

Deep Adaptive Multi-Intention Inverse Reinforcement Learning

Ariyan Bighashdel^(✉), Panagiotis Meletis, Pavol Jancura, and Gijs Dubbelman

Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands
{a.bighashdel, p.c.meletis, p.jancura, g.dubbelman}@tue.nl

Abstract. This paper presents a deep Inverse Reinforcement Learning (IRL) framework that can learn an *a priori* unknown number of nonlinear reward functions from unlabeled experts' demonstrations. For this purpose, we employ the tools from Dirichlet processes and propose an adaptive approach to simultaneously account for both complex and unknown number of reward functions. Using the conditional maximum entropy principle, we model the experts' multi-intention behaviors as a mixture of latent intention distributions and derive two algorithms to estimate the parameters of the deep reward network along with the number of experts' intentions from unlabeled demonstrations. The proposed algorithms are evaluated on three benchmarks, two of which have been specifically extended in this study for multi-intention IRL, and compared with well-known baselines. We demonstrate through several experiments the advantages of our algorithms over the existing approaches and the benefits of online inferring, rather than fixing beforehand, the number of expert's intentions.

Keywords: Inverse reinforcement learning · Multiple intentions · Deep learning.

1 Introduction

The task of learning from demonstrations (LfD) lies in the heart of many artificial intelligence applications [28, 37]. By observing the expert's behavior, an agent learns a mapping between world states and actions. This so-called *policy* enables the agent to select and perform an action, given the current world state. Despite the fact that this policy can be directly learned from expert's behaviors, inferring the *reward function* underlying the policy is generally considered the most succinct, robust, and transferable methodology for the LfD task [1]. Inferring the reward function, which is the objective of Inverse Reinforcement Learning (IRL), is often very challenging in real-world scenarios. The demonstrations come from multiple experts who can have different intentions, and their behaviors are consequently not well modeled with a single reward function. Therefore, in this study, we research and extend the concept of *mixture of conditional maximum entropy models* and propose a deep IRL framework to infer an *a priori* unknown number of reward functions from experts' demonstrations without intention labels.

Standard IRL can be described as the problem of extracting a reward function, which is consistent with the observed behaviors [33]. Obtaining the exact reward function is an ill-posed problem, since many different reward functions can explain the same observed behaviors [25, 39]. Ziebart et al. [39] tackled this ambiguity by employing the principle of maximum entropy [15]. The principle states that the probability distribution, which best represents the current state of knowledge, is the one with the largest entropy [15]. Therefore, Ziebart et al. [39] chose the distribution with maximal information entropy to model the experts' behaviors. The maximum entropy IRL has been widely employed in various applications [16, 34]. However, this method suffers from a strong assumption that the experts have one single intention in all demonstrations. In this study, we explore the principle of the mixture of maximum entropy models [30] that inherits the advantages of maximum entropy principle, while at the same time is capable of modeling multi-intention behaviors.

In many real-world applications, the demonstrations are often collected from multiple experts whose intentions are potentially different from each other [2, 3, 5, 9]. This leads to multiple reward functions, which is in direct contradiction with the single reward assumption in traditional IRL. To address this problem, Babes et al. [5] proposed a clustering-IRL scheme where the class of each demonstration is jointly learned via the respective reward function. Despite the recovery of multiple reward functions, the number of clusters in this method is assumed to be known *a priori*. To overcome this assumption, Choi et al. [9] presented a non-parametric Bayesian approach using the Dirichlet Process Mixture (DPM) to infer an unknown number of reward functions from unlabeled demonstrations. However, the proposed method is formulated based on the assumption that the reward functions are formed by a linear combination of a set of world state features. In our work, we do not make this assumption on linearity and model the reward functions using deep neural networks.

DPM is a stochastic process in the Bayesian non-parametric framework that deals with mixture models with a countably infinite number of mixture components [24]. In general, full Bayesian inference in DPM models is not feasible, and instead, approximate methods like Monte-Carlo Markov chain (MCMC) [4, 19] and variational inference [7] are employed. When deep neural networks are involved in DPM (e.g. deep nonlinear reward functions in IRL), approximate methods may not be able to scale with high dimensional parameter spaces. MCMC sampling methods are shown to be slow in convergence [7, 29] and variational inference algorithms suffer from restrictions in the distribution family of the observable data, as well as various truncation assumptions for the variational distribution to yield a finite dimensional representation [11, 23]. These limitations apparently make approximate Bayesian inference methods inapplicable for DPM models with deep neural networks. Apart from that, the algorithms for maximum likelihood estimations like standard EM are no longer tractable when dealing with DPM models. The main reason is that the number of mixture components exponentially grows with non-zero probabilities, and after some iterations, the Expectation-step would be no longer available in a closed-form.

However, inspired by two variants of EM algorithms that cope with infeasible Expectation-step [8,36], we propose two solutions in which the Expectation-step is either estimated numerically with sampling (based on Monte Carlo EM [36]) or computed analytically and then replaced with a sample from it (based on stochastic EM [8]).

This study’s main contribution is to develop an IRL framework where one can benefit from the strength of 1) maximum entropy principle, 2) deep nonlinear reward functions, and 3) account for an unknown number of experts’ intentions. To the best of our knowledge, we are the first to present an approach that can combine all these three capabilities.

In our proposed framework, the experts’ behavioral distribution is modeled as a mixture of conditional maximum entropy models. The reward functions are parameterized as a deep reward network, consisting of two parts: 1) a base reward model, and 2) an adaptively growing set of intention-specific reward models. The base reward model takes as input the state features and outputs a set of reward features shared in all intention-specific reward models. The intention-specific reward models take the reward features and output the rewards for the respective expert’s intention. A novel adaptive approach, based on the concept of the Chinese Restaurant Process (CRP), is proposed to infer the number of experts’ intentions from unlabeled demonstrations. To train the framework, we propose and compare two novel EM algorithms. One is based on stochastic EM and the other on Monte Carlo EM. In Section 3, this problem of multi-intention IRL is defined, following our two novel EM algorithms in Section 4. The results are evaluated on three available simulated benchmarks, two of which are extended in this paper for multi-intention IRL, and compared with two baselines [5,9]. These experimental results are reported in Section 5 and Section 6 is devoted to conclusions. The source code to reproduce the experiments is publicly available¹.

2 RELATED WORKS

In the past decades, a number of studies have addressed the problem of multi-intention IRL. A comparison of various methods for multi-intention IRL, together with our approach, is depicted in Table 1.

In an early work, Dimitrakakis and Rothkopf [10] formulated the problem of learning from unlabeled demonstrations as a multi-task learning problem. By generalizing the Bayesian IRL approach of Ramachandran and Amir [32], they assumed that each observed trajectory is responsible for one specific reward function, all of which shares a common prior. The same approach has also been employed by Noothigattu et al. [27], who assumed that each expert’s reward function is a random permutation of one sharing reward function. Babes et al. [5] took a different approach and addressed the problem as a clustering task with IRL. They proposed an EM approach that clusters the observed trajectories by

¹ <https://github.com/tue-mps/damiirl>

inferring the rewards function for each cluster. Using maximum likelihood, they estimated the reward parameters for each cluster.

The main limitation in EM clustering approach is that the number of clusters has to be specified as an input parameter [5, 26]. To overcome this assumption, Choi and Kim [9] employed a non-parametric Bayesian approach via the DPM model. Using MCMC sampler, they were able to infer an unknown number of reward functions, which are linear combinations of state features. Other authors have also employed the same methodology in the literature [2, 22, 31].

All above methods are developed on the basis of model-based reinforcement learning (RL), in which the model of the environment is assumed to be known. In the past few years, a couple of approximate, model-free methods have been developed for IRL with multiple reward functions [13, 14, 20, 21]. Such methods aimed to solve large-scale problems by approximating the Bellman optimality equation with model-free RL.

In this study, we constrain ourselves to model-based RL and propose a multi-intention IRL approach to infer an unknown number of experts' intentions and corresponding nonlinear reward functions from unlabeled demonstrations.

| Models | Type | | Features | | |
|--------------------------------|-------------|------------|--------------------------|----------------------|------------------------|
| | Model based | Model free | Unlabeled demonstrations | Unknown # intentions | Non-linear reward fun. |
| Dimitrakakis and Rothkopf [10] | ✓ | | ✓ | | |
| Babes et al. [5] | ✓ | | ✓ | | |
| Nguyen et al. [26] | ✓ | | ✓ | | |
| Choi and Kim [9] | ✓ | | ✓ | ✓ | |
| Rajasekaran et al. [31] | ✓ | | ✓ | ✓ | |
| Li et al. [20] | | ✓ | ✓ | | ✓ |
| Hausman et al. [13] | | ✓ | ✓ | | ✓ |
| Lin and Zhang [21] | | ✓ | | | ✓ |
| Hsiao et al. [14] | | ✓ | ✓ | | ✓ |
| Ours | ✓ | | ✓ | ✓ | ✓ |

Table 1. Comparison of proposed models for multi-intention IRL.

3 PROBLEM DEFINITION

In this section, the problem of multi-intention IRL is defined. To facilitate the flow, we first formalize the multi-intention RL problem. For both problems, we follow the conventional modelling of the environment as a Markov Decision Process (MDP). A finite state MDP in a multi-intention RL problem is a tuple $(S, A, T, \gamma, b_0, R_1, R_2, \dots, R_K)$ where S is the state space, A is the action space, $T : S \times A \times S \rightarrow [0, 1]$ is the transition probability function, $\gamma \in [0, 1]$ is the discount factor, $b_0(s)$ is the probability of starting in state s , and $R_k : S \rightarrow \mathbb{R}$ is the k^{th} reward function with K to be the total number of intentions. A policy is

a mapping function $\pi_k : S \rightarrow A \ \forall k \in \{1, 2, \dots, K\}$. The value of policy π_k with respect to the k^{th} reward function is the expected discounted reward for following the policy and is defined as $V_{R_k}^\pi = \mathbb{E}[\sum_t \gamma^t R_k(s_t) | b_0]$. The optimal policy (π_k^*) for the k^{th} reward function is the policy that maximizes the value function for all states and satisfies the respective Bellman optimality equation [35].

In multi-intention IRL, the context of this study, a finite-state MDP\(\mathbb{R}\) is a tuple $(S, A, T, \gamma, b_0, \boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \dots, \boldsymbol{\tau}^M)$ where $\boldsymbol{\tau}^m$ is the m^{th} demonstration and M is the total number of demonstrations. In this work, it is assumed that there is a total of K intentions, each of which corresponds to one reward function, so that $\boldsymbol{\tau}^m$ with length T_τ is generated from the optimal policy (π_k^*) of the k^{th} reward function. It is further assumed that the demonstrations are without intention labels, i.e. they are *unlabeled*. Therefore, the goal is to infer the number of intentions K and the respective reward function of each intention. In the next section, we model the experts' behaviors as a mixture of conditional maximum entropy models, parameterize the reward functions via deep neural networks, and propose a novel approach to infer an unknown number of experts' intentions from unlabeled demonstrations.

4 APPROACH

In the proposed framework for multi-intention IRL, the experts' behavioral distribution is modeled as a mixture of conditional maximum entropy models. The Mixture of conditional maximum entropy models is a generalization of standard maximum entropy formulation for cases where the data distributions arise from a mixture of simpler underlying latent distributions [30]. According to this principal, a mixture of conditional maximum entropy models is a promising candidate to justify the multi-intention behaviors of the experts. The experts' behaviors with the k^{th} intention is defined via a conditional maximum entropy distribution:

$$p(\boldsymbol{\tau} | \eta_k = 1, \Psi) = \exp(R_k(\boldsymbol{\tau}, \Psi_k)) / Z_k, \quad (1)$$

where $\boldsymbol{\eta} = \{\eta_1, \eta_2, \dots, \eta_K | \forall \eta_k \in \{0, 1\}, \sum_{k=1}^K \eta_k = 1\}$ is the latent intention vector, $R_k(\boldsymbol{\tau}, \Psi_k) = \sum_{s \in \boldsymbol{\tau}} R_k(s, \Psi_k)$ is the reward of the trajectory with respect to the k^{th} reward function with $R_k(s, \Psi_k)$ as the state reward value, and Z_k is the k^{th} partition function.

We define the k^{th} reward function as: $R_k(s, \Psi_k) = R_{\Psi_k}(\mathbf{f}_s)$, where R_{Ψ_k} is a deep neural network with finite set of parameters $\Psi_k = \{\Theta_0, \Theta_k\}$ which consists of a base reward model R_{Θ_0} and an intention-specific reward model R_{Θ_k} (See Fig. 1). The base reward model with finite set of parameters Θ_0 takes the state feature vector \mathbf{f}_s and outputs the state reward feature vector \mathbf{r}_s : $\mathbf{r}_s = R_{\Theta_0}(\mathbf{f}_s)$. The state reward feature vector \mathbf{r}_s that is produced by the base reward model is input to all intention-specific reward models. The k^{th} intention-specific reward model with finite set of parameters Θ_k , takes the state reward feature vector \mathbf{r}_s and outputs the state reward value: $R_k(s, \Psi_k) = R_{\Theta_k}(\mathbf{r}_s)$. Therefore the total set of reward parameters is $\Psi = \{\Theta_0, \Theta_1, \dots, \Theta_K\}$. The reward of the

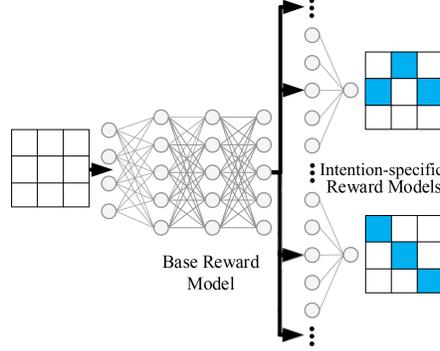


Fig. 1. Schematics of deep reward network.

trajectory τ with respect to the k^{th} reward function can be further obtained as: $R_k(\tau, \Psi_k) = \boldsymbol{\mu}(\tau)^\top \mathbf{R}_{\Psi_k}(\tau)$, where $\boldsymbol{\mu}(\tau)$ is the expected State Visitation Frequency (SVF) vector for trajectory τ and $\mathbf{R}_{\Psi_k}(\tau) = \{R_{\Psi_k}(\mathbf{f}_s) | \forall s \in S\}$ is the vector of reward values of all states with respect to the k^{th} reward function.

In order to infer the number of intentions K , we propose an adaptive approach in which the number of intentions adaptively changes whenever a trajectory is visited/re-visited. For this purpose, at each iteration we first assume to have $M - 1$ demonstrated trajectories $\{\tau^1, \tau^2, \dots, \tau^{m-1}, \tau^{m+1}, \dots, \tau^M\}$ that are already assigned to K intentions with known latent intention vectors $\mathbf{H}^{-m} = \{\boldsymbol{\eta}^1, \boldsymbol{\eta}^2, \dots, \boldsymbol{\eta}^{m-1}, \boldsymbol{\eta}^{m+1}, \dots, \boldsymbol{\eta}^M\}$. Then, we visit/re-visit a demonstrated trajectory τ^m and the task is to obtain the latent intention vector $\boldsymbol{\eta}^m$, which can be assigned to a new intention $K + 1$, and update the reward parameters Ψ . As emphasized before, our work aims to develop a method in which K , the number of intentions, is a priori unknown and can, in theory, be arbitrarily large. Now we define the predictive distribution for the trajectory τ^m as a mixture of conditional maximum entropy models:

$$p(\tau^m | \mathbf{H}^{-m}, \Psi) = \sum_{k=1}^{K+1} p(\tau^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m}) \quad (2)$$

where $p(\eta_k^m = 1 | \mathbf{H}^{-m})$ is the prior intention assignment for trajectory τ^m , given all other latent intention vectors. In the case of K intentions, we define a multinomial prior distribution over all latent intention vectors $\mathbf{H} = \{\mathbf{H}^{-m}, \boldsymbol{\eta}^m\}$:

$$p(\mathbf{H} | \boldsymbol{\phi}) = \prod_{k=1}^K \phi_k^{M_k} \quad (3)$$

where M_k is the number of trajectories with intention k and $\boldsymbol{\phi}$ is the vector of mixing coefficients $\boldsymbol{\phi} = \{\phi_1, \phi_2, \dots, \phi_K\}$ with Dirichlet prior distribution $p(\boldsymbol{\phi}) = \text{Dir}(\alpha/K)$, where α is the concentration parameter. As $K \rightarrow \infty$ the main problematic parameters are the mixing coefficients. Marginalizing out the

mixing coefficients and separating the latent intention vector for m^{th} trajectory yield (see Appendix A for full derivation):

$$\begin{aligned} p(\eta_k^m = 1 | \mathbf{H}^{-m}) &= \frac{M_k^{-m}}{M - 1 + \alpha} \\ p(\eta_{K+1}^m = 1 | \mathbf{H}^{-m}) &= \frac{\alpha}{M - 1 + \alpha} \end{aligned} \quad (4)$$

where M_k^{-m} is the number of trajectories assigned to intention k excluding the m^{th} trajectory, $p(\eta_k^m = 1 | \mathbf{H}^{-m})$ is the prior probability of assigning the new trajectory m to intention $k \in \{1, 2, \dots, K\}$, and $p(\eta_{K+1}^m = 1 | \mathbf{H}^{-m})$ is the prior probability of assigning the new trajectory m to intention $K + 1$. Equation (4) is known as the CRP representation for DPM [24]. Considering the exchangeability property [12], the following optimization problem is defined:

$$\max_{\Psi} L^m(\Psi) = \log \sum_{k=1}^{K+1} p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m}) \quad \forall m \in \{1, 2, \dots, M\} \quad (5)$$

The parameters Ψ can be estimated via Expectation Maximization (EM) [6]. Differentiating $L^m(\Psi)$ with respect to $\psi \in \Psi$ yields the following E-step and M-step (see Appendix B for full derivation):

E-step Evaluation of the posterior distribution over the latent intention vector $\forall k \in \{1, 2, \dots, K\}$:

$$\gamma_k^m = \frac{M_k^{-m} \prod_{t=0}^{T_\tau-1} \pi_k(a_t | s_t)}{\alpha \prod_{t=0}^{T_\tau-1} \pi_{K+1}(a_t | s_t) + \sum_{\hat{k}=1}^K M_{\hat{k}}^{-m} \prod_{t=0}^{T_\tau-1} \pi_{\hat{k}}(a_t | s_t)} \quad (6)$$

and for $k = K + 1$:

$$\gamma_k^m = \frac{\alpha \prod_{t=0}^{T_\tau-1} \pi_k(a_t | s_t)}{\alpha \prod_{t=0}^{T_\tau-1} \pi_{K+1}(a_t | s_t) + \sum_{\hat{k}=1}^K M_{\hat{k}}^{-m} \prod_{t=0}^{T_\tau-1} \pi_{\hat{k}}(a_t | s_t)} \quad (7)$$

where we have defined $\gamma_k^m = p(\eta_k^m = 1 | \boldsymbol{\tau}^m, \mathbf{H}^{-m}, \Psi)$.

M-step update of the parameter value $\psi \in \Psi$ with gradient of:

$$\nabla_{\psi} L(\Psi) = \sum_{k=1}^{K+1} \gamma_k^m (\boldsymbol{\mu}(\boldsymbol{\tau}^m) - \mathbb{E}_{p(\boldsymbol{\tau} | \eta_k = 1, \Psi)}[\boldsymbol{\mu}(\boldsymbol{\tau})])^\top \frac{d\mathbf{R}_{\Psi_k}(\boldsymbol{\tau})}{d\psi} \quad (8)$$

where $\mathbb{E}_{p(\boldsymbol{\tau} | \eta_k = 1, \Psi)}[\boldsymbol{\mu}(\boldsymbol{\tau})]$ is the expected SVF vector under the parameterized reward function R_{Ψ_k} [39].

When K approaches infinity, the EM algorithm is no longer tractable since the number of mixture components exponentially grows with non-zero probabilities. As a result, after some iterations, the E-step would be no longer available in a closed-form. We propose two solutions for estimation of the reward parameters which are inspired by stochastic and Monte Carlo EM algorithms. Both proposed solutions are deeply evaluated and compared with in Section 5.

Algorithm 1: Adaptive multi-intention IRL based on stochastic EM

```

Initialize  $K, \Theta_0, \Theta_1, \Theta_2, \dots, \Theta_K, M_1, M_2, \dots, M_K$ ;
while iteration < MaxIter do
  Solve for  $\pi_1, \pi_2, \dots, \pi_K$ ;
  for  $m = 1$  to  $M$  do
    Initialize  $\Theta_{K+1}$  and solve for  $\pi_{K+1}$ ;
    E-step Obtain  $\gamma_k^m \forall k \in \{1, 2, \dots, K, K+1\}$ ;
    S-step Sample  $\eta_k^m \sim \gamma_k^m$ ;
    if  $\eta_{K+1}^m = 1$  then
      |  $K = K + 1$ ;
    end
    Remove  $K_u$  unoccupied intentions:  $K = K - K_u$ ;
    Update  $M_1, M_2, \dots, M_K$ ;
    M-step Update  $\psi \in \{\Theta_0, \Theta_1, \Theta_2, \dots, \Theta_K\}$  by (8);
  end
end

```

4.1 First solution with stochastic expectation maximization

Stochastic EM, introduces a stochastic step (S-step) after the E-step that represents the full expectation with a single sample [8]. Alg. 1 presents the summary of the first solution to multi-intention IRL via stochastic EM algorithm when the number of intentions is no longer known.

Given (6) and (7), first the posterior distribution over the latent intention vector $\boldsymbol{\eta}^m$ for trajectory $\boldsymbol{\tau}^m \in \{\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \dots, \boldsymbol{\tau}^M\}$ is obtained. Then, the full expectation is estimated with a sample $\boldsymbol{\eta}^m$ from the posterior distribution. Finally, the reward parameters are updated via (8).

4.2 Second solution with Monte Carlo expectation maximization

The Monte Carlo EM algorithm is a modification of the EM algorithm where the expectation in the E-step is computed numerically via Monte Carlo simulations [36]. As indicated, Alg. 1 relies on the full posterior distribution which can be time-consuming. Therefore, another solution for multi-intention IRL is presented in which the E-step is performed through Metropolis-Hastings sampler (see Alg. 2 for the summary).

First, a new intention assignment for m^{th} trajectory, $\boldsymbol{\eta}^{*m}$, is sampled from the prior distribution of (4), then $\boldsymbol{\eta}^m = \boldsymbol{\eta}^{*m}$ is set with the acceptance probability of $\min\{1, \frac{p(\boldsymbol{\tau}^m | \boldsymbol{\eta}^{*m}, \Psi)}{p(\boldsymbol{\tau}^m | \boldsymbol{\eta}^m, \Psi)}\}$ where (see Appendix C for full derivation):

$$\frac{p(\boldsymbol{\tau}^m | \eta_{k^*}^{*m} = 1, \Psi)}{p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi)} = \frac{\prod_{t=1}^{T_\tau} \pi_{k^*}(a_t^m | s_t^m)}{\prod_{t=1}^{T_\tau} \pi_k(a_t^m | s_t^m)} \quad (9)$$

with $k \in \{1, 2, \dots, K\}$ and $k^* \in \{1, 2, \dots, K, K+1\}$.

Algorithm 2: Adaptive multi-intention IRL based on Monte Carlo EM

```

Initialize  $K, \Theta_0, \Theta_1, \Theta_2, \dots, \Theta_K, M_1, M_2, \dots, M_K$ ;
while  $iteration < MaxIter$  do
    Solve for  $\pi_1, \pi_2, \dots, \pi_K$ ;
    for  $m = 1$  to  $M$  do
        Obtain  $p(\boldsymbol{\eta}^m | \mathbf{H}^{-m}, \alpha)$ ;
        Sample  $\boldsymbol{\eta}^{*m} \sim p(\boldsymbol{\eta}^m | \mathbf{H}^{-m}, \alpha)$ ;
        if  $\eta_{K+1}^{*m} = 1$  then
            Initialize  $\Theta_{K+1}$  and solve for  $\pi_{K+1}$ ;
        end
        E-step Assign  $\boldsymbol{\eta}^{*m} \rightarrow \boldsymbol{\eta}^m$  by probability of  $\min\{1, \frac{p(\boldsymbol{\tau}^m | \boldsymbol{\eta}^{*m}, \Psi)}{p(\boldsymbol{\tau}^m | \boldsymbol{\eta}^m, \Psi)}\}$ ;
        if  $\eta_{K+1}^m = 1$  then
             $K = K + 1$ ;
        end
        Remove  $K_u$  unoccupied intentions:  $K = K - K_u$ ;
        Update  $M_1, M_2, \dots, M_K$ ;
        M-step Update  $\psi \in \{\Theta_0, \Theta_1, \Theta_2, \dots, \Theta_K\}$  by (8);
    end
end

```

5 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our proposed methods through several experiments with three goals: 1) to show the advantages of our methods in comparison with the baselines in environments with both linear and non-linear rewards, 2) to demonstrate the advantages of adaptively inferring the number of intentions rather than predefining a fixed number, and 3) to depict the strengths and weaknesses of our proposed algorithms with respect to each other.

5.1 Benchmarks

In order to deeply compare the performances of various models, the experiments are conducted on three different environments: GridWorld, Multi-intention ObjectWorld, and Multi-intention BinaryWorld. Variants of all three environments have been widely employed in IRL literature [18, 38].

GridWorld [9] is a 8×8 environment with 64 states and four actions per state with 20% probability of moving randomly. The grids are partitioned into non-overlapping regions of size 2×2 , and the feature function is defined by a binary indicator function for each region. Three reward functions are generated with linear combinations of state features and reward weights which are sampled to have a non-zero value with the probability of 0.2. The main idea behind using this environment is to compare all the models in aspects other than their capability of handling linear/non-linear reward functions.

Multi-intention ObjectWorld (M-ObjectWorld) is our extension of ObjectWorld [18] for multi-intention IRL. ObjectWorld is a 32×32 grid of states with

five actions per state with a 30% chance of moving in a different random direction. The objects with two different inner and outer colors are randomly placed, and the binary state features are obtained based on the Euclidean distance to the nearest object with a specific inner or outer color. Unlike ObjectWorld, M-ObjectWorld has six different reward functions, each of which corresponds to one intention. The intentions are defined for each cell based on three rules: 1) within 3 cells of outer color one and within 2 cells of outer color two, 2) Just within 3 cells of outer color one, and 3) everywhere else (see Table 2). Due to the large number of irrelevant features and the nonlinearity of the reward rules, the environment is challenging for methods that learn linear reward functions. Fig. 2 (top three) shows a 8×8 zoom-in of M-ObjectWorld with three reward functions and respective optimal policies.

Multi-intention BinaryWorld (M-BinaryWorld) is our extension of BinaryWorld [38] for multi-intention IRL. Similarly, BinaryWorld has 32×32 states, five actions per state with a 30% chance of moving in a different random direction. But every state is randomly occupied with one of the two-color objects. The feature vector for each state consequently consists of a binary vector, encoding the color of each object in 3×3 neighborhood. Similar to M-ObjectWorld, six different intentions can be defined for each cell of M-BinaryWorld based on three rules: 1) four neighboring cells have color one, 2) five neighboring cells have color one, and 3) everything else (see Table 2). Since in M-BinaryWorld the reward depends on a higher representation for the basic features, the environment is arguably more challenging than the previous ones. Therefore, most of the experiments are carried in this environment. Fig. 2 (bottom three) shows a 8×8 zoom-in of M-BinaryWorld with three different reward functions and policies.

In order to assess the generalizability of the models, the experiments are also conducted on *transferred* environments. In transferred environments, the learned reward functions are re-evaluated on new randomized environments.

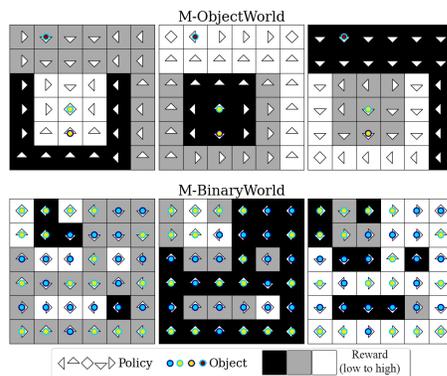


Fig. 2. 8×8 zoom-ins of M-ObjectWorld (top three) and M-BinaryWorld (bottom three) with three reward functions.

| Intention | Reward rule | | |
|-----------|-------------|-----|-----|
| | 1 | 2 | 3 |
| A | +5 | -10 | 0 |
| B | -10 | 0 | +5 |
| C | 0 | +5 | -10 |
| D | -10 | +5 | 0 |
| E | +5 | 0 | -10 |
| F | 0 | -10 | +5 |

Table 2. Reward values in M-ObjectWorld and M-BinaryWorld

5.2 Models

In this study, we compare our methods with existing approaches that can handle IRL with multiple intentions and constrain the experiments to model-based methods. The following models are evaluated on the benchmarks:

- EM-MLIRL(K), proposed by Babes et al. [5]. This method requires the number of experts’ intentions K to be known. To research the influence on setting K for this method, we use $K \in \{2, 3, 4\}$.
- DPM-BIRL, a non-parametric multi-intention IRL method proposed by Choi and Kim [9].
- SEM-MIIRL, our proposed solution based on stochastic EM.
- MCEM-MIIRL, our proposed solution based on Monte Carlo EM.
- KEM-MIIRL, a simplified variant of our approach where the concentration parameter is zero and the number of intentions are fixed to $K \in \{2, 5\}$.

5.3 Metric

Following the same convention used in [9], the imitation performance is evaluated by the average of expected value difference (EVD). The EVD measures the performance difference between the expert’s optimal policy and the optimal policy induced by the learned reward function. For $m \in \{1, 2, \dots, M\}$, $\text{EVD} = |V_{\tilde{R}^m}^{\tilde{\pi}^m} - V_{\tilde{R}^m}^{\pi^m}|$, where $\tilde{\pi}^m$ and \tilde{R}^m are the true policy and reward function for m^{th} demonstration, respectively, and π^m is the predicted policy under the predicted reward function demonstration. In all experiments, a lower average-EVD corresponds to better imitation performance.

5.4 Implementations details

In our experiments, we employed a fully connected neural network with five hidden layers of dimension 256 and a rectified linear unit for the base reward model, and a set of linear functions represents the intention-specific reward models. The reward network is trained for 200 epochs using Adam [17] with a fixed learning rate of 0.001. For easing the reproducibility of our work, the source code is shared with the community at <https://github.com/tue-mps/damiirl>.

5.5 Results

Each experiment is repeated for 6 times with different random environments, and the results are shown in the form of means (lines) and standard errors (shadings). The demonstration length for GridWorld is fixed to 40 time-steps and for both M-ObjectWorld and M-BinaryWorld is 8 time-steps.

Fig. 3 and Fig. 4 show the imitation performances of our SEM-MIIRL and MCEM-MIIRL in comparison with two baselines, EM-MLIRL(K) and DPM-BIRL, for varying number of demonstrations per reward function in original and transferred environments, respectively. Each expert is assigned to one out

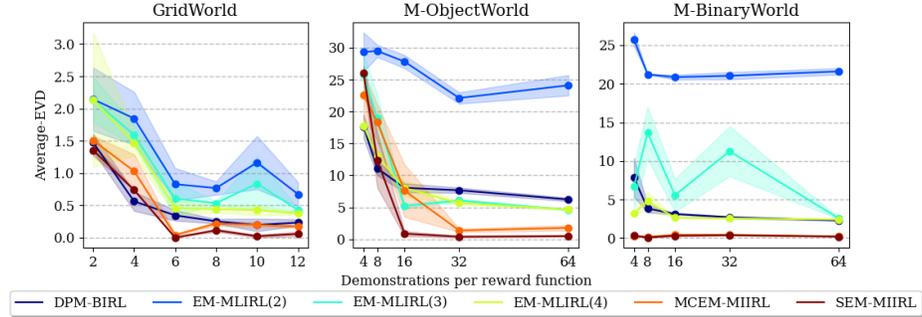


Fig. 3. Imitation performance in comparison with the baselines. Lower average-EVD is better.

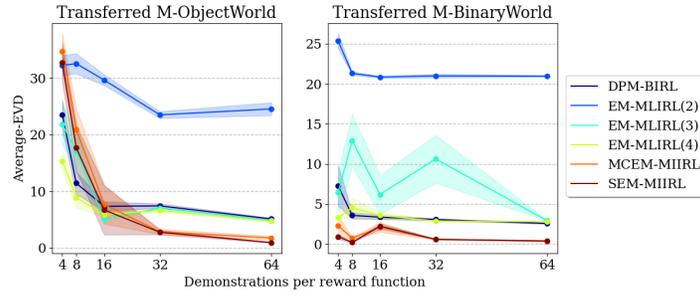


Fig. 4. Imitation performance in comparison with the baselines in transferred environments. Lower average-EVD is better.

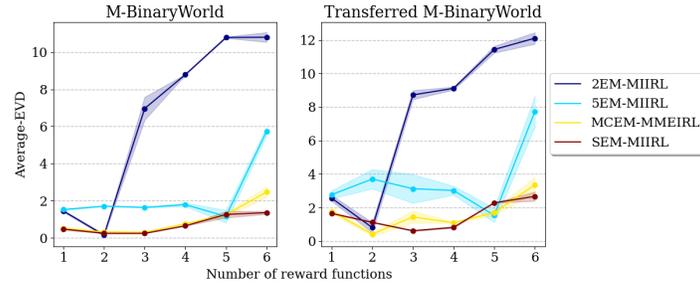


Fig. 5. Effects of overestimating/underestimating vs inferring the number of reward functions in original (left) and transferred (right) M-BinaryWorlds. Lower average-EVD is better

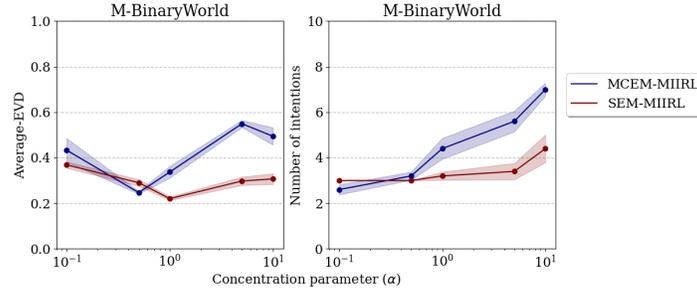


Fig. 6. Effects of α on Average-EVD (left) and number of predicted intentions (right). Lower average-EVD is better

of three reward functions (intentions A, B, and C in M-ObjectWorld and M-BinaryWorld) and the concentration parameter is set to one. The results show clearly that our methods achieve significant lower average-EVD errors when compared to existing methods, especially in nonlinear environments of M-ObjectWorld and M-BinaryWorld, with SEM-MIIRL slightly outperforming MCEM-MIIRL

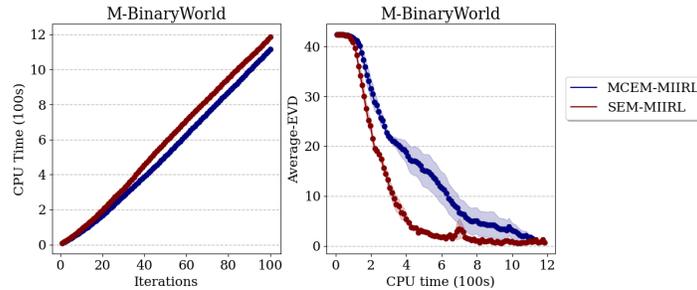


Fig. 7. Execution time (right) and Convergence (left). Lower average-EVD is better.

To address the importance of inferring the number of intentions, we have compared the performances of our SEM-MIIRL and MCEM-MIIRL with two simplified variants, 2EM-MIIRL and 5EM-MIIRL, where the concentration parameter is set to zero and the number of intentions is fixed and equal to 2 and 5, respectively. Fig. 5 shows the results of these comparisons for a varying number of true reward functions from one to six (from intention: {A} to {A, B, C, D, E, F}) in both original and transferred M-BinaryWorld. The number of demonstrations is fixed to 16 per reward function and $\alpha = 1$ for both SEM-MIIRL and MCEM-MIIRL. As depicted, overestimation and underestimation of the number of reward functions, as happens frequently in both 2EM-MIIRL and 5EM-MIIRL, deteriorate the imitation performance. This while the adaptability in SEM-MIIRL and MCEM-MIIRL yields to less sensitivity with changes in the number of true reward functions.

Further experiments are conducted to deeply assess and compare MCEM-MIIRL and SEM-MIIRL. Fig. 6 depicts the effects of the concentration parameter on both Average-EVD and number of predicted intentions. The number of demonstrations is fixed to 16 per reward function and intentions are $\{A, B, C\}$. As shown, the best value for the concentration parameter is between 0.5 to 1, with lower values leading to higher Average-EVD and lower number of predicted intentions, while higher values result in higher Average-EVD and higher number of predicted intentions for both MCEM-MIIRL and SEM-MIIRL. The final experiment is devoted to the convergence behavior of MCEM-MIIRL and SEM-MIIRL. The number of demonstrations is again fixed to 16 per reward function, intentions are $\{A, B, C\}$ and the concentration parameter is set to 1. As shown in Fig. 7 (left image), the per-iteration execution time of MCEM-MIIRL is lower than SEM-MIIRL. The main reason is that SEM-MIIRL evaluates the posterior distribution over all latent intentions. However, this extra operation guarantees faster converges of SEM-MIIRL, making it overall the more efficient than MCEM-MIIRL as can be seen in Fig. 7 (right image).

6 CONCLUSIONS

We proposed an inverse reinforcement learning framework to recover complex reward functions by observing experts whose behaviors originate from an unknown number of intentions. We presented two algorithms that are able to consistently recover multiple, highly nonlinear reward functions and whose benefits were pointed out through a set of experiments. For this, we extended two complex benchmarks for multi-intention IRL in which our algorithms distinctly outperformed the baselines. We also demonstrated the importance of inferring rather than underestimating or overestimating the number of experts' intentions

Having shown the benefits of our approach in inferring the unknown number of experts' intention from a collection of demonstrations via model-based RL, we aim to extend the same approach in model-free environments by employing approximate RL methods.

Acknowledgments

This research has received funding from ECSEL JU in collaboration with the European Union's 2020 Framework Programme and National Authorities, under grant agreement no. 783190.

References

1. Abbeel, P., Coates, A., Quigley, M. & Ng, A. An application of reinforcement learning to aerobatic helicopter flight. *Advances In Neural Information Processing Systems*. pp. 1-8 (2007)

2. Almingol, J., Montesano, L. & Lopes, M. Learning multiple behaviors from unlabeled demonstrations in a latent controller space. *International Conference On Machine Learning*. pp. 136-144 (2013)
3. Almingol, J. & Montesano, L. Learning multiple behaviours using hierarchical clustering of rewards. *2015 IEEE/RSJ International Conference On Intelligent Robots And Systems (IROS)*. pp. 4608-4613 (2015)
4. Andrieu, C., De Freitas, N., Doucet, A. & Jordan, M. An introduction to MCMC for machine learning. *Machine Learning*. **50**, 5-43 (2003)
5. Babes, M., Marivate, V., Subramanian, K. & Littman, M. Apprenticeship learning about multiple intentions. *Proceedings Of The 28th International Conference On Machine Learning (ICML-11)*. pp. 897-904 (2011)
6. Bishop, C. Pattern recognition and machine learning. (springer,2006)
7. Blei, D. & Jordan, M. Variational methods for the Dirichlet process. *Proceedings Of The Twenty-first International Conference On Machine Learning*. pp. 12 (2004)
8. Celeux, G. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*. **2** pp. 73-82 (1985)
9. Choi, J. & Kim, K. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. *Advances In Neural Information Processing Systems*. pp. 305-313 (2012)
10. Dimitrakakis, C. & Rothkopf, C. Bayesian multitask inverse reinforcement learning. *European Workshop On Reinforcement Learning*. pp. 273-284 (2011)
11. Echraïbi, A., Flocon-Cholet, J., Gosselin, S. & Vaton, S. On the Variational Posterior of Dirichlet Process Deep Latent Gaussian Mixture Models. *ArXiv Preprint arXiv:2006.08993*. (2020)
12. Gershman, S. & Blei, D. A tutorial on Bayesian nonparametric models. *Journal Of Mathematical Psychology*. **56**, 1-12 (2012)
13. Hausman, K., Chebotar, Y., Schaal, S., Sukhatme, G. & Lim, J. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. *Advances In Neural Information Processing Systems*. pp. 1235-1245 (2017)
14. Hsiao, F., Kuo, J. & Sun, M. Learning a Multi-Modal Policy via Imitating Demonstrations with Mixed Behaviors. *ArXiv Preprint arXiv:1903.10304*. (2019)
15. Jaynes, E. Information theory and statistical mechanics. *Physical Review*. **106**, 620 (1957)
16. Jin, J., Petrich, L., Dehghan, M., Zhang, Z. & Jagersand, M. Robot eye-hand coordination learning by watching human demonstrations: a task function approximation approach. *2019 International Conference On Robotics And Automation (ICRA)*. pp. 6624-6630 (2019)
17. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *ArXiv Preprint arXiv:1412.6980*. (2014)
18. Levine, S., Popovic, Z. & Koltun, V. Nonlinear inverse reinforcement learning with gaussian processes. *Advances In Neural Information Processing Systems*. pp. 19-27 (2011)
19. Li, Y., Schofield, E. & Gönen, M. A tutorial on Dirichlet process mixture modeling. *Journal Of Mathematical Psychology*. **91** pp. 128-144 (2019)
20. Li, Y., Song, J. & Ermon, S. Infogail: Interpretable imitation learning from visual demonstrations. *Advances In Neural Information Processing Systems*. pp. 3812-3822 (2017)
21. Lin, J. & Zhang, Z. Acgail: Imitation learning about multiple intentions with auxiliary classifier gans. *Pacific Rim International Conference On Artificial Intelligence*. pp. 321-334 (2018)

22. Michini, B. & How, J. Bayesian nonparametric inverse reinforcement learning. *Joint European Conference On Machine Learning And Knowledge Discovery In Databases*. pp. 148-163 (2012)
23. Nalisnick, E. & Smyth, P. Stick-Breaking Variational Autoencoders. *5th International Conference On Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. (2017)
24. Neal, R. Markov chain sampling methods for Dirichlet process mixture models. *Journal Of Computational And Graphical Statistics*. **9**, 249-265 (2000)
25. Ng, A., Russell, S. & Others Algorithms for inverse reinforcement learning.. *Icml*. **1** pp. 2 (2000)
26. Nguyen, Q., Low, B. & Jaillet, P. Inverse reinforcement learning with locally consistent reward functions. *Advances In Neural Information Processing Systems*. pp. 1747-1755 (2015)
27. Noothigattu, R., Yan, T. & Procaccia, A. Inverse Reinforcement Learning From Like-Minded Teachers. *Manuscript*. (2020)
28. Odom, P. & Natarajan, S. Actively Interacting with Experts: A Probabilistic Logic Approach. *Machine Learning And Knowledge Discovery In Databases*. pp. 527-542 (2016)
29. Papamarkou, T., Hinkle, J., Young, M. & Womble, D. Challenges in Bayesian inference via Markov chain Monte Carlo for neural networks. *ArXiv Preprint arXiv:1910.06539*. (2019)
30. Pavlov, D., Popescul, A., Pennock, D. & Ungar, L. Mixtures of conditional maximum entropy models. *Proceedings Of The 20th International Conference On Machine Learning (ICML-03)*. pp. 584-591 (2003)
31. Rajasekaran, S., Zhang, J. & Fu, J. Inverse Reinforce Learning with Nonparametric Behavior Clustering. *ArXiv Preprint arXiv:1712.05514*. (2017)
32. Ramachandran, D. & Amir, E. Bayesian Inverse Reinforcement Learning.. *IJCAI*. **7** pp. 2586-2591 (2007)
33. Russell, S. Learning agents for uncertain environments. *Proceedings Of The Eleventh Annual Conference On Computational Learning Theory*. pp. 101-103 (1998)
34. Shkurti, F., Kakodkar, N. & Dudek, G. Model-Based Probabilistic Pursuit via Inverse Reinforcement Learning. *2018 IEEE International Conference On Robotics And Automation (ICRA)*. pp. 7804-7811 (2018)
35. Sutton, R. & Barto, A. Reinforcement learning: An introduction. (MIT press,2018)
36. Wei, G. & Tanner, M. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal Of The American Statistical Association*. **85**, 699-704 (1990)
37. Wei, H., Chen, C., Liu, C., Zheng, G. & Li, Z. Learning to Simulate on Sparse Trajectory Data. *Machine Learning And Knowledge Discovery In Databases: Applied Data Science Track*. pp. 530-545 (2021)
38. Wulfmeier, M., Ondruska, P. & Posner, I. Maximum entropy deep inverse reinforcement learning. *ArXiv Preprint arXiv:1507.04888*. (2015)
39. Ziebart, B., Maas, A., Bagnell, J. & Dey, A. Maximum Entropy Inverse Reinforcement Learning. *Proceedings Of The 23rd National Conference On Artificial Intelligence - Volume 3*. pp. 1433-1438 (2008)

Appendix A

We assume that we have $M - 1$ demonstrated trajectories with a set of known latent intention vectors $\mathbf{H}^{-m} = \{\boldsymbol{\eta}^1, \boldsymbol{\eta}^2, \dots, \boldsymbol{\eta}^{m-1}, \boldsymbol{\eta}^{m+1}, \dots, \boldsymbol{\eta}^M\}$ with K intentions. Then, we have a new demonstrated trajectory $\boldsymbol{\tau}^m$ and the task is to obtain the latent intention vector $\boldsymbol{\eta}^m$, which can be a new intention $K + 1$, and update the reward parameters Ψ . We are willing to consider growing/infinite number of intentions.

In the case of K intentions, we define a Categorical prior distribution over $\mathbf{H} = \{\mathbf{H}^{-m}, \boldsymbol{\eta}^m\}$:

$$\begin{aligned} p(\mathbf{H}|\boldsymbol{\phi}) &= \prod_{m=1}^M \text{Cat}(\boldsymbol{\phi}) \\ &= \prod_{k=1}^K \phi_k^{M_k} \end{aligned} \quad (10)$$

where M_k is the number of trajectories with intention k and $\boldsymbol{\phi}$ is the vector of mixing coefficients $\boldsymbol{\phi} = \{\phi_1, \phi_2, \dots, \phi_K\}$ with prior distribution of:

$$\begin{aligned} p(\boldsymbol{\phi}) &= \text{Dir}(\alpha/K) \\ &= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{k=1}^K \pi_k^{\alpha/K-1} \end{aligned} \quad (11)$$

where α is the concentration parameter. The main problematic variable as $K \rightarrow \infty$ are the mixing coefficients. We marginalize out $\boldsymbol{\phi}$:

$$\begin{aligned} p(\mathbf{H}) &= \int p(\mathbf{H}|\boldsymbol{\phi})p(\boldsymbol{\phi}) \\ &= \frac{\Gamma(\alpha)}{\Gamma(M + \alpha)} \prod_{k=1}^K \frac{\Gamma(M_k + \alpha/K)}{\Gamma(\alpha/K)} \end{aligned} \quad (12)$$

Given that:

$$p(\mathbf{H}) = p(\boldsymbol{\eta}^m | \mathbf{H}^{-m})p(\mathbf{H}^{-m}) \quad (13)$$

we can define the conditional prior over $\boldsymbol{\eta}^m = \{\eta_1^m, \eta_2^m, \dots, \eta_K^m\}$ as:

$$p(\eta_k^m = 1 | \mathbf{H}^{-m}) = \frac{M_k^{-m} + \alpha/K}{M - 1 + \alpha} \quad (14)$$

where M_k^{-m} is the number of trajectories with intention k excluding $\boldsymbol{\tau}^m$. By letting $K \rightarrow \infty$, we reach:

$$p(\eta_k^m = 1 | \mathbf{H}^{-m}) = \frac{M_k^{-m}}{M - 1 + \alpha} \quad (15)$$

where $p(\eta_k^m = 1 | \mathbf{H}^{-m})$ is the prior probability of assigning the trajectory $\boldsymbol{\tau}^m$ to intention $k \in \{1, 2, \dots, K\}$. Since:

$$\sum_{k=1}^K p(\eta_k^m = 1 | \mathbf{H}^{-m}) = \frac{M-1}{M-1+\alpha} \neq 1 \quad (16)$$

we define $p(\eta_{K+1}^m = 1 | \mathbf{H}^{-m})$ as the prior probability of assigning the trajectory $\boldsymbol{\tau}^m$ to intention $k+1$:

$$\begin{aligned} p(\eta_{K+1}^m = 1 | \mathbf{H}^{-m}) &= 1 - \frac{M-1}{M-1+\alpha} \\ &= \frac{\alpha}{M-1+\alpha} \end{aligned} \quad (17)$$

Equations (15) and (17) are known as Chinese Restaurant Process [19].

Appendix B

Given the predictive distribution for m^{th} trajectory:

$$p(\boldsymbol{\tau}^m | \mathbf{H}^{-m}, \Psi) = \sum_{k=1}^{K+1} p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m}) \quad (18)$$

the following optimization problem can be defined $\forall m \in \{1, 2, \dots, M\}$ by employing the exchangeability property [12]:

$$\max_{\Psi} L^m(\Psi) = \log \sum_{k=1}^{K+1} p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m}) \quad (19)$$

The parameters Ψ can be estimated via Expectation Maximization (EM) [6]. Differentiating the log-likelihood function $L(\Psi)$ with respect to $\psi \in \Psi$ yields:

$$\begin{aligned} \nabla_{\psi} L^m &= \frac{\nabla_{\psi} \sum_{k=1}^{K+1} p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{\sum_{\hat{k}} p(\boldsymbol{\tau}^m | \eta_{\hat{k}}^m = 1, \Psi) p(\eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m})} \\ &= \sum_{k=1}^{K+1} \frac{\nabla_{\psi} p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{\sum_{\hat{k}} p(\boldsymbol{\tau}^m | \eta_{\hat{k}}^m = 1, \Psi) p(\eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m})} \end{aligned} \quad (20)$$

A standard trick in setting up the EM procedure is to introduce the posterior distribution over the latent intention vector $\boldsymbol{\eta}^m$ [6]:

$$\begin{aligned} \gamma_k^m &= p(\eta_k^m = 1 | \boldsymbol{\tau}^m, \mathbf{H}^{-m}, \Psi) = \frac{p(\boldsymbol{\tau}^m, \eta_k^m = 1 | \mathbf{H}^{-m}, \Psi)}{\sum_{\hat{k}=1}^{K+1} p(\boldsymbol{\tau}^m, \eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m}, \Psi)} \\ &= \frac{p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{\sum_{\hat{k}=1}^{K+1} p(\boldsymbol{\tau}^m | \eta_{\hat{k}}^m = 1, \Psi) p(\eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m})} \end{aligned} \quad (21)$$

Now the term under summation in (20) can be written as:

$$\begin{aligned}
 & \frac{\nabla_{\psi} p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{\sum_{\hat{k}=1}^{K+1} p(\boldsymbol{\tau}^m | \eta_{\hat{k}}^m = 1, \Psi) p(\eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m})} \\
 &= \frac{p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{\sum_{\hat{k}=1}^{K+1} p(\boldsymbol{\tau}^m | \eta_{\hat{k}}^m = 1, \Psi) p(\eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m})} \frac{\nabla_{\psi} p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})} \\
 &= \gamma_k^m \frac{\nabla_{\psi} p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})} \\
 &= \gamma_k^m \nabla_{\psi} \log p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})
 \end{aligned} \tag{22}$$

Performing the differentiation of the second term in (22) yields:

$$\begin{aligned}
 & \nabla_{\psi} \log p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m}) \\
 &= \nabla_{\psi} \log p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) + \nabla_{\psi} \log p(\eta_k^m = 1 | \mathbf{H}^{-m}) \\
 &= \nabla_{\psi} \log \left(\frac{\exp(R_k(\boldsymbol{\tau}^m, \psi_k))}{Z(k)} \right) \\
 &= \nabla_{\psi} (R_k(\boldsymbol{\tau}^m, \psi_k) - \log Z(k)) \\
 &= \nabla_{\psi} (R_k(\boldsymbol{\tau}^m, \psi_k) - \log \sum_{\boldsymbol{\tau}} \exp(R_k(\boldsymbol{\tau}, \psi_k))) \\
 &= \frac{dR_k(\boldsymbol{\tau}^m, \psi_k)}{d\psi} - \frac{\sum_{\boldsymbol{\tau}} \exp(R_k(\boldsymbol{\tau}, \psi_k)) \frac{dR_k(\boldsymbol{\tau}, \psi_k)}{d\psi}}{\sum_{\boldsymbol{\tau}} \exp(R_k(\boldsymbol{\tau}, \psi_k))} \\
 &= \frac{dR_k(\boldsymbol{\tau}^m, \psi_k)}{d\psi} - \sum_{\boldsymbol{\tau}} p(\boldsymbol{\tau} | \eta_k = 1, \Psi) \frac{dR_k(\boldsymbol{\tau}, \psi_k)}{d\psi} \\
 &= (\boldsymbol{\mu}(\boldsymbol{\tau}^m) - \mathbb{E}_{p(\boldsymbol{\tau} | \eta_k = 1, \Psi)}[\boldsymbol{\mu}(\boldsymbol{\tau})])^{\top} \frac{d\mathbf{R}_{\psi_k}(\boldsymbol{\tau})}{d\psi}
 \end{aligned} \tag{23}$$

Therefore (20) results in:

$$\nabla_{\psi} L = \sum_{k=1}^{K+1} \gamma_k^m (\boldsymbol{\mu}(\boldsymbol{\tau}^m) - \mathbb{E}_{p(\boldsymbol{\tau} | \eta_k = 1, \Psi)}[\boldsymbol{\mu}(\boldsymbol{\tau})])^{\top} \frac{d\mathbf{R}_{\psi_k}(\boldsymbol{\tau})}{d\psi} \tag{24}$$

which is known as the M-step. The posterior distribution over the latent intention vector $\boldsymbol{\eta}^m$ can be obtained as:

$$\begin{aligned}
 \gamma_k^m &= \frac{p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{\sum_{\hat{k}=1}^{K+1} p(\boldsymbol{\tau}^m | \eta_{\hat{k}}^m = 1, \Psi) p(\eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m})} \\
 &= \frac{b_0(s_0) \prod_{t=0}^{T-1} T(s_{t+1} | s_t, a_t) \pi_k(a_t | s_t) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{\sum_{\hat{k}=1}^{K+1} b_0(s_0) \prod_{t=0}^{T-1} T(s_{t+1} | s_t, a_t) \pi_{\hat{k}}(a_t | s_t) p(\eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m})} \\
 &= \frac{\prod_{t=0}^{T-1} \pi_k(a_t | s_t) p(\eta_k^m = 1 | \mathbf{H}^{-m})}{\sum_{\hat{k}=1}^{K+1} \prod_{t=0}^{T-1} \pi_{\hat{k}}(a_t | s_t) p(\eta_{\hat{k}}^m = 1 | \mathbf{H}^{-m})}
 \end{aligned} \tag{25}$$

Using (15) and (17) yields $\forall k \in \{1, 2, \dots, K\}$:

$$\gamma_k^m = \frac{M_k^{-m} \prod_{t=0}^{T-1} \pi_k(a_t | s_t)}{\alpha \prod_{t=0}^{T-1} \pi_{K+1}(a_t | s_t) + \sum_{\hat{k}=1}^K M_{\hat{k}}^{-m} \prod_{t=0}^{T-1} \pi_{\hat{k}}(a_t | s_t)} \quad (26)$$

and for $K + 1$:

$$\gamma_k^m = \frac{\alpha \prod_{t=0}^{T-1} \pi_k(a_t | s_t)}{\alpha \prod_{t=0}^{T-1} \pi_{K+1}(a_t | s_t) + \sum_{\hat{k}=1}^K M_{\hat{k}}^{-m} \prod_{t=0}^{T-1} \pi_{\hat{k}}(a_t | s_t)} \quad (27)$$

Which are known as the E-step.

Appendix C

The likelihood ratio for the m^{th} trajectory is obtained as:

$$\begin{aligned} \frac{p(\boldsymbol{\tau}^m | \eta_{k^*}^{*m} = 1, \Psi)}{p(\boldsymbol{\tau}^m | \eta_k^m = 1, \Psi)} &= \frac{b_0(s_0) \prod_{t=1}^{T_\tau} T(s_{t+1} | s_t, a_t) \pi_{k^*}(a_t^m | s_t^m)}{b_0(s_0) \prod_{t=1}^{T_\tau} T(s_{t+1} | s_t, a_t) \pi_k(a_t^m | s_t^m)} \\ &= \frac{\prod_{t=1}^{T_\tau} \pi_{k^*}(a_t^m | s_t^m)}{\prod_{t=1}^{T_\tau} \pi_k(a_t^m | s_t^m)} \end{aligned} \quad (28)$$

with $k \in \{1, 2, \dots, K\}$ and $k^* \in \{1, 2, \dots, K, K + 1\}$.s