burgard W (2016) Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In: Artificial intelligence and statistics. pp 102–110
Herman M, Gindele T, Wagner J, Schmitt F, Burgard W (2016) Simultaneous estimation of rewards and dynamics from noisy expert demonstrations. In: European Symposium on artificial intelligence, computational intelligence and machine learning (ESANN 2016)
Hirakawa T, Yamashita T, Tamaki T, Fujiyoshi H, Umezu Y, Takeuchi I, Matsumoto S, Yoda K (2018) Can AI predict animal movements? filling gaps in animal trajectories using inverse reinforcement learning. Ecosphere 9(10):e02447
Ho J, Ermon S (2016) Generative adversarial imitation learning. In: Advances in neural information processing systems. pp 4565–4573
Ho J, Gupta J, Ermon S (2016) Model-free imitation learning with policy optimization. In: International conference on machine learning. pp 2760–2769
Howard M, Mitrovic D, Vijayakumar S (2010) Transferring impedance control strategies between heterogeneous systems via apprenticeship learning. In: 2010 10th IEEE-RAS international conference on humanoid robots. IEEE, pp 98–105
Hussein A, Gaber MM, Elyan E, Jayne C (2017) Imitation learning: a survey of learning methods. ACM Comput Surv (CSUR) 50(2):21
Hussein M, Mohammed Y, Ali SA (2015) Learning from demonstration using variational Bayesian inference. In: International conference on industrial, engineering and other applications of applied intelligent systems. Springer, pp 371–381
Imani M, Braga-Neto U (2018) Control of gene regulatory networks using Bayesian inverse reinforcement learning. IEEE/ACM Trans Comput Biol Bioinform 16(4):1250–1261
Inga J, Köpf F, Flad M, Hohmann S (2017) Individual human behavior identification using an inverse reinforcement learning method. In: 2017 IEEE international conference on systems, man, and cybernetics (SMC). IEEE, pp 99–104
Kalakrishnan M, Pastor P, Righetti L, Schaal S (2013) Learning objective functions for manipulation. In: Robotics and automation (ICRA), 2013 IEEE international conference on. IEEE, pp 1331–1336
Kalakrishnan M, Theodorou E, Schaal S (2010) Inverse reinforcement learning with PI^2. In: The snowbird workshop, submitted to. Citeseer
Kalman RE (1964) When is a linear control system optimal? J Basic Eng 86(1):51–60
Kanazawa M, Nakaura S, Sampei M (2009) Inverse optimal control problem for bilinear systems: Application to the inverted pendulum with horizontal and vertical movement. In: Proceedings of the 48h IEEE conference on decision and control (CDC) held jointly with 2009 28th Chinese control conference. IEEE, pp 2260–2267
Kangasrääsiö A, Kaski S (2018) Inverse reinforcement learning from summary data. Mach Learn 107(8–10):1517–1535
Kappen HJ (2005) Linear theory for control of nonlinear stochastic systems. Phys Rev Lett 95(20):200201
Kim B, Pineau J (2016) Socially adaptive path planning in human environments using inverse reinforcement learning. Int J Soc Robot 8(1):51–66
Kitani KM, Ziebart BD, Bagnell JA, Hebert M(2012) Activity forecasting. In: European conference on computer vision. Springer, pp 201–214
Klein E, Geist M, Pietquin O (2011) Reducing the dimentionality of the reward space in the inverse reinforcement learning problem. In: Proceedings of the IEEE workshop on machine learning algorithms, systems and applications (MLASA 2011). Honolulu (USA). Citeseer
Klein E, Geist M, Piot B, Pietquin O (2012) Inverse reinforcement learning through structured classification. In: Advances in neural information processing systems. pp 1007–1015

Klein E, Piot B, Geist M, Pietquin O (2013) A cascaded supervised learning approach to inverse reinforcement learning. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 1–16

Kretzschmar, H., Kuderer, M., Burgard, W. (2014) Learning to predict trajectories of cooperatively navigating agents. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE, pp. 4015–4020

Kretzschmar H, Spies M, Sprunk C, Burgard W (2016) Socially compliant mobile robot navigation via inverse reinforcement learning. Int J Robot Res 35(11):1289–1307

Krishnan S, Garg A, Liaw R, Thananjeyan B, Miller L, Pokorny FT, Goldberg K (2018) SWIRL: a sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards. Int J Robot Res 0278364918784350

Krstic M, Li ZH (1998) Inverse optimal design of input-to-state stabilizing nonlinear controllers. IEEE Trans Autom Control 43(3):336–350

Krstic M, Tsiotras P (1999) Inverse optimal stabilization of a rigid spacecraft. IEEE Trans Autom Control 44(5):1042–1049

Kuderer M, Gulati S, Burgard W (2015) Learning driving styles for autonomous vehicles from demonstration. In: 2015 IEEE international conference on robotics and automation (ICRA). IEEE, pp 2641–2646

Kuderer M, Kretzschmar H, Burgard W (2013) Teaching mobile robots to cooperatively navigate in populated environments. In: 2013 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 3138–3143

Kuderer M, Kretzschmar H, Sprunk C, Burgard W (2012) Feature-based prediction of trajectories for socially compliant navigation. In: Robotics: science and systems

Kunapuli G, Odom P, Shavlik JW, Natarajan S (2013) Guiding autonomous agents to better behaviors through human advice. In: Data mining (ICDM), 2013 IEEE 13th international conference on. IEEE, 409–418

Lee, K., Choi, S., Oh, S (2016) Inverse reinforcement learning with leveraged Gaussian processes. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 3907–3912

Lee K, Rucker M, Scherer WT, Beling PA, Gerber MS, Kang H (2017) Agent-based model construction using inverse reinforcement learning. In: 2017 Winter simulation conference (WSC). IEEE, pp 1264–1275

Lee K, Ulkuatam S, Beling P, Scherer W (2018) Generating synthetic bitcoin transactions and predicting market price movement via inverse reinforcement learning and agent-based modeling. J Artif Soc Soc Simul 21(3)

Lee SJ, Popović Z (2010) Learning behavior styles with inverse reinforcement learning. ACM Trans Graph (TOG) 29(4):122

Levine S, Koltun V (2012) Continuous inverse optimal control with locally optimal examples. In: Proceedings of the 29th international coference on international conference on machine learning. Omnipress, pp 475–482

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

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

Li K, Rath M, Burdick JW (2018) Inverse reinforcement learning via function approximation for clinical motion analysis. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 610–617

Lin X, Adams SC, Beling PA (2019) Multi-agent inverse reinforcement learning for certain general-sum stochastic games. J Artif Intell Res 66:473–502

Lin X, Beling PA, Cogill R (2018) Multiagent inverse reinforcement learning for two-person zero-sum games. IEEE Trans Games 10(1):56–68

Lopes M, Cederbourg T, Oudeyer PY (2011) Simultaneous acquisition of task and feedback models. In: 2011 IEEE international conference on development and learning (ICDL), vol 2. IEEE, pp 1–7

Lopes M, Melo F, Montesano L (2009) Active learning for reward estimation in inverse reinforcement learning. In: Joint european conference on machine learning and knowledge discovery in databases. Springer, pp 31–46

Lopes M, Melo FS, Montesano L (2007) Affordance-based imitation learning in robots. In: 2007 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 1015–1021

Luo W, Chu YC, Ling KV (2005) Inverse optimal adaptive control for attitude tracking of spacecraft. IEEE Trans Autom Control 50(11):1639–1654

Mainprice J, Hayne R, Berenson D (2015) Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning. In: 2015 IEEE international conference on robotics and automation (ICRA). IEEE, pp 885–892

Majumdar A, Singh S, Mandlekar A, Pavone M (2017) Risk-sensitive inverse reinforcement learning via coherent risk models. In: Robotics: science and systems

Makino T, Takeuchi J (2012) Apprenticeship learning for model parameters of partially observable environments. In: Proceedings of the 29th international coference on international conference on machine learning. Omnipress, pp 891–898

Mathe S, Sminchisescu C (2013) Action from still image dataset and inverse optimal control to learn task specific visual scanpaths. In: Advances in neural information processing systems. pp 1923–1931

Melo FS, Lopes M (2010) Learning from demonstration using MDP induced metrics. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 385–401

Mendez JAM, Shivkumar S, Eaton E (2018) Lifelong inverse reinforcement learning. In: Advances in neural information processing systems, pp 4502–4513

Michini B, Cutler M, How JP (2013) Scalable reward learning from demonstration. In: Robotics and automation (ICRA), 2013 IEEE international conference on. IEEE, pp 303–308

Michini B, How JP (2012) Bayesian nonparametric inverse reinforcement learning. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, p. 148–163

Michini B, How JP (2012) Improving the efficiency of Bayesian inverse reinforcement learning. In: Robotics and automation (ICRA), 2012 IEEE international conference on IEEE, pp 3651–3656

Michini B, Walsh TJ, Agha-Mohammadi AA, How JP (2015) Bayesian nonparametric reward learning from demonstration. IEEE Trans Rob 31(2):369–386

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

Mori T, Howard M, Vijayakumar S (2011) Model-free apprenticeship learning for transfer of human impedance behaviour. In: 2011 11th IEEE-RAS international conference on humanoid robots. IEEE, pp 239–246

Muelling K, Boularias A, Mohler B, Schölkopf B, Peters J (2013) Inverse reinforcement learning for strategy extraction. In: ECML PKDD 2013 workshop on machine learning and data mining for sports analytics (MLSA 2013). pp 1–9

Muelling K, Boularias A, Mohler B, Schölkopf B, Peters J (2014) Learning strategies in table tennis using inverse reinforcement learning. Biol Cybern 108(5):603–619

Nakamura N, Nakamura H, Nishitani H (2011) Global inverse optimal control with guaranteed convergence rates of input affine nonlinear systems. IEEE Trans Autom Control 56(2):358–369

Nash JF (1950) Equilibrium points in n-person games. Proc Natl Acad Sci 36(1):48–49

Natarajan S, Kunapuli G, Judah K, Tadepalli P, Kersting K, Shavlik J (2010) Multi-agent inverse reinforcement learning. In: 2010 Ninth international conference on machine learning and applications. IEEE, pp 395–400

Neu G, Szepesvári C (2007) Apprenticeship learning using inverse reinforcement learning and gradient methods. In: Proceedings of the twenty-third conference on uncertainty in artificial intelligence. AUAI Press, pp 295–302

Neu G, Szepesvári C (2009) Training parsers by inverse reinforcement learning. Mach Learn 77(2–3):303

Ng AY, Russell SJ(2000) Algorithms for inverse reinforcement learning. In: ICML. pp 663–670

Nguyen QP, Low BKH, Jaillet P (2015) Inverse reinforcement learning with locally consistent reward functions. In: Advances in neural information processing systems. pp 1747–1755

Odom P, Natarajan S (2016) Active advice seeking for inverse reinforcement learning. In: Proceedings of the 2016 international conference on autonomous agents & multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 512–520

Okal B, Arras KO (2016) Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 2889–2895

Okal B, Gilbert H, Arras KO (2015) Efficient inverse reinforcement learning using adaptive state-graphs. In: Learning from demonstration: inverse optimal control, reinforcement learning and lifelong learning workshop at robotics: science and systems (RSS), Rome, Italy

Ornelas F, Sanchez EN, Loukianov AG (2010) Discrete-time inverse optimal control for nonlinear systems trajectory tracking. In: 49th IEEE conference on decision and control (CDC). IEEE, pp 4813–4818

Ornelas F, Sanchez EN, Loukianov A.G (2011) Discrete-time nonlinear systems inverse optimal control: a control Lyapunov function approach. In: 2011 IEEE international conference on control applications (CCA). IEEE, pp 1431–1436
Ornelas-Tellez F, Sanchez EN, Loukianov AG (2012) Discrete-time neural inverse optimal control for nonlinear systems via passivation. IEEE Trans Neural Netw Learn Syst 23(8):1327–1339
Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
Pan X, Shen Y (2018) Human-interactive subgoal supervision for efficient inverse reinforcement learning. In: Proceedings of the 17th international conference on autonomous agents and multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 1380–1387
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: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2096–2101
Pflueger M, Agha A, Sukhatme GS (2019) Rover-IRL: inverse reinforcement learning with soft value iteration networks for planetary rover path planning. IEEE Robot Autom Lett 4(2):1387–1394
Piot B, Geist M, Pietquin O (2013) Learning from demonstrations: is it worth estimating a reward function?. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp.17–32
Piot B, Geist M, Pietquin O (2014) Boosted and reward-regularized classification for apprenticeship learning. In: Proceedings of the 2014 international conference on autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 1249–1256
Piot B, Geist M, Pietquin O (2017) Bridging the gap between imitation learning and inverse reinforcement learning. IEEE Trans Neural Netw Learn Syst 28(8):1814–1826
Powell M (1998) Direct search algorithms for optimization calculations. Acta Numer 7:287–336
Puterman ML (2014) Markov decision processes: discrete stochastic dynamic programming. Wiley, New Jersey
Qiao Q, Beling PA (2011) Inverse reinforcement learning with Gaussian process. In: American control conference (ACC). IEEE, pp 113–118
Qiao Q, Beling PA(2013) Recognition of agents based on observation of their sequential behavior. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 33–48
Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
Ramachandran D, Amir E (2007) Bayesian inverse reinforcement learning. In: IJCAI, vol 7. pp 2586–2591
Ranchod P, Rosman B, Konidaris G (2015) Nonparametric bayesian reward segmentation for skill discovery using inverse reinforcement learning. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 471–477
Ratliff N, Bagnell JA, Srinivasa SS (2007) Imitation learning for locomotion and manipulation. In: 2007 7th IEEE-RAS international conference on humanoid robots. IEEE, pp 392–397
Ratliff N, Bradley D, Bagnell JA, Chestnutt J (2006) Boosting structured prediction for imitation learning. In: Proceedings of the 19th international conference on neural information processing systems. MIT Press, pp 1153–1160
Ratliff N, Ziebart B, Peterson K, Bagnell JA, Hebert M, Dey AK, Srinivasa S (2009) Inverse optimal heuristic control for imitation learning. In: Artificial intelligence and statistics. pp 424–431
Ratliff ND, Bagnell JA, Zinkevich MA (2006) Maximum margin planning. In: Proceedings of the 23rd international conference on machine learning. pp 729–736
Ratliff ND, Silver D, Bagnell JA (2009) Learning to search: functional gradient techniques for imitation learning. Auton Robot 27(1):25–53
Reddy TS, Gopikrishna V, Zaruba G, Huber M (2012) Inverse reinforcement learning for decentralized noncooperative multiagent systems. In: Systems, man, and cybernetics (SMC), 2012 IEEE international conference on. IEEE, pp 1930–1935
Rhinehart N, Kitani K (2018) First-person activity forecasting from video with online inverse reinforcement learning. IEEE transactions on pattern analysis and machine intelligence
Rothkopf CA, Ballard D (2010) Credit assignment in multiple goal embodied visuomotor behavior. Front Psychol 1:173
Rothkopf CA, Ballard DH (2013) Modular inverse reinforcement learning for visuomotor behavior. Biol Cybern 107(4):477–490
Rothkopf CA, Dimitrakakis C (2011) Preference elicitation and inverse reinforcement learning. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 34–48
Ruiz-Cruz R, Sanchez EN, Ornelas-Tellez F, Loukianov AG, Harley RG (2013) Particle swarm optimization for discrete-time inverse optimal control of a doubly fed induction generator. IEEE Trans Cybern 43(6):1698–1709

Russell S (1998) Learning agents for uncertain environments. In: Proceedings of the eleventh annual conference on Computational learning theory. ACM, pp 101–103
Sadigh D, Dragan AD, Sastry S, Seshia SA (2017) Active preference-based learning of reward functions. In: Robotics: science and systems (RSS)
Scobee DR, Royo VR, Tomlin CJ, Sastry SS (2018) Haptic assistance via inverse reinforcement learning. In: 2018 IEEE international conference on systems, man, and cybernetics (SMC). IEEE, pp 1510–1517
Settles B (2012) Active learning. Synth Lect Artif Intell Mach Learn 6(1):1–114
Shiarlis K, Messias J, Whiteson S(2016) Inverse reinforcement learning from failure. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, pp. 1060–1068. International Foundation for Autonomous Agents and Multiagent Systems
Shimosaka M, Kaneko T, Nishi K (2016) Modeling risk anticipation and defensive driving on residential roads with inverse reinforcement learning. In: 17th international IEEE conference on intelligent transportation systems (ITSC). IEEE x, pp 1694–1700
Shimosaka M, Sato J, Takenaka K, Hitomi K (2017) Fast inverse reinforcement learning with interval consistent graph for driving behavior prediction. In: AAAI. pp 1532–1538
Shkurti F, Kakodkar N, Dudek G (2018) Model-based probabilistic pursuit via inverse reinforcement learning. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 7804–7811
Singh S, Lacotte J, Majumdar A, Pavone M (2018) Risk-sensitive inverse reinforcement learning via semi- and non-parametric methods. Int J Robot Res 37(13–14):1713–1740
Šošić A, KhudaBukhsh WR, Zoubir AM, Koeppl H(2017) Inverse reinforcement learning in swarm systems. In: Proceedings of the 16th conference on autonomous agents and multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 1413–1421
Šošic A, Zoubir AM, Koeppl H (2018) Inverse reinforcement learning via nonparametric subgoal modeling. In: AAAI spring symposium on data-efficient reinforcement learning
Šošić A, Zoubir AM, Rueckert E, Peters J, Koeppl H (2018) Inverse reinforcement learning via nonparametric spatio-temporal subgoal modeling. J Mach Learn Res 19(1):2777–2821
Stanley KO, Miikkulainen R (2002) Evolving neural networks through augmenting topologies. Evol Comput 10(2):99–127
Suay HB, Brys T, Taylor ME, Chernova S(2016) Learning from demonstration for shaping through inverse reinforcement learning. In: Proceedings of the 2016 international conference on autonomous agents & multiagent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp 429–437
Sugiyama M, Takeuchi I, Suzuki T, Kanamori T, Hachiya H, Okanohara D (2010) Least-squares conditional density estimation. IEICE Trans Inf Syst 93(3):583–594
Sun L, Zhan W, Tomizuka M (2018) Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning. In: 2018 21st international conference on intelligent transportation systems (ITSC). IEEE, pp 2111–2117
Sunnåker M, Busetto AG, Numminen E, Corander J, Foll M, Dessimoz C (2013) Approximate Bayesian computation. PLoS Comput Biol 9(1):e1002803
Surana A(2014) Unsupervised inverse reinforcement learning with noisy data. In: Decision and control (CDC), 2014 IEEE 53rd annual conference on. IEEE, pp 4938–4945
Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT press, Cambridge
Syed U, Bowling M, Schapire RE (2008) Apprenticeship learning using linear programming. In: Proceedings of the 25th international conference on machine learning. ACM, pp 1032–1039
Syed U, Schapire RE (2008) A game-theoretic approach to apprenticeship learning. In: Advances in neural information processing systems. pp 1449–1456
Syed U, Schapire R.E(2010) A reduction from apprenticeship learning to classification. In: Advances in neural information processing systems. pp 2253–2261
Tadepalli P, Ok D (1998) Model-based average reward reinforcement learning. Artif Intell 100(1–2):177–224
Taylor ME, Stone P (2009) Transfer learning for reinforcement learning domains: a survey. J Mach Learn Res 10:1633–1685
Todorov E(2007) Linearly-solvable Markov decision problems. In: Advances in neural information processing systems. pp 1369–1376
Tossou AC, Dimitrakakis C(2013) Probabilistic inverse reinforcement learning in unknown environments. In: Proceedings of the twenty-ninth conference on uncertainty in artificial intelligence. AUAI Press, pp 635–643
Toussaint M, Storkey A(2006) Probabilistic inference for solving discrete and continuous state Markov decision processes. In: Proceedings of the 23rd international conference on machine learning. ACM, pp 945–952
Uchibe E (2016) Deep inverse reinforcement learning by logistic regression. In: International conference on neural information processing. Springer, pp 23–31

Uchibe E (2018) Model-free deep inverse reinforcement learning by logistic regression. Neural Process Lett 47(3):891–905
Uchibe E, Doya K (2014) Inverse reinforcement learning using dynamic policy programming. In: Development and learning and epigenetic robotics (ICDL-Epirob), 2014 joint IEEE international conferences on. IEEE, pp 222–228
Valko M, Ghavamzadeh M, Lazaric A (2013) Semi-supervised apprenticeship learning. In: European workshop on reinforcement learning. pp 131–142
Weiß G(1995) Distributed reinforcement learning. In: The biology and technology of intelligent autonomous agents. Springer, pp 415–428
Woodworth B, Ferrari F, Zosa TE, Riek LD(2018) Preference learning in assistive robotics: Observational repeated inverse reinforcement learning. In: Machine learning for healthcare conference. pp 420–439
Wulfmeier M, Rao D, Wang DZ, Ondruska P, Posner I (2017) Large-scale cost function learning for path planning using deep inverse reinforcement learning. Int J Robot Res 36(10):1073–1087
Wulfmeier M, Wang DZ, Posner I(2016) Watch this: Scalable cost-function learning for path planning in urban environments. In: Intelligent robots and systems (IROS), 2016 IEEE/RSJ international conference on. IEEE, pp 2089–2095
Xu C, Tao D, Xu C(2013) A survey on multi-view learning. arXiv preprint arXiv:1304.5634
Yamaguchi S, Naoki H, Ikeda M, Tsukada Y, Nakano S, Mori I, Ishii S (2018) Identification of animal behavioral strategies by inverse reinforcement learning. PLoS Comput Biol 14(5):e1006122
Yang SY, Qiao Q, Beling PA, Scherer WT, Kirilenko AA (2015) Gaussian process-based algorithmic trading strategy identification. Quant Finance 15(10):1683–1703
Yang SY, Yu Y, Almahdi S (2018) An investor sentiment reward-based trading system using Gaussian inverse reinforcement learning algorithm. Expert Syst Appl 114:388–401
You C, Lu J, Filex D, Tsiotras P (2019) Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. Robot Auton Syst 114:1–18
Yu C, Liu J, Zhao H (2019) Inverse reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. BMC Med Inform Decis Mak 19(2):57
Yu L, Yu T, Finn C, Ermon S(2019) Meta-inverse reinforcement learning with probabilistic context variables. In: Advances in neural information processing systems. pp 11749–11760
Zhang X, Zhang K, Miehling E, Basar T (2019) Non-cooperative inverse reinforcement learning. In: Advances in neural information processing systems. pp 9482–9493
Zhao J, Xie X, Xu X, Sun S (2017) Multi-view learning overview: recent progress and new challenges. Inform Fusion 38:43–54
Zheng J, Liu S, Ni LM (2014) Robust Bayesian inverse reinforcement learning with sparse behavior noise. In: AAAI. pp 2198–2205
Zhifei S, Meng Joo E (2012) A survey of inverse reinforcement learning techniques. Int J Intell Comput Cybern 5(3):293–311
Zhou W, Li W (2018) Safety-aware apprenticeship learning. In: International conference on computer aided verification. Springer, pp 662–680
Zhou Z, Bloem M, Bambos N (2018) Infinite time horizon maximum causal entropy inverse reinforcement learning. IEEE Trans Autom Control 63(9):2787–2802
Zhu Y, Mottaghi R, Kolve E, Lim JJ, Gupta A , Fei-Fei L, Farhadi A(2017) Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: Robotics and automation (ICRA), 2017 IEEE international conference on. IEEE, pp 3357–3364
Ziebart B, Dey A, Bagnell JA(2012) Probabilistic pointing target prediction via inverse optimal control. In: Proceedings of the 2012 ACM international conference on intelligent user interfaces. ACM, pp 1–10
Ziebart BD, Bagnell JA, Dey AK (2013) The principle of maximum causal entropy for estimating interacting processes. IEEE Trans Inf Theory 59(4):1966–1980
Ziebart BD, Maas AL, Bagnell JA, Dey AK (2008) Maximum entropy inverse reinforcement learning. AAAI 8:1433–1438 (**Chicago, IL, USA**)
Ziebart BD, Maas AL, Bagnell JA, Dey AK(2009) Human behavior modeling with maximum entropy inverse optimal control. In: AAAI spring symposium: human behavior modeling. p 92
Ziebart BD, Ratliff N, Gallagher G, Mertz C, Peterson K, Bagnell JA, Hebert M, Dey AK, Srinivasa, S (2009) Planning-based prediction for pedestrians. In: 2009 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 3931–3936
Zou Q, Li H, Zhang R (2018) Inverse reinforcement learning via neural network in driver behavior modeling. In: 2018 IEEE intelligent vehicles symposium (IV). IEEE, pp 1245–1250

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com