Generalized Maximum Causal Entropy for Inverse Reinforcement Learning

Tien Mai[1]*, **Kennard Chan**[2], **Patrick Jaillet** [3]
[1]Singapore-MIT Alliance for Research and Technology
[2]Nanyang Techological University
[3]ORC, EECS, Massachusetts Institute of Technology

Abstract

We consider the problem of learning from demonstrated trajectories with inverse reinforcement learning (IRL). Motivated by a limitation of the classical maximum entropy model (Ziebart, Bagnell, and Dey 2010) in capturing the structure of the network of states, we propose an IRL model based on a generalized version of the causal entropy maximization problem, which allows us to generate a class of maximum entropy IRL models. Our generalized model has an advantage of being able to recover, in addition to a reward function, another expert's function that would (partially) capture the impact of the connecting structure of the states on experts' decisions. Empirical evaluation on a real-world dataset and a grid-world dataset shows that our generalized model outperforms the classical ones, in terms of recovering reward functions and demonstrated trajectories.

Introduction

We are interested in inverse reinforcement learning (IRL) (Russell 1998; Abbeel and Ng 2004; Ziebart et al. 2008), which refers to the problem of learning and imitating experts' behavior by observing their demonstrated trajectories of states and actions in some planning space. The experts are assumed to make actions by optimizing some accumulated rewards associated with states that they visit and the actions they make. The learner then aims at recovering such rewards to understand how decisions are made, and ultimately to imitate experts' behavior. The rationale behind IRL is that although a reward function might be a succinct and generalizable representation of an expert behavior, but it is often difficult for the experts to provide their reward functions, as opposed to giving demonstrations. So far, IRL has been successfully applied in a wide range of problems such as predicting driver route choice behavior (Ziebart et al. 2008) or planning for quadruped robots (Ratliff, Silver, and Bagnell 2009).

Maximum entropy IRL (Ziebart et al. 2008; Ziebart, Bagnell, and Dey 2010) is a powerful probabilistic approach that has received a significant amount of attention over the decade. The main advantages of this IRL framework is that it allows the removal of ambiguity between demonstrations and the expert policy and to cast the reward learning as a maximum likelihood estimation problem. One of the interesting properties of the framework is that the action distribution can be interpreted as a solution to a causal entropy maximization problem under constraints on the empirical expectation of the rewards, which also provides a worst-case prediction log-loss guarantee (Ziebart, Bagnell, and Dey 2010).

In fact, through demonstrations, most of the IRL models will return a reward function associated with states and actions, but give no information about the effect of the connecting structure of the states on expert's decisions. In other words, the way states are connected to others may have a significant impact on expert's policy, but, to the best of our knowledge, this is not captured thoroughly by the classical IRL models. For example, when travelling in a transportation network, an experienced taxi driver may not only consider travelling costs, but also take into consideration the correlation between possible paths. In the next sections, we will provide a simple example to illustrate this issue. This type of issue has been widely investigated by numerous econometrics studies (Train 2009).

Motivated by the above issue of the classical IRL models, we propose a generalized IRL model based on the principle of maximum causal entropy. More precisely, we propose a generalized version of the causal entropy function considered in Ziebart, Bagnell, and Dey (2010) and show that solving the corresponding generalized causal entropy maximization problem will yield a formulation to infer action probabilities for the reward learning problem. Our generalized model is more flexible and robust than the classical ones, in the sense that it allows to recover, in addition to an expert's reward function, a function that may partially capture the correlation between different trajectories. From a theoretical point of view, our generalized model is also consistent with the maximum causal entropy scheme, and also holds a worst-case prediction log-loss guarantee. We provide experiments using a real-world taxi trajectories and a grid-world dataset. Our results show that the generalized model performs better than other classical IRL ones, in terms of recovering expert's reward functions and recovering demonstrated trajectories.

Related work. Our algorithm directly generalize the max-

*mai.tien@smart.mit.edu

imum causal entropy model proposed in Ziebart, Bagnell, and Dey (2010), so it is closely related to IRL methods proposed by Ho and Ermon (2016); Fu, Luo, and Levine (2017); Finn, Levine, and Abbeel (2016); and Levine, Popovic, and Koltun (2011). The generative adversarial imitation learning algorithm proposed by Ho and Ermon (2016) is a powerful approach that allows to learn directly from demonstration without recovering a reward function. Nevertheless, in many scenarios, a reward function returned from IRL might be useful to infer expert's intentions or to avoid re-optimizing a reward function in a new environment. Finn et al. (2016) show a connection between generative adversarial networks (GANs) (Goodfellow et al. 2014), maximum entropy IRL and energy-based models. They also propose the adversarial IRL framework that allows to learn a reward function based on the GANs idea. Fu, Luo, and Levine (2017) develop an algorithm based on this adversarial IRL framework, which provide a way to recover a reward function that is "robust" in different dynamic settings. These GANs-based algorithms all rely on the maximum causal IRL framework (Ziebart, Bagnell, and Dey 2010), thus can be adapted to use with our generalized IRL model. There are also some methods aiming at learning nonlinear reward functions through, e.g., boosting structured prediction (Bagnell et al. 2007), deep neural networks (Wulfmeier, Ondruska, and Posner 2015) or Gaussian processes (Levine, Popovic, and Koltun 2011), which might also benefit from our generalized IRL model.

Background

An IRL model typically relies on a Markov Decision Process (MDP), which consists of states, actions and transition probabilities when making an action at any state. We first consider an MDP for an agent, defined as $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is a set of states $\mathcal{S} = \{1, 2, \dots, |\mathcal{S}|\}$, $\mathcal{A}$ is a finite set of actions, $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition probability function, i.e., $p(s'|a, s)$ is the probability of moving to state $s' \in \mathcal{S}$ from $s \in \mathcal{S}$ by performing action $a \in \mathcal{A}$, $R(a, s|\theta)$ is a reward function of parameters θ and a feature vector $F(a, s)$ associated with making decision $a \in \mathcal{A}$ at state $s \in \mathcal{S}$, and γ is a discount factor.

In this work we consider the case of finite time horizon and undiscounted MDP. We first denote $\mathbf{A}, \mathbf{S}$ as sequences of actions and states: $\mathbf{A} = \{a_0, \dots, a_{T-1}\}$, $\mathbf{S} = \{s_0, \dots, s_{T-1}\}$, where $a_t \in \mathcal{A}, s_t \in \mathcal{S}$ are the action and state at time $t \in \{0, \dots, T-1\}$. The probability of $\mathbf{A}$ causally conditioned on $\mathbf{S}$ is defined as $P(\mathbf{A}||\mathbf{S}) = \prod_{t=0}^{T-1} P(a_t|s_t)$, and the causal entropy of $\mathbf{A}$ conditional on $\mathbf{S}$ is defined as (Kramer 1998; Permuter, Kim, and Weissman 2008)

$$H(\mathbf{A}||\mathbf{S}) = -\mathbb{E}_{\mathbf{S},\mathbf{A}}[\ln P(\mathbf{A}||\mathbf{S})] = -\sum_{\mathbf{A},\mathbf{S}} P(\mathbf{A},\mathbf{S}) \ln P(\mathbf{A}||\mathbf{S})$$
$$= -\sum_{\mathbf{A},\mathbf{S}} \prod_{t=0}^{T-1} P(a_t|s_t) p(s_{t+1}|a_t, s_t) \ln P(\mathbf{A}||\mathbf{S}).$$

Then, we seek an action distribution that maximizes the following maximum causal entropy function under a constraint on the empirical expectation of the reward (Ziebart, Bagnell, and Dey 2010).

$$\begin{aligned} \underset{P(a_t|s_t)}{\text{maximize}} \quad & H(\mathbf{A}||\mathbf{S}) \\ \text{subject to} \quad & \mathbb{E}_{\mathbf{S},\mathbf{A}}[R(\mathbf{S},\mathbf{A})] = \tilde{\mathbb{E}}_{\mathbf{S},\mathbf{A}}[R(\mathbf{S},\mathbf{A})] \\ & \sum_{a_t \in \mathcal{A}^t} P(a_t|s_t) = 1, \ \forall s_t, \end{aligned}$$

where $R(\mathbf{S}, \mathbf{A})$ is the accumulated reward of actions $\mathbf{A}$ and states $\mathbf{S}$, which is a sum of action/state rewards as $R(\mathbf{S}, \mathbf{A}) = \sum_{t=0}^{T-1} R(a_t, s_t)$, and $\mathcal{A}^t$ is the set of possible actions at time t and $\tilde{\mathbb{E}}[R(\mathbf{S}, \mathbf{A})]$ is the empirical expectation of the reward. Ziebart, Bagnell, and Dey (2010) show that solving the above maximum entropy problem will yield a recursive formulation to infer the action distribution, making the training of the corresponding IRL model tractable

$$\begin{gathered} \ln Z_{a_t|s_t} = R(s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} p(s_{t+1}|a_t, s_t) \ln Z_{s_{t+1}}, \\ Z_{s_t} = \sum_{a_t \in \mathcal{A}^t} Z_{a_t|s_t}, \ P(a_t|s_t) = Z_{a_t|s_t}/Z_{s_t}, \qquad (1) \\ t = 0, \dots, T-1, \ a_t \in \mathcal{A}^t, s_t \in \mathcal{S}. \end{gathered}$$

The above formulation provides a way to infer the probability of any demonstrated trajectory and to train the IRL model by maximum likelihood estimation. We note that when the dynamics are deterministic, i.e., the transition probabilities only take values of 0 or 1, then the maximum causal entropy IRL proposed by Ziebart, Bagnell, and Dey (2010) is identical to the maximum entropy IRL model introduced in their previous work (Ziebart et al. 2008), which is also an energy-based model over all possible trajectories (Finn et al. 2016).

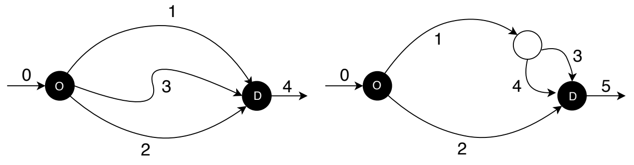
Generalized Maximum Causal Entropy

In this section, we will start by showing a bottleneck of the classical maximum entropy models (Ziebart et al. 2008; Ziebart, Bagnell, and Dey 2010). We then propose our generalized maximum causal entropy (GMCE) IRL based on the maximum causal entropy principle. We will also provide an algorithm that can be used to practically train the GMCE IRL model. Lastly, we will take an example to show how the GMCE gets over the limitation of the classical model.

Bottleneck of the Classical Models

One of the issues of the maximum causal entropy (MCE) IRL model specified by (1) is that it only relies on a reward function associated with states and actions and might not be able to capture the structure of the network of states, which would lead to an unreasonable probability distribution over trajectories. We will take two simple examples (Fig. 1a and 1b) and bring some insights from econometrics to illustrate this issue.

In these examples, we assume that there are three paths going from an initial node (denoted by O) to a terminal node (D). Links are the states of the model and we number them as in the figures. To make the examples simple, we assume that an action is defined as moving from a state to another state, which means that the MDP is deterministic. In the left example, there are three possible paths to go from O to D as $\{0,1,4\}, \{0,3,4\}$ and $\{0,2,4\}$ and in Fig. 1b there are also three paths connecting O and D as $\{0,1,4,5\}, \{0,1,3,5\}$ and $\{0,2,5\}$. We further assume that all the paths have the same rewards, i.e., $R(1) = R(2) = R(3)$ for Ex. 1 and $R(1) + R(3) = R(1) + R(4) = R(2)$ for Ex. 2.



(a) Example 1

(b) Example 2

Figure 1: Simple examples to illustrate a limitation of the classical IRL model capturing the structure of the network of states

Clearly, the MCE IRL model assigns the same probability of 1/3 to each of the three paths in each example. That makes sense for Ex. 1 where there is no overlap between the three paths. However, for Ex. 2, the MCE IRL model will still assign a probability of 1/3 to each of the three paths despite the overlap between two of the paths. More precisely, if the reward of States 3 and 4 are much smaller than the reward of State 1 (but the three paths still have the same rewards), then we expect that the probabilities of Path $\{0,2,5\}$ should approach 1/2 and the properties of Paths $\{0,1,3,5\}$ and $\{0,1,4,5\}$ should be close to 1/4.

Moreover, if we look at the ratio between the probabilities of any two paths, that ratio does not change no matter what the other paths are. For instance, if Paths $\{0,1,3,5\}$ and $\{0,2,5\}$ have the same rewards, then $P(\{0,1,3,5\})/P(\{0,2,5\})$ is always equal to 1, even if we add more states to connect O and D, or remove some possible paths. This property of the MCE IRL also refers to a well-known issue in econometrics, called the Independence from Irrelevant Alternatives (IIA) property (Train 2009), which would result in inaccurate prediction in many applications.

Generalized Maximum Causal Entropy

To deal with the aforementioned issue, we generalize the MCE IRL model. To this end, let us define a generalized causal entropy function of actions $\mathbf{A}$ conditional on a sequence of states $\mathbf{S}$

$$H^g(\mathbf{A}||\mathbf{S}) = \mathbb{E}_{\mathbf{A},\mathbf{S}}\left[-\sum_{t=0}^{T} \ln G(P(a_t|s_t)|s_t)\right], \qquad (2)$$

where $G(p|s) : \mathbb{R}_+ \times \mathcal{S} \to \mathbb{R}_+$. Note in particular that while we will ultimately be interested in the case where $p \in [0,1]$, we extend the definition of G to encompass all $p \in \mathbb{R}_+$. To derive closed-form recursive equations for the action probabilities, we require G to satisfy the following conditions (Assumption 1), for any $p \in \mathbb{R}_+$ and $s \in \mathcal{S}$.

Assumption 1 *Function $G(p|s) : \mathbb{R}_+ \times \mathcal{S} \to \mathbb{R}_+$ satisfies the following conditions*

(i) $G(p|s)$ and $\partial G(p|s)/\partial p$ both exist and are positive

(ii) $G(p|s)$ is invertible, i.e., there exists an unique function $G^{-1}(h|s) : \mathbb{R}_+ \times \mathcal{S} \to \mathbb{R}_+$ such that

$$G^{-1}(G(p|s)|s) = p,\ \forall p \in \mathbb{R}_+$$

(iii) There exists a mapping $\mu : \mathcal{S} \to \mathbb{R}_+$ such that

$$\frac{p\partial \ln(G(p|s))}{\partial p} = \mu(s).$$

Note that this implies there exists a mapping $\nu : \mathcal{S} \to \mathbb{R}$ such that $G(p|s) = e^{\nu(s)}p^{\mu(s)}$. Moreover, the above conditions also imply that function $G(p|\cdot)$ is multiplicative, i.e., $G(p_1p_2|s) = G(p_1|s)G(p_2|s),\ \forall p_1, p_2 \in \mathbb{R}_+$. This property will be useful for deriving a closed-form solution for the generalized maximum causal entropy problem.

Now, we aim at solving the following generalized causal entropy maximization problem under the generalized causal entropy function defined in (2). A solution to this problem will provide a way to infer a probability distribution over actions and states

$$\begin{aligned}
\underset{P(a_t|s_t),\ \forall a_t, s_t}{\text{maximize}} \quad & H^g(\mathbf{A}||\mathbf{S}) && \text{(P2)}\\
\text{subject to} \quad & \mathbb{E}_{\mathbf{S},\mathbf{A}}[R(\mathbf{S},\mathbf{A})] = \tilde{\mathbb{E}}_{\mathbf{S},\mathbf{A}}[R(\mathbf{S},\mathbf{A})] \\
& \sum_{a_t \in \mathcal{A}^t} P(a_t|s_t) = 1 \\
& \forall s_t \in \mathcal{S},\ t = 0, \dots, T-1.
\end{aligned}$$

The following theorem indicates that, under the conditions imposed on function $G(\cdot)$ in Assumption 1, there are closed forms to compute an optimal solution to (P2), making the training of the GMCE IRL model practically tractable.

Theorem 1 *If function $G(p|s)$ satisfies Assumption 1 and $P(a_t|s_t),\ \forall a_t \in \mathcal{A},\ s_t \in \mathcal{S}$ is a optimal solution to the generalized maximum causal entropy problem* (P2)*, then these probabilities can be computed by the following recursive equations*

$$\begin{aligned}
Y_{a_t|s_t} &= \lambda R(s_t, a_t) \\
&\quad + \sum_{s_{t+1} \in \mathcal{S}} p(s_{t+1}|a_t, s_t) \ln G(Z_{s_{t+1}}|s_{t+1}) \\
Z_{a_t|s_t} &= G^{-1}\left(e^{Y_{a_t|s_t}}|s_t\right),\ Z_{s_t} = \sum_{a_t \in \mathcal{A}^t} Z_{a_t|s_t} \\
P(a_t|s_t) &= Z_{a_t|s_t}/Z_{s_t}, \\
&\quad t = 0, \dots, T-1,\ a_t \in \mathcal{A}^t, s_t \in \mathcal{S}.
\end{aligned}$$

Proof. (*sketched*). It suffices to find $P(a_t|s_t)$ that maximize $\mathcal{D} = H^g(\mathbf{A}||\mathbf{S}) + \lambda\mathbb{E}_{\mathbf{S},\mathbf{A}}[R(\mathbf{S},\mathbf{A})]$ for a constant λ. We denote

$$\mathcal{D} = \mathbb{E}_{\mathbf{S},\mathbf{A}}\left[-\sum_{t=0}^{T} \ln G(P(a_t|s_t)|s_t) + \lambda R(\mathbf{S},\mathbf{A})\right].$$

Using the method of Lagrange Multipliers, for each $s_t \in \mathcal{S}$, we require that $\partial\mathcal{D}/\partial P(a_t|s_t)$ are equal over all actions $a_t \in \mathcal{A}^t$. Taking the derivative of $\mathcal{D}$ with respect to $P(a_t|s_t)$, removing parts that are equal over $a_t \in \mathcal{A}^t$ and using Assumption 1, we obtain

$$P(a_t|s_t) \propto G^{-1}\left(\exp\left(\lambda R(s_t,a_t) + U(s_t,a_t)\right)|s_t\right), \quad (3)$$

where

$$U(s_t,a_t) = \sum_{k=t+1}^{T-1} \mathbb{E}_{s_k,a_k}\Big[\ln G(P(a_k|s_k)|s_k) - \lambda R(s_k,a_k)|a_t,s_t\Big].$$

Now, by the multiplicativity of $G(p|\cdot)$ in p, we can reduce $U(s_t,a_t)$ as

$$U(s_t,a_t) = \sum_{s_{t+1}} P(s_{t+1}|a_t,s_t)\ln G(Z_{s_{t+1}}|s_{t+1}). \quad (4)$$

Combining (3) and (4), we obtain the desired results. ■

In fact, if $G(p|s) = p$, then we obtain the recursions in (1) and the GMCE becomes the classical MCE IRL model.

It is also possible to prove a worst-case prediction log-loss guarantee for the solution given in Theorem 1, as follows.

Theorem 2 *A solution to* (P2) *minimizes the following generalized worst-case prediction log-loss*

$$\inf_{Q\in\Delta^T} \sup_{\substack{P\in\Delta^T \\ \tilde{\mathbb{E}}^P(R)=\eta}} \mathbb{E}^P_{\boldsymbol{A},\boldsymbol{S}}\left[-\sum_{t=0}^{T-1}\ln G(Q(a_t|s_t)|s_t)\right], \quad (5)$$

where $\tilde{\mathbb{E}}^P(R) = \mathbb{E}^P_{\boldsymbol{A},\boldsymbol{S}}[R(\boldsymbol{A},\boldsymbol{S})]$ *and* $\eta = \tilde{\mathbb{E}}_{\boldsymbol{S},\boldsymbol{A}}[R(\boldsymbol{S},\boldsymbol{A})]$ *(i.e., empirical expectation reward) and* $\Delta^T = \{P(a_t|s_t), a_t \in \mathcal{A}^t,\ s_t \in \mathcal{S}, \sum_{a_t\in\mathcal{A}^t} P(a_t|s_t) = 1\}$.

Proof. (*sketched*). We can follow the same strategy in Grünwald, Dawid, and others (2004) to prove the result. First, one can show that

$$\inf_{Q\in\Delta^T} \sum_{\mathbf{A},\mathbf{S}} P(\mathbf{A},\mathbf{S})\left[-\sum_{t=0}^{T-1}\ln G(Q(a_t|s_t)|s_t)\right] \quad (6)$$

is only achieved uniquely at $Q = P$. This result can proved by taking the derivative of the Lagrangian function with respect to a variable $Q(a_t|s_t)$. Setting this derivative to zero and using Assumption 1 we can show that if Q is a solution to (6), then $Q(a_t|s_t)$ needs to be equal to $P(a_t|s_t)$ for all a_t, s_t.

Thus, the generalized maximum causal entropy (P2) can be written as

$$\sup_{\substack{P\in\Delta^T \\ \tilde{\mathbb{E}}^P(R)=\eta}} \inf_{Q\in\Delta^T} \sum_{\mathbf{A},\mathbf{S}} P(\mathbf{A},\mathbf{S})\left[-\sum_{t=0}^{T-1}\ln G(Q(a_t|s_t)|s_t)\right]. \quad (7)$$

Now, we see that the above objective function is convex in Q (if we fix P) and concave in P (if we fix Q), so using the *Neumann's minimax theorem*, we can switch the sup-inf order and obtain an equivalent inf-sup problem, which is (5). As we have seen, $Q = P$ is the unique solution to the infimum problem of (7) and P that achieve the supremum of (7) is a solution to the maximum causal entropy problem (P2). This leads to our desired result. ■

Theorem 2 says that the generalized maximum causal entropy can be viewed as a *zero-sum game* where the opponent chooses a distribution over actions/states to maximize the predictor's generalized log-loss value, and the predictor try to choose a distribution to minimize it. This interesting result indeed holds for the classical MCE IRL model and Theorem 2 indicates that this also holds for our generalized model with a generalized log-loss function.

Learning Algorithm

We describe the main steps for computing the Log-likelihood and its gradient in Algorithm 1. The algorithm performs a backward procedure from $t = T$ to $t = 0$. To make the algorithm general, we just assume that reward $R(a_t,s_t)$ is a function of feature vector $\mathcal{F}(a_t,s_t)$ and parameter θ and $\nabla_\theta R(a_t,s_t)$ is the gradient of $R(a_t,s_t)$ with respect to θ. In a typical setting, if the rewards have a linear form, then we can write $R(a_t,s_t) = \theta^{\mathrm{T}}\mathcal{F}(a_t,s_t)$ and $\nabla_\theta R(a_t,s_t) = \mathcal{F}(a_t,s_t)$. Here, we also assume that $G(p|s)$ is a function of p, a feature vector associated with state s and some parameters to be inferred through the training. That is, we can write $G(p|s) = G(p|s,\theta')$, where θ' is a vector of parameters of its own. The gradient $\nabla_\theta G^{-1}(e^{Y_{a_t|s_t}}|s_t)$ in (8) refers to the gradient vector of G with respect to its own parameters θ'. The gradient vector of the the log-likelihood can be straightforwardly derived from the recursive equations in Theorem 1.

As having said, the conditions in Assumption 1 also implies that G has the form $G(p|s) = e^{v(s)}p^{\mu(s)}$. One selection that would be of interest is $G(p|s) = p^{(\theta^\mu)^{\mathrm{T}}\mathcal{F}(s)}$, where θ^μ is a vector of parameters to be inferred and $\mathcal{F}(s)$ is a feature vector associated with state $s \in \mathcal{S}$. We also denote θ^R as the parameter vector for the reward function and in a linear setting, we can write $R(a_t,s_t) = (\theta^R)^{\mathrm{T}}\mathcal{F}(a_t,s_t)$. The inverse of $G(p|s)$ becomes $G^{-1}(p|s) = \exp\left(\ln p/((\theta^\mu)^{\mathrm{T}}\mathcal{F}(s))\right)$. If we substitute this function into Eq. 3, we obtain the following equations to compute "Z" values and the action/state probabilities

$$\begin{aligned}
Y_{a_t|s_t} &= \frac{(\theta^R)^{\mathrm{T}}\mathcal{F}(s_t,a_t)}{(\theta^\mu)^{\mathrm{T}}\mathcal{F}(s_t)} \\
&\quad + \sum_{s_{t+1}\in\mathcal{S}} p(s_{t+1}|a_t,s_t)\frac{(\theta^\mu)^{\mathrm{T}}\mathcal{F}(s_{t+1})}{(\theta^\mu)^{\mathrm{T}}\mathcal{F}(s_t)}\ln Z_{s_{t+1}}, \qquad (9)\\
Z_{s_t} &= \sum_{a_t\in\mathcal{A}^t}\exp\left(Y_{a_t|s_t}\right),\ P(a_t|s_t) = \frac{\exp(Z_{a_t|s_t})}{Z_{s_t}}, \\
&\quad t = T-1,\ldots,0,\ a_t \in \mathcal{A}^t, s_t \in \mathcal{S}.
\end{aligned}$$

Illustrative Example

In the following we show how the GMCE IRL model gets over the aforementioned bottleneck of the classical models. We take the example in Fig. 1b and keep the assumption that all the three paths from State 0 to State 5 have the same rewards. More precisely, we set $R(0) = R(5) = 0$, $R(2) = -5$, $R(1) = -4$ and $R(3) = R(4) = -1$. As being said, under the MCE IRL model, all the three paths have the same probabilities of 1/3. Nevertheless, it seems more reasonable

Algorithm 1 *Log-likelihood and gradient computation*

```
for $t = T, ..., 1$ do
    if $t = T$ then
        $\forall a_t, s_t$, set $Y_{a_t|s_t} = R(a_t, a_t), U_{a_t|s_t} = \nabla_\theta R(a_t, a_t)$ and $Z_{s_t} = 1, D_{s_t} = 0$
    else $\forall a_t, s_t$
        $Y_{a_t|s_t} \leftarrow R(s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} p(s_{t+1}|a_t, s_t) \ln G(Z_{s_{t+1}}|s_{t+1})$
        $E_{s_t, s_{t+1}} \leftarrow \left.\frac{\partial G^{-1}(z|s_t)}{\partial z}\right|_{z=Z_{s_{t+1}}} \frac{D_{s_{t+1}}}{G(Z_{s_{t+1}}|s_{t+1})}$
        $U_{a_t|s_t} \leftarrow \nabla_\theta R(s_t, a_t) + \sum_{s_{t+1} \in \mathcal{S}} p(s_{t+1}|a_t, s_t) E_{s_t, s_{t+1}}$
    end if
    For all $a_t \in \mathcal{A}^t, s_t \in \mathcal{S}$
        $Z_{a_t|s_t} \leftarrow G^{-1}(e^{Y_{a_t|s_t}});\ Z_{s_t} \leftarrow \sum_{a_t} Z_{a_t|s_t} s$
        $D_{a_t|s_t} \leftarrow \left.\frac{\partial G^{-1}(z|s_t)}{\partial z}\right|_{z=e^{Y_{a_t|s_t}}} e^{Y_{a_t|s_t}} U_{a_t|s_t} + \nabla_\theta G^{-1}(e^{Y_{a_t|s_t}}|s_t)$    (8)
        $D_{s_t} \leftarrow \sum_{a_t} D_{a_t|s_t}$
end for
For any observation $(\tilde{s}_t, \tilde{a}_t)$    ▷ LL and its gradients
    $\ln P(\tilde{a}_t|\tilde{s}_t) \leftarrow Z_{\tilde{a}_t|\tilde{s}_t} / Z_{\tilde{s}_t},$
    $\nabla_\theta \ln P(\tilde{a}_t|\tilde{s}_t) \leftarrow \frac{D_{\tilde{a}_t|\tilde{s}_t}}{Z_{\tilde{a}_t|\tilde{s}_t}} - \frac{D_{\tilde{s}_t}}{Z_{\tilde{s}_t}}$
```

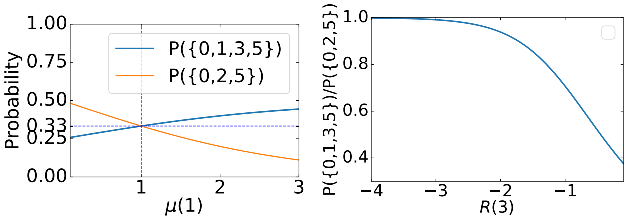


Figure 2: Example of probability distribution over three simple paths. The left figure plots the probabilities of Paths $\{0, 2, 5\}$ and $\{0, 1, 3, 5\}$ when $\mu(1)$ varies. The right figure plots ratio $P(\{0, 1, 4, 5\})/P(\{0, 2, 5\})$ when the reward of State 3 varies.

to have a probability of more than 1/3 for Path $\{0, 2, 5\}$. We now show how this can be achieved by our GMCE model.

We take the GMCE IRL model specified by $G(p|s) = p^{\mu(s)}$. Since the two paths $\{0, 1, 3, 5\}$ and $\{0, 1, 4, 5\}$ all go though State 2 but Path $\{0, 2, 5\}$ does not overlap with any other paths, we vary $\mu(1)$ while keeping $\mu(s) = 1$ for all $s \neq 1$. The probabilities of the three paths with respect to different $\mu(1)$ are plotted in the left sub-figure of Fig. 2, noting that $P(\{0, 1, 3, 5\})$ always equals to $P(\{0, 1, 4, 5\})$. Clearly, $P(\{0, 2, 5\}) > 1/3$ with $\mu(1) < 1$ and $P(\{0, 2, 5\}) > 1/3$ otherwise. When $\mu(1)$ goes 0, the probability of $\{0, 2, 5\}$ approaches 1/2 and the probabilities of $\{0, 1, 3, 5\}$ and $\{0, 1, 4, 5\}$ approach 1/4, which would be more reasonable path probabilities given the network structure.

To illustrate how the GMCE model relaxes the IIA issue mentioned above, we fix $\mu(1) = 0.5$ and set $\mu(s) = 1$ for all $s \neq 1$. The right sub-figure of Fig. 2 plots ratios between $P(\{0, 1, 3, 5\})$ and $P(\{0, 2, 5\})$ with different $R(4)$ (i.e. reward of an irrelevant state). This ratio goes from 1 to 0.4 when we increase the $R(4)$ from -4 (low reward) to 0 (high reward), noting that if $\mu(1) = 1$ (i.e. MCE model), then these ratios are always equal to 1 regardless the reward of State 4. These values are rational from a behavioral point of view, as when $R(4)$ is very low, we expect that Path $\{0, 1, 4, 5\}$ would become unlikely to be chosen, and Paths $\{0, 1, 4, 5\}$ and $\{0, 2, 5\}$ would be similar in terms of attractiveness. On the contrary, if $R(4)$ is very high, then due to the overlap between $\{0, 1, 4, 5\}$ and $\{0, 1, 3, 5\}$, Path $\{0, 1, 4, 5\}$ is very unlikely to be chosen and we could expect that Path $\{0, 1, 3, 5\}$ would be less attractive than Path $\{0, 1, 2, 5\}$, even though they have the same rewards.

In general, the value $\mu(s)$ would affect the attractiveness of all the trajectories passing through that state. So, by recovering such values from demonstrated trajectories, we would expect to better learn how the structure of the network of states affects experts' behavior. Our experiments below also show that the GMCE model performs better than the classical IRL models in recovering expert's trajectories.

Experiments

This section evaluates the GMCE IRL model using two different datasets, namely, a real-world dataset which contains trajectories of taxi drivers (we shall refer to this dataset as the Transport dataset hereafter) and a simulated dataset obtained from a classical grid-world. We use the GMCE model specified by Eq. 9. The model has two vectors of parameters to be inferred through the training, namely, θ^R for the action/state reward function and θ^μ for $\mu(s)$. For the sake of comparison, we will compare our generalized model with the MCE model (Ziebart, Bagnell, and Dey 2010). In the following, we first describe our datasets, and then show the comparison results.

Datasets and Experimental Settings

We will evaluate the models based on log-likelihood and path-matching. We first split each dataset into a training set and a test set. The training set is first used to train the models. Next, for each trajectory in the test set, we feed its first state and last state to the models. Each model will then generate its most likely path based on the first and last states given. Matching is then performed between the particular trajectory from the test set and each of the most likely paths from the models. For each of the most likely paths, we count the number of states which also appeared in the test trajectory. Each count is then divided by the length of the test trajectory in order to give us a percentage of matching. The average matching metric is computed by taking the average of all the percentages of matching computed from every trajectory in the test set for a particular model. The 90% Matching metric is the count of all percentages of matching that are greater or equal to 90%, divided by the total number of percentages of matching.

Transport dataset. The Transport dataset contains a total of 1832 trajectories of taxi drivers. The road network consists of 7288 links, which are regarded as states in our model. At each state, the set of available actions for a taxi driver is to move to one of the connected next states with no uncertainty. This means that the corresponding MDP is deterministic in nature. Four features are used to describe each of the states. Three of the features take binary values representing left-turn, U-turn, and incident-constant. The latter is to count the number of intersections on each route trajectory. The fourth feature is the travel time between each pair of connected links. These features have been used in some established route choice modeling studies (Fosgerau, Frejinger, and Karlstrom 2013; Mai, Fosgerau, and Frejinger 2015). Note that the application can be treated as a finite horizon problem, as it is rational to assume that a driver only considers paths that contain a finite number of links (i.e. states).

We use the aforementioned four features to define the reward function. For the $\mu(s)$, we use the number of incoming links and outgoing links at each state defined as follows. We first organize the trajectories in the Transport dataset into subsets according to which terminal state they end in. The number of incoming links and outgoing links used in the definition of μ function includes only the incoming links and outgoing links that are active in each subset of trajectories. Depending on which subset of trajectories that the model is handling, the number of incoming links and outgoing links for each state will be different. This is done because if we consider the entire dataset, most of the states have similar number of incoming and outgoing links. Thus, without splitting into the different subsets, the information of incoming and outgoing links will not be useful for the model.

Grid-world dataset. The trajectories in the grid-world dataset is generated from a 5x5 grid-world. The agent starts from the bottom-leftmost grid and has to move to the top-rightmost grid, which is also the only terminal state. The actions available in each grid are move left, move right, move up, move down, or stay in the same grid. Unlike the Transport dataset, the grid-world dataset is non-deterministic in nature, i.e., there is a 80% chance that the agent will move in accordance with its intended action, and the remaining 20% probability is distributed evenly to the remaining available actions.

The actual rewards are given in the top-left sub-figure of Fig. 3. Given the actual rewards, we apply Bellman's value iteration (Bellman 1957) to obtain the optimal policy for the grid-world. This optimal policy is then used in the grid-world to generate 200 trajectories. Out of these 200 trajectories, 160 trajectories are used to form the training set and the remaining 40 constitutes the test set. The MCE and the GMCE models are both trained using the training set and then evaluated using the test set. There are 5×5 states and 5×5 features and each feature corresponds to a position on the 5×5 grid, which is a state. For each state, the feature corresponding to it will take a value of 1, while the other features take zero values. The features used for the definition of the reward function are the same as those used for the $\mu(s)$ values, for all $s \in \mathcal{S}$.

Comparison Results

Transport dataset. We place 80% of the taxi trajectories into the training set and the remainder into the test set. To compare the reward functions returned by the MCE model and our GMCE model as well as evaluate the impact of the μ on the performance of our GMCE model, we also take the rewards $R(a_t, s_t)$ from the GMCE after training and set $\mu(s) = 1$ for every state $s \in \mathcal{S}$. By doing so, the GMCE becomes a MCE model but with a different reward function.

	MCE	GMCE	GMCE ($\mu(s) = 1$)
Log Prob. (training)	-2074.3	**-1988.8**	-2226.6
Log Prob. (test)	-566.4	**-523.4**	-613.8
Avg. Matching	87.6%	**89.3**%	88.9%
90% Matching	63.5%	**67.1%**	**67.1%**
Prob. of most likely path	61.0%	63.0%	**71.1%**

Table 1: Comparison of the performance of the different IRL models using the Transport dataset

We report the comparison results in Table 1. The first and second rows clearly show that the GMCE model return significantly larger log-likelihood values for both training and test sets, as compared to the other models. For both measures on the third and fourth rows (Avg. Matching and 90% Matching), we see that the GMCE model outperforms the MCE ones. While the GMCE with $\mu(s) = 1$ performs equivalently to the GMCE in terms of 90% Matching, it performs worse in terms of the average matching of paths. Interestingly, the GMCE with $\mu(s) = 1$ provides better 90% Matching values then the MCE model. On the last row of Table 1, we provide the average probabilities of the most likely paths produced by each of the models. In other words, these values indicate how likely is the model going to produce the most likely path given a pair of origin and destination. The results show that, on average, the GMCE models tend to assign higher probabilities to their most likely trajectories, as compared to the classical one.

Grid-world dataset. The top-right sub-figure of Fig. 3 shows the rewards recovered by the MCE model. The top-rightmost grid is correctly assigned a zero reward. Moreover, except for grid (4,3), the other grids are assigned somewhat significant negative rewards, which is consistent with the actual rewards of the grid-world. However, there are two main issues occurring with these recovered rewards. First, in the actual rewards of the grid-world, all grids except the top-rightmost one have the same reward value of -10. But this not captured by the recovered rewards of the MCE model. Second, a few grids, notably grid (4,3), are assigned reward values that are similar to the top-rightmost grid, which makes MCE's rewards are quite different from the actual one.

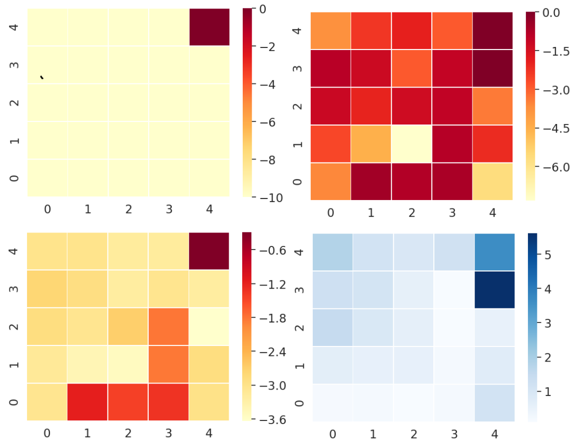


Figure 3: Actual rewards and rewards recovered by the MCE and GMCE models. From left to right, top to bottom: Actual rewards, rewards by MCE, rewards and $\mu(s)$ values returned by the GMCE.

The bottom-left sub-figure of Fig. 3 illustrates the rewards recovered by our GMCE model. Except for 5 grids near the bottom of the grid-world, all the other grids are assigned significantly negative rewards. These rewards are clearly better as compared to those from the MCE in two aspects. First, the GMCE model recovers a greater number of significantly negative grids. Second, the rewards assigned to all the grids excluding the top-rightmost grid are mostly uniform and quite similar in values.

For the sake of illustration, in the bottom-right sub-figure of Fig. 3, we provide a heatmap of the values of μ returned by the GMCE model. Clearly, these μ values are not uniformly distributed over the grids. Instead, there are two grids at the top-rightmost being assigned significantly high values while the other grids takes quite similar and low values. This distribution of μ values, interestingly, is quite similar to the distribution of the actual rewards.

We now move to other comparing measures evaluating the ability of the models to recover demonstrated trajectories. The comparison results are reported in Table 2. The first two rows show the log-likelihood values attained by the three models (MCE, GMCE and GMCE with $\mu(s) = 1$), which clearly indicate that the GMCE model returns significantly larger log-likelihood values as compared to the classical MCE, on both training and test sets. On the third and fourth rows, we see that the MCE and GMCE perform equivalently in terms of Avg. Matching and 90% Matching. The reason is that both models have the same most likely path. This most likely path performs noticeably well as it matches most of the trajectories in the test set. However, the last row shows that the GMCE has a much higher chance of producing this most likely path as compared to the MCE model. In other words, the GMCE is more likely to produce a path that models the trajectories in the test set. In this sense, the GMCE outperforms the MCE model.

	MCE	GMCE	GMCE ($\mu(s) = 1$)
Log Prob. (training)	-1984.2	**-879.8**	-3538.5
Log Prob. (test)	-533.1	**-241.9**	-936.0
Avg. Matching	**87.8%**	**87.8%**	78.1%
90% Matching	**67.5%**	**67.5%**	0.0%
Prob. of most likely path	0.0%	**8.0%**	0.0%

Table 2: Comparison of different IRL models using the grid-world dataset

Discussions

Our IRL model and algorithm are general and can be applied to many maximum causal IRL related methods and applications, e.g., adversarial IRL (Finn et al. 2016), robust IRL (Fu, Luo, and Levine 2017), and IRL with nonlinear reward functions (Wulfmeier, Ondruska, and Posner 2015; Levine, Popovic, and Koltun 2011; Bagnell et al. 2007). Finn et al. (2016) show that if the dynamics are deterministic, the MCE becomes an energy-based model. This is however not the case for our GMCE model. This suggests an interesting question that whether there is a generalized version of the energy-based model that can represent our GMCE.

Our training algorithm for the GMCE shares the same structure with the classical maximum entropy IRL, which requires to estimate "Z" values. The estimation of such values might be expensive for large or continuous domains. One could consider a speed-up technique, e.g., parallel computing, to improve the training algorithm. Future research may look into an approximation method such as the guided cost learning (Finn, Levine, and Abbeel 2016), noting that the "Z" values in our generalized model has a more complex structure than those from the classical maximum entropy IRL, leading to the fact that the application of the guided cost learning is not straightforward.

Our idea on generalizing causal entropy functions would be a potential direction to develop new and more general IRL or imitation learning algorithms. For example, one could consider a generalized causal entropy function to develop a generalized version of the generative adversarial imitation learning algorithm proposed by Ho and Ermon (2016). Such generalized model would give a richer expert's policy function and may lead to better imitation results.

Conclusion

In this work, we developed a generalized IRL model that is consistent with the principle of the maximum causal entropy framework and holds a worst-case prediction log-loss guarantee. Our generalized model and algorithm have an advantage of being able to recover an additional expert's function that may capture the impact of the structure of the network on expert's policies. Our experiments clearly indicated the advantage of our generalized approach as compared to the classical ones. Many IRL models and applications would potentially benefit from our approach. In future work, we plan to develop generalized algorithms for IRL and imitation learning in the contexts of unknown or uncertain dynamics.

References

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

[2007] Bagnell, J.; Chestnutt, J.; Bradley, D. M.; and Ratliff, N. D. 2007. Boosting structured prediction for imitation learning. In *Advances in Neural Information Processing Systems*, 1153–1160.

[1957] Bellman, R. 1957. *Dynamic Programming*. Princeton University Press.

[2016] Finn, C.; Christiano, P.; Abbeel, P.; and Levine, S. 2016. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*.

[2016] Finn, C.; Levine, S.; and Abbeel, P. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, 49–58.

[2013] Fosgerau, M.; Frejinger, E.; and Karlstrom, A. 2013. A link based network route choice model with unrestricted choice set. *Transportation Research Part B: Methodological* 56:70–80.

[2017] Fu, J.; Luo, K.; and Levine, S. 2017. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*.

[2014] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

[2004] Grünwald, P. D.; Dawid, A. P.; et al. 2004. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *the Annals of Statistics* 32(4):1367–1433.

[2016] Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems*, 4565–4573.

[1998] Kramer, G. 1998. *Directed information for channels with feedback*. Hartung-Gorre.

[2011] Levine, S.; Popovic, Z.; and Koltun, V. 2011. Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in Neural Information Processing Systems*, 19–27.

[2015] Mai, T.; Fosgerau, M.; and Frejinger, E. 2015. A nested recursive logit model for route choice analysis. *Transportation Research Part B: Methodological* 75:100–112.

[2008] Permuter, H. H.; Kim, Y.-H.; and Weissman, T. 2008. On directed information and gambling. In *2008 IEEE International Symposium on Information Theory*, 1403–1407. IEEE.

[2009] Ratliff, N. D.; Silver, D.; and Bagnell, J. A. 2009. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots* 27(1):25–53.

[1998] Russell, S. J. 1998. Learning agents for uncertain environments. In *COLT*, volume 98, 101–103.

[2009] Train, K. E. 2009. *Discrete choice methods with simulation*. Cambridge university press.

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

[2010] Ziebart, B. D.; Bagnell, J. A.; and Dey, A. K. 2010. Modeling interaction via the principle of maximum causal entropy. In *Proceedings of the Twenty-seventh International Conference on Machine Learning (ICML'10)*, 1255–1262.

[2008] Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of The Twenty-third AAAI Conference on Artificial Intelligence (AAAI'08)*, volume 8, 1433–1438. Chicago, IL, USA.