Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Robust Bayesian Inverse Reinforcement Learning with Sparse Behavior Noise

Jiangchuan Zheng[1], **Siyuan Liu**[3], and **Lionel M. Ni**[1,2]
[1]Department of Computer Science and Engineering,
[2]Guangzhou HKUST Fok Ying Tung Research Institute,
The Hong Kong University of Science and Technology, Hong Kong, China
[3]Heinz College, Carnegie Mellon University, Pittsburgh, USA
jczheng@cse.ust.hk, siyuan@cmu.edu, ni@cse.ust.hk

Abstract

Inverse reinforcement learning (IRL) aims to recover the reward function underlying a Markov Decision Process from behaviors of experts in support of decision-making. Most recent work on IRL assumes the same level of trustworthiness of all expert behaviors, and frames IRL as a process of seeking reward function that makes those behaviors appear (near)-optimal. However, it is common in reality that noisy expert behaviors disobeying the optimal policy exist, which may degrade the IRL performance significantly. To address this issue, in this paper, we develop a robust IRL framework that can accurately estimate the reward function in the presence of behavior noise. In particular, we focus on a special type of behavior noise referred to as sparse noise due to its wide popularity in real-world behavior data. To model such noise, we introduce a novel latent variable characterizing the reliability of each expert action and use Laplace distribution as its prior. We then devise an EM algorithm with a novel variational inference procedure in the E-step, which can automatically identify and remove behavior noise in reward learning. Experiments on both synthetic data and real vehicle routing data with noticeable behavior noise show significant improvement of our method over previous approaches in learning accuracy, and also show its power in de-noising behavior data.

Introduction

In problems of *reinforcement learning* (RL), given the reward function in a state space (as a mapping from states to rewards) and the environment dynamics modeled as a Markov Decision Process (MDP), an agent learns to make decisions in such a way that maximizes the accumulated reward it will receive in the long term. Yet in practice, the reward function that an agent is using is usually unknown, and the goal is to recover that reward function based on sample observations of expert agents' sequential decision-making behaviors. This is referred to as *inverse reinforcement learning* (IRL). A basic assumption is that expert agents act in accordance with the optimal policy (as a mapping from states to actions) induced by the reward function, and hence the key intuitive notion in IRL is to search a reward function that makes their demonstrated behaviors appear (near)-optimal.

IRL can be used to achieve two objectives: *reward learning* and *apprenticeship learning*. In *reward learning*, the reward function in the task is itself of interest from knowledge discovery's perspective, for example, learning the latent cost of each road segment from vehicle routing data can facilitate various smart-city applications such as road type inference. In *apprenticeship learning*, the aim is to imitate the optimal policy in support of future decision-making, such as provide route recommendation to new drivers. Although expert agents' behaviors can directly be used to represent the optimal policy, they may only cover partial state space, contain noise, or be too sensitive to environment dynamics. A more promising way is to apply IRL to estimate the reward function, which completely determines the optimal policy and allows for generalization in face of environment change.

Most existing work on IRL assumes that all expert demonstrations are reliable or trustworthy. This is in the sense that i) they are all optimizing the same reward function as what we aim to learn in determining their policies, and ii) their sequential action-taking behaviors consistently obey the optimal policy. In practice, however, such assumptions do not always hold, and noisy or misleading demonstrations do exist: i) an agent may act to optimize a different reward function due to its unfamiliarity with the task, and ii) even if an agent has the correct knowledge of the reward function, it may behave in a fraudulent way, i.e., deliberately deviate from the optimal policy in choosing its actions to achieve certain purposes. Take the modeling of routing preferences of taxi drivers as an example. In this example, each road segment (i.e., state) is associated with a latent cost (i.e., a negative reward) which is jointly determined by a variety of latent factors such as speed limit and safety. Normally, a taxi driver is attempting to reach the destination as efficiently as possible by optimizing such latent costs, but there do exist exceptions: i) new taxi drivers, who are inexperienced, may bear a partially incorrect knowledge of road costs in mind and thus are not acting optimally in routing; ii) certain fraudulent drivers, albeit experienced, may deliberately detour or traverse inefficient roads in order to make more profits. In a word, noisy behaviors refer to those agent actions that apparently disobey the optimal policy in the task. It is clear that such anomalous actions can mislead traditional IRL methods to estimate an incorrect reward function and thus generate an inferior policy, which may result in poor decision-making performance. Motivated by this challenge, in this paper we study how to improve the *robustness* of IRL, i.e., how to

estimate the reward function accurately from expert demonstration data even in the presence of behavior noise.

We are particularly interested in one special type of behavior noise referred to as *sparse noise* due to its popularity in real-world applications (Zhang et al. 2011). A key trait of sparse noise is that most demonstrations are highly trustworthy, i.e., they are (near)-optimal w.r.t. the underlying reward function, while certain demonstrations may be significantly anomalous. This type of noise is commonly encountered in real-world applications in which expert agents are not manually filtered. For example, in taxi data, most taxi drivers are experienced and faithful in routing the passengers, while a few drivers may be less experienced or behave fraudulently.

Modeling sparse behavior noise in IRL can bring several crucial benefits. First, by doing so, we are able to recover the reward function more accurately from many imperfect demonstration data sets, and hence improve the performance of apprenticeship learning. Second, robust IRL essentially performs a de-noising process, which may help us achieve data cleaning, or automatically identify anomalous agents such as detecting fraudulent taxi drivers.

In this paper, we propose a probabilistic IRL framework that can recover the reward function in a way robust against a few significant outliers in the demonstration data. In particular, to automatically identify and separate noisy demonstrations from reliable ones, we extend the Bayesian IRL framework in (Ramachandran and Amir 2007) to explicitly model the trustworthiness of each demonstrated action as a latent variable. We then devise an expectation-maximization (EM) framework in which we alternate between estimating the rewards and inferring the reliability of each action until convergence. Those actions with small inferred reliability will be deemed as outliers and thus ignored in reward learning. To model sparse behavior noise, we change the reliability variable in a novel way and impose a Laplace prior on the new variable. This yet brings challenges to the inference step, which we tackle by exploiting the infinite mixture representation of Laplace distribution and designing an efficient variational inference procedure. Experimental results show that our robust model outperforms other methods in both reward learning and demonstration data de-noising.

Preliminaries

In the Bayesian IRL (BIRL) framework proposed by (Ramachandran and Amir 2007), consider a standard MDP with $\boldsymbol{R}$ being its reward function. Let $\mathcal{O} = \{(s_j, a_j)\}_{j=1\ldots N}$ denote a sequence of observations of an expert agent's behavior, where a_j is the action that the agent took when it was at state s_j. W.l.o.g., we use (s, a) to denote a particular state-action pair in the behavior sequence. The likelihood of (s, a) given $\boldsymbol{R}$ is defined in the form of a softmax function as:

$$P((s,a)|\alpha;\boldsymbol{R}) = \frac{e^{\alpha Q^{\pi^\star(\boldsymbol{R})}(s,a)}}{\sum_{a'} e^{\alpha Q^{\pi^\star(\boldsymbol{R})}(s,a')}} \quad (1)$$

where $\pi^\star(\boldsymbol{R})$ is the optimal policy w.r.t. to $\boldsymbol{R}$ estimated via RL, and $Q^{\pi^\star(\boldsymbol{R})}(s, a)$ is the expected accumulated reward after taking action a at state s under $\pi^\star$. (See (Sutton and Barto 1998) for details). $\alpha \in [0, 1]$ is the temperature parameter currently assumed to be 1. Since in RL, the optimal policy always chooses the action with the largest Q value at each state, Eq. (1) essentially quantifies the extent to which (s, a) obeys the optimal policy. The task of IRL is then to estimate $\boldsymbol{R}$ such that most state-action pairs have high likelihoods, i.e., appear to be (near)-optimal. Let $\mathcal{D} = \{\mathcal{O}^i\}_{i=1\ldots M}$ denote the data set containing M behavior sequences demonstrated by multiple expert agents. The likelihood of $\mathcal{D}$ is then naturally defined as $P(\mathcal{D}|\boldsymbol{R}) = \prod_{i=1}^{M}\prod_{j=1}^{N_i} P((s_j^i, a_j^i)|\boldsymbol{R})$. BIRL adopts a Bayesian perspective to estimate the most likely $\boldsymbol{R}$, i.e., it infers $P(\boldsymbol{R}|\mathcal{D}) \propto P(\mathcal{D}|\boldsymbol{R})P(\boldsymbol{R})$ using sampling method, where $P(\boldsymbol{R})$ is a prior distribution imposed on $\boldsymbol{R}$ (usually assumed to be Gaussian or uniform distribution).

Robust Bayesian IRL (RBIRL)

In this section, we first discuss our extension of BIRL to model sparse behavior noise in the demonstration data, and then elaborate on the learning and inference algorithms.

The Model

Recall the temperature parameter α in Eq. (1). α in fact quantifies the reliability or trustworthiness of the particular demonstration (s, a); the larger α is, the more sensitive that Eq. (1) will be to $\boldsymbol{R}$, and hence the learning of $\boldsymbol{R}$ will take (s, a) more into account. Ideally, a noisy demonstration should be associated with a small α so that the IRL algorithm will ignore it when learning $\boldsymbol{R}$. In this sense, α plays the role of *weight* for the observed demonstration (s, a).

(Ramachandran and Amir 2007) treats α as a single known parameter, which implicitly assumes the same reliability level for all demonstrations. This lacks robustness in the case where untrustworthy demonstrations are present in the data. In contrast, we assume that no knowledge is available about how reliable each demonstration (s, a) is, and instead aim to automatically infer such information from the data. To this end, we treat α as a demonstration-specific latent variable (i.e., each (s, a) is associated with a distinct α) whose value needs to be inferred. This is in spirit similar to *weighted regression* where the sample weights are unknown before learning. Given the reward function $\boldsymbol{R}$, the marginal likelihood of a demonstration (s, a) is then written as:

$$P((s,a)|\boldsymbol{R}) = \int_0^1 P((s,a)|\alpha;\boldsymbol{R})P(\alpha)d\alpha$$

where $P(\alpha)$ is the prior distribution on α which we shall define later. Recall that for sparse noise, most demonstrations are highly reliable while a few may be significantly anomalous. Considering the physical meaning of α, this is equivalent to expect that most α's are 1 or very close to 1, while a few α's are allowed to be significantly small near 0. Therefore, a key step to model sparse behavior noise is to impose a prior distribution on α which has the property that highly encourages α to be 1 when α is relatively large, while at the same time does not penalize too much for a relatively small α. With such a prior distribution, the IRL framework can learn reward function in a way that makes most demonstrations appear (near)-optimal while at the same time prevents a

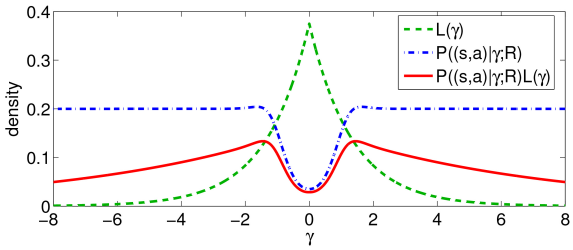

Figure 1: Graphical illustration of the Laplace prior (dashed), likelihood (dash-dotted), and posterior (solid), of γ for a *noisy* demonstration (s, a) w.r.t. the given $\boldsymbol{R}$

few anomalous demonstrations from "distorting" the reward function, which is what we desire.

However, it is difficult to find a well-known distribution defined over $[0, 1]$ (i.e., the domain of α) which has the desired property mentioned above. To address this issue, we change variable by defining $\alpha = e^{-\gamma^2}$, where γ is the new variable. Our goal now is to define a prior distribution on γ (i.e., $P(\gamma)$) which indirectly makes α have the desired property. Note that since $\lim_{\gamma\to\pm\infty} \alpha = 0$ and $\lim_{\gamma\to 0} \alpha = 1$, a desirable $P(\gamma)$ then should highly encourage γ to be 0 when $|\gamma|$ is small, while does not penalize too much when $|\gamma|$ is relatively large. In statistics, a well-known distribution which has such a property is *Laplace distribution*, due to its long tail in regions far away from 0 and its sharp peak in the neighborhood of 0 (see the dashed curve in Figure 1). We thus define $P(\gamma)$ as Laplace distribution, namely $P(\gamma) = L(\gamma) = \frac{\beta^2}{2} e^{-\beta^2|\gamma|}$, where β is the hyperparameter which measures the precision of γ. With the new latent variable γ, the conditional likelihood of a particular (s, a) is then written as:

$$P((s,a)|\gamma;\boldsymbol{R}) = \frac{e^{e^{-\gamma^2} Q^{\pi^\star(\boldsymbol{R})}(s,a)}}{\sum_{a'} e^{e^{-\gamma^2} Q^{\pi^\star(\boldsymbol{R})}(s,a')}}$$

and the marginal likelihood of (s, a) becomes:

$$P((s,a)|\boldsymbol{R}) = \int_{-\infty}^{+\infty} P((s,a)|\gamma;\boldsymbol{R}) L(\gamma) d\gamma$$

The posterior of γ, $P(\gamma|(s,a);\boldsymbol{R}) \propto P((s,a)|\gamma;\boldsymbol{R})L(\gamma)$, informs how reliable the demonstration (s, a) is w.r.t. the reward function $\boldsymbol{R}$. As an illustration, Figure 1 plots the likelihood function (dash-dotted) and the posterior distribution (solid) of γ for an artificial demonstration which notably deviates from the optimal policy induced by $\boldsymbol{R}$. As a result of such deviation, the likelihood function disfavors the region near 0 and hence the posterior concentrates its mass on γ of large absolute value (i.e., small α). Note that the long tail of Laplace prior (dashed) allows the posterior to place adequate probability mass on γ of large absolute value to make the expectation of α sufficiently small, so that the algorithm can treat this demonstration as a significant noise and thus ignore it in updating $\boldsymbol{R}$. In contrast, if instead a non-robust prior without a long tail such as Gaussian is used, the inferred expectation of α will not be small enough due to insufficient posterior mass placed on γ of large absolute value, and hence this particular noisy demonstration may mislead the learning of the reward function $\boldsymbol{R}$.

Learning and Inference

By treating $\boldsymbol{R}$ as model parameters and $\{\gamma_k\}_{k=1\ldots K}$ (K is the total number of state-action pairs in the data) as latent variables, we adopt the expectation-maximization (EM) algorithm to solve the robust IRL problem. The complete data log likelihood given parameters, namely $\log P(\mathcal{D}, \{\gamma_k\}_K|\boldsymbol{R})$, is written as:

$$\sum_{k=1}^{K} \left[\log P((s_k,a_k)|\gamma_k;\boldsymbol{R}) + \log L(\gamma_k)\right]$$

The so-called $\mathcal{Q}$-function of $\boldsymbol{R}$ given the currently learned $\boldsymbol{R}$ (denoted as $\boldsymbol{R}^{old}$), namely $\mathcal{Q}(\boldsymbol{R}|\boldsymbol{R}^{old})$, computes the expectation of the complete data log likelihood, which is:

$$\sum_{k=1}^{K} E_{\gamma_k\sim P(\gamma_k|(s_k,a_k);\boldsymbol{R}^{old})}\left[\log P((s_k,a_k)|\gamma_k;\boldsymbol{R})\right] \qquad (2)$$

where we omit the term $\log L(\gamma_k)$ which is unrelated to $\boldsymbol{R}$.

In the E-step, we need to infer the posterior distribution of γ: $P(\gamma|(s,a);\boldsymbol{R}) \propto P((s,a)|\gamma;\boldsymbol{R})L(\gamma)$ [1] (we omit the subscript k to avoid clutter). However, due to the non-conjugacy between $P((s,a)|\gamma;\boldsymbol{R})$ and $L(\gamma)$, the inference of $P(\gamma|(s,a);\boldsymbol{R})$ is intractable analytically. Therefore, we resort to sampling method to make approximate inference. But due to the non-smooth nature and some other characteristics of Laplace distribution, the sampling procedure may be difficult and inefficient (the detailed reasons shall be elaborated later in this subsection). To address this issue, we rewrite the Laplace distribution as an infinite mixture of Gaussian distributions according to (Lange and Sinsheimer 1993) as follows:

$$L(\gamma) = \int_0^{+\infty} \mathcal{N}(\gamma|0,\tau) Expon(\tau|\beta^2) d\tau$$

where $\mathcal{N}(\gamma|0,\tau) = \frac{1}{\sqrt{2\pi\tau}} e^{-\frac{\gamma^2}{2\tau}}$ is the Guassian distribution, and $Expon(\tau|\beta^2) = \frac{\beta^2}{2} e^{-\frac{\beta^2\tau}{2}}$ is the exponential distribution. Then, the marginal likelihood of (s, a), namely $P((s,a)|\boldsymbol{R})$, is written as:

$$\int_{-\infty}^{+\infty}\int_0^{+\infty} P((s,a)|\gamma;\boldsymbol{R})\mathcal{N}(\gamma|0,\tau)Expon(\tau|\beta^2)d\tau d\gamma$$

In this new formulation, each observation (s, a) is associated with two latent variables: γ and τ. Then, in order to infer $P(\gamma|(s,a);\boldsymbol{R}^{old})$, we need to infer the joint posterior distribution on (γ, τ), namely $P(\gamma,\tau|(s,a);\boldsymbol{R}^{old})$, which is proportional to:

$$P((s,a)|\gamma;\boldsymbol{R}^{old})\mathcal{N}(\gamma|0,\tau)Expon(\tau|\beta^2) \qquad (3)$$

[1] This is a legal density function of γ, which can be proved from the facts that $\int L(\gamma)d\gamma = 1$ and that $P((s,a)|\gamma;\boldsymbol{R})$ as a function of γ is lower and upper bounded.

Due to the intractability of this inference problem, we resort to variational method for approximate inference. But before doing that, we first change variable τ by defining $\lambda = \tau^{-1}$ for computational convenience which shall become clear later, and instead infer $P(\gamma, \lambda|(s,a); \boldsymbol{R}^{old})$. With the relation $\lambda = \tau^{-1}$, the posterior distribution on (γ, λ) is:

$$P(\gamma, \lambda|(s,a); \boldsymbol{R}^{old}) = P(\gamma, \tau|(s,a); \boldsymbol{R}^{old})|\frac{\partial \tau}{\partial \lambda}|$$

$$\propto P((s,a)|\gamma; \boldsymbol{R}^{old})\mathcal{N}(\gamma|0, \lambda^{-1})e^{-\frac{\beta^2}{2\lambda}}\frac{1}{\lambda^2}$$

$$= P((s,a)|\gamma; \boldsymbol{R}^{old})\sqrt{\frac{\lambda}{2\pi}}e^{-\frac{\lambda\gamma^2}{2}}e^{-\frac{\beta^2}{2\lambda}}\frac{1}{\lambda^2} \quad (4)$$

where we make use of Eq. (3). Denote the un-normalized version of $P(\gamma, \lambda|(s,a); \boldsymbol{R}^{old})$ (the right-hand side of $\propto$ in Eq. (4)) as $\tilde{P}(\gamma, \lambda; (s,a), \boldsymbol{R}^{old})$. By variational method (Bishop and Nasrabadi 2006), we make the full factorization assumption and find two variational distributions $q_1(\gamma)$, $q_2(\lambda)$ to minimize the KL-divergence from $q_1(\gamma)q_2(\lambda)$ to $P(\gamma, \lambda|(s,a); \boldsymbol{R}^{old})$. This is equivalent to alternately updating the following two equations until convergence:

$$\log q_2(\lambda) = E_{\gamma \sim q_1(\gamma)}[\log \tilde{P}(\gamma, \lambda; (s,a), \boldsymbol{R}^{old})] + C_1 \quad (5)$$

$$\log q_1(\gamma) = E_{\lambda \sim q_2(\lambda)}[\log \tilde{P}(\gamma, \lambda; (s,a), \boldsymbol{R}^{old})] + C_2 \quad (6)$$

where C_1, C_2 account for the logarithm of normalizing constants. From Eq. (5) and Eq. (4), we have:

$$\log q_2(\lambda) = E_{q_1(\gamma)}[\log\sqrt{\lambda}e^{-\frac{\lambda\gamma^2}{2}}e^{-\frac{\beta^2}{2\lambda}}\frac{1}{\lambda^2}] + C$$

$$= \log\sqrt{\lambda} - \frac{\lambda}{2}E_{q_1(\gamma)}[\gamma^2] - \frac{\beta^2}{2\lambda} + \log(\frac{1}{\lambda^2}) + C$$

where C accounts for terms irrelevant to λ. By straightforward mathematical manipulation, we have:

$$q_2(\lambda) \propto \sqrt{\frac{1}{\lambda^3}}\exp\Big(-\frac{\beta^2(\lambda - \frac{\beta}{\sqrt{E_{q_1(\gamma)}[\gamma^2]}})^2}{2\frac{\beta^2}{E_{q_1(\gamma)}[\gamma^2]}\lambda}\Big) \quad (7)$$

Eq. (7) states that $q_2(\lambda)$ is an *inverse Gaussian distribution* (Shuster 1968). Its expectation has a closed form [2] as:

$$E_{q_2(\lambda)}[\lambda] = \frac{\beta}{\sqrt{E_{q_1(\gamma)}[\gamma^2]}} \quad (8)$$

On the other hand, by manipulating Eq. (6), we have:

$$q_1(\gamma) \propto P((s,a)|\gamma; \boldsymbol{R}^{old})\mathcal{N}(\gamma|0, E_{q_2(\lambda)}[\lambda]^{-1}) \quad (9)$$

from which we know that $q_1(\gamma)$ is an even function, and hence its variance $Var_{q_1(\gamma)}[\gamma] = E_{q_1(\gamma)}[\gamma^2]$. [3]

We then use Eq. (9) and Eq. (8) to alternate between updating $q_1(\gamma)$ and $q_2(\lambda)$ until convergence. However, from Eq. (9), there is no closed form for $E_{q_1(\gamma)}[\gamma^2]$, which needs to be used in Eq. (8). To solve this problem, we now propose an efficient importance sampling-based method to estimate $E_{q_1(\gamma)}[\gamma^2]$. In particular, we aim to get samples from $q_1(\gamma)$ using importance sampling method. To do this, for efficiency concern, it is desirable to have a proposal distribution which is easy to sample and at the same time roughly approximates $q_1(\gamma)$ in shape. Nevertheless, such a proposal distribution is difficult to find since $q_1(\gamma)$ may be multi-modal [4]. To tackle this, we note that $q_1(\gamma)$ is symmetric, and over $[0, +\infty)$ or $(-\infty, 0]$, $q_1(\gamma)$ is a puedoconcave function and hence is uni-modal. Specifically, denote the right-hand side of Eq. (9) as $\tilde{q_1}(\gamma)$, then $q_1(\gamma) = \frac{1}{Z_1}\tilde{q_1}(\gamma)$ where Z_1 is the normalizing constant. We then have:

$$E_{q_1(\gamma)}[\gamma^2] = \frac{1}{Z_1}\int_{-\infty}^{+\infty}\tilde{q_1}(\gamma)\gamma^2 d\gamma = \frac{2}{Z_1}\int_{0}^{+\infty}\tilde{q_1}(\gamma)\gamma^2 d\gamma$$

where we have used the fact that $\tilde{q_1}(\gamma)\gamma^2$ is an even function due to the evenness of $\tilde{q_1}(\gamma)$, and that it is equal to 0 at the point $\gamma = 0$. We now define:

$$\tilde{q_1^{'}}(\gamma) = \begin{cases} \tilde{q_1}(\gamma) & if\ \ \gamma \geq 0 \\ 0 & otherwise \end{cases} \quad (10)$$

then we have:

$$E_{q_1(\gamma)}[\gamma^2] = \int_{-\infty}^{+\infty} q_1^{'}(\gamma)\gamma^2 d\gamma = E_{q_1^{'}(\gamma)}[\gamma^2]$$

where $q_1^{'}(\gamma) = \frac{1}{Z_1^{'}}\tilde{q_1^{'}}(\gamma)$ is a legal density function over $(-\infty, +\infty)$, and $Z_1^{'} = \frac{Z_1}{2}$ is the normalizing constant. By Eq. (10) and the one-side puedo-concativity of $q_1(\gamma)$, $q_1^{'}(\gamma)$ is uni-modal over $(-\infty, +\infty)$. This fact, together with the presence of the Gaussian in Eq. (9), determines that $q_1^{'}(\gamma)$ can be well approximated by a Gaussian distribution, which is easy to sample. We then apply Laplace approximation (Bishop and Nasrabadi 2006) to approximate the density function $q_1^{'}(\gamma)$ as a Gaussian, and treat it as the proposal distribution. In particular,

$$\tilde{q_1^{'}}(\gamma) \simeq \tilde{q_1^{'}}(\gamma^\star)e^{-\frac{A(\gamma-\gamma^\star)^2}{2}} \quad (11)$$

where $\gamma^\star$ is the mode of the function $\tilde{q_1^{'}}(\gamma)$, and $A = -\bigtriangledown\bigtriangledown\log\tilde{q_1^{'}}(\gamma^\star)$. $\gamma^\star$ can be found easily by gradient descent as $\tilde{q_1^{'}}(\gamma)$ is uni-modal [5]. The Gaussian approximation is thus $\mathcal{N}(\gamma|\gamma^\star, A^{-1})$. Then, by the standard derivation of importance sampling (Bishop and Nasrabadi 2006), we have:

$$E_{q_1(\gamma)}[\gamma^2] = \frac{E_{\mathcal{N}(\gamma|\gamma^\star, A^{-1})}\Big[\frac{\tilde{q_1^{'}}(\gamma)}{\tilde{q_1^{'}}(\gamma^\star)e^{-\frac{A(\gamma-\gamma^\star)^2}{2}}}\gamma^2\Big]}{E_{\mathcal{N}(\gamma|\gamma^\star, A^{-1})}\Big[\frac{\tilde{q_1^{'}}(\gamma)}{\tilde{q_1^{'}}(\gamma^\star)e^{-\frac{A(\gamma-\gamma^\star)^2}{2}}}\Big]} \quad (12)$$

[2] This shows the advantage of changing variable as $\lambda = \tau^{-1}$.

[3] $Var_{q_1(\gamma)}[\gamma]$ provides an intuitive estimation of the expected value of α, i.e., the larger the posterior variance of γ is, the smaller the posterior expectation of α will be.

[4] It can be deduced from the expression of $q_1(\gamma)$ that, when the observed action a is not the optimal action at state s w.r.t. the current reward function, $q_1(\gamma)$ must be multi-modal.

[5] The derivations of the gradient and the second derivative of $\tilde{q_1^{'}}(\gamma)$ are straightforward, and are omitted here to save space.

So to estimate $E_{q_1(\gamma)}[\gamma^2]$, we firstly obtain samples $\{\gamma_i\}_{i=1..n}$ from $\mathcal{N}(\gamma|\gamma^\star, A^{-1})$, and then estimate

$$E_{q_1(\gamma)}[\gamma^2] \simeq \frac{\sum_{i=1}^n w_i \gamma_i^2}{\sum_{i=1}^n w_i} \quad (13)$$

where the weight $w_i = \tilde{q_1'}(\gamma_i)/\tilde{q_1'}(\gamma^\star)e^{-\frac{A(\gamma_i-\gamma^\star)^2}{2}}$. Eq. (11) implies that most weights will be close to 1, which makes the sampling quite efficient. At this point, the motivation for rewriting $L(\gamma)$ as an infinite mixture of Gaussian is becoming clearer: i) $P((s,a)|\gamma;\boldsymbol{R}^{old})L(\gamma)$ over $[0,+\infty)$ is not puedo-concave and may have both local maximum and local minimum, making it hard to apply Laplace approximation, ii) the shape of $P((s,a)|\gamma;\boldsymbol{R}^{old})L(\gamma)$ over $[0,+\infty)$ is significantly different from Gaussian due to the Laplace prior, which can make the sampling procedure based on Gaussian proposal distribution inefficient, and iii) the Laplace approximation requires the evaluation of the gradient and the second derivative of the posterior of γ, which may be hard if the prior is non-smooth like Laplace distribution.

In summary, the variational inference procedure alternates between updating $E_{q_2(\lambda)}[\lambda]$ and $E_{q_1(\gamma)}[\gamma^2]$ using Eq. (8) and Eq. (13) until convergence. From the $\mathcal{Q}$-function in Eq. (2), the E-step needs to estimate $E_{q_1(\gamma)}[e^{-\gamma^2}]$ [6]. We adopt the similar importance sampling method discussed before to estimate $E_{q_1(\gamma)}[e^{-\gamma^2}]$. The only difference is that the function is now changed from γ^2 to $e^{-\gamma^2}$. Note that the estimation of $E_{q_1(\gamma)}[e^{-\gamma^2}]$ only needs to be done after the variational procedure converges. So we reserve all the samples $\{\gamma_i\}$ and weights $\{w_i\}$ obtained in the final iteration of the variational procedure, and use them to estimate $E_{q_1(\gamma)}[e^{-\gamma^2}]$.

In practice, we can add a penalty factor η to control the regularization strength of the Laplace prior, i.e., the complete likelihood of a single observation (s,a) is written as $\log P((s,a)|\gamma;\boldsymbol{R}) + \eta \log L(\gamma)$. Equivalently, Eq. (9) becomes $q_1(\gamma) \propto P((s,a)|\gamma;\boldsymbol{R}^{old})^{\frac{1}{\eta}}\mathcal{N}(\gamma|0, E_{q_2(\lambda)}[\lambda]^{-1})$, and other derivations remain unchanged.

Recall the $\mathcal{Q}$-function in Eq. (2), for a particular observation (s,a), the $\mathcal{Q}$-function in terms of α is

$$E_{q(\alpha)}[\alpha]A_a - E_{q(\alpha)}[\log \sum_{a'} e^{\alpha A_{a'}}]$$

where $A_a = Q^{\pi^\star(\boldsymbol{R})}(s,a)$, $A_{a'} = Q^{\pi^\star(\boldsymbol{R})}(s,a')$. We have already discussed the estimation of $E_{q(\alpha)}[\alpha]$. Although it is intractable to compute $E_{q(\alpha)}[\log \sum_{a'} e^{\alpha A_{a'}}]$, there is quite a few work on finding an efficient upper bound of the expectation of log-sum-exponentials. Here we use the bound derived in (Bouchard 2007) to get the following upper bound:

$$\sum_{a'} \Big(\frac{1}{2}E_{q(\alpha)}[\alpha]A_{a'} + \lambda(\xi_{a'})A_{a'}^2 E_{q(\alpha)}[\alpha^2]\Big) + C \quad (14)$$

where $\lambda(\xi) = \frac{1}{2\xi}(\frac{1}{1+e^{-\xi}} - \frac{1}{2})$, and C is an irrelevant constant in terms of optimizing $\boldsymbol{R}$. This holds for any $\xi_{a'} \in (0, +\infty)$. $E_{q(\alpha)}[\alpha^2]$ can be estimated in a straightforwardly similar way as we estimate $E_{q(\alpha)}[\alpha]$, which is omitted here.

Now we have completed the E-step. In the M-step, we re-estimate $\boldsymbol{R}$ to maximize the lower bound of the $\mathcal{Q}$-function as derived in Eq. (14), using the Policy Walk sampling algorithm proposed by (Ramachandran and Amir 2007). Its basic idea is to sample $\boldsymbol{R}$ by random walk using Metropolis-Hastings algorithm until the Markov chain mixes to the posterior of $\boldsymbol{R}$ and then return the sample mean. We alternate between the E-step and the M-step until convergence. Then we obtain the reward function $\boldsymbol{R}$, and the reliability of each demonstration, namely $\{E_{q(\alpha)}[\alpha]\}$. Algorithm 1 summarizes the main steps of our RBIRL model. The complexity of one EM iteration (Line 3-14) is $O(KM + N)$, where M is the number of iterations that the variational inference procedure for one demonstration takes in the E-step, which is the major computational overhead. But due to our carefully designed importance sampling procedure, the E-step converges very quickly, as shall be shown in the experiments. N is the number of sampling iterations in the M-step.

Algorithm 1: Robust Bayesian IRL (RBIRL)

Input : $MDP = \{S, A, T\}$, $\{(s_k, a_k)\}_{k=1...K}$, β
Output: $\boldsymbol{R}$, $\{E_{q(\alpha_k)}[\alpha_k]\}_{k=1...K}$

1 Initialize $\boldsymbol{R}$ to $\boldsymbol{R}^0$;
2 **while** $\boldsymbol{R}$ *not converge* **do**
3 **for** *k=1 to K* **do**
4 // a is $E_{q_2(\lambda)}[\lambda]$, b is $E_{q_1(\gamma)}[\gamma^2]$;
5 Initialize $a = a_0$, $b = b_0$;
6 **while** *a, b not converge* **do**
7 Approximate $q_1(\gamma) \propto P((s_k,a_k)|\gamma;\boldsymbol{R})\mathcal{N}(\gamma|0,a^{-1})$ as a Gaussian $\mathcal{N}(\gamma|\gamma^\star, A^{-1})$ by Eq. (11);
8 Collect samples $\{\gamma\}$ from $\mathcal{N}(\gamma|\gamma^\star, A^{-1})$, and compute associated weights $\{w\}$ according to Eq. (12);
9 Update b using Eq. (13);
10 Update $a = \beta/\sqrt{b}$. (Eq. (8));
11 **end**
12 Use $\{\gamma\}$ and $\{w\}$ obtained in the final round of the iteration (Line 6-11) to estimate $E_{q(\alpha_k)}[\alpha_k]$ and $E_{q(\alpha_k)}[\alpha_k^2]$;
13 **end**
14 Infer $P(\boldsymbol{R}|\mathcal{D}) \propto P(\mathcal{D}|\boldsymbol{R})P(\boldsymbol{R})$ using Policy Walk algorithm and set $\boldsymbol{R}$ as the sample mean. ($P(\mathcal{D}|\boldsymbol{R})$ is replaced as the exponential of the lower bound of the $\mathcal{Q}$-function using $\{E_{q(\alpha_k)}[\alpha_k], E_{q(\alpha_k)}[\alpha_k^2]\}_{k=1...K}$);
15 **end**

Experiments

We carry out comparative experiments on both synthetic grid world-based data and real-world vehicle routing data to show the accuracy of our method in reward and policy learning from noisy demonstration data, as well as its ability of de-noising behavior data.

[6] By $\alpha = e^{-\gamma^2}$, $E_{q_1(\gamma)}[e^{-\gamma^2}]$ is an approximation of $E_{q(\alpha)}[\alpha]$, where $q(\alpha)$ is the posterior distribution of α given (s,a) and $\boldsymbol{R}^{old}$.

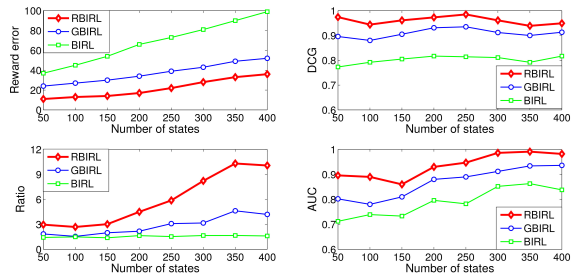

Figure 2: Comparison of IRL performance on 4 metrics

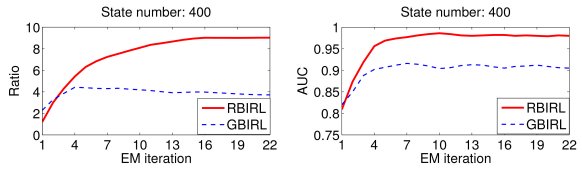

Figure 3: Comparison of RBIRL and GBIRL on the convergence of the performance of noisy trajectory detection

Experiments on Synthetic Data

We assess the performance of several IRL methods quantitatively on grid worlds [7] of various state numbers. Grid world is a board with each grid (i.e., state) in it associated with a negative reward (i.e., cost), and the agent aims to learn to escape to one of the terminal states in a way that incurs the smallest accumulated cost. For a particular grid world setting, we generate negative rewards drawn randomly from i.i.d. Gaussian priors $\mathcal{N}(0, 100)$. Then we simulate many expert agents on it and use their trajectories as the input to the IRL algorithm. Each action in the trajectory is drawn from the multinomial distribution over actions determined by Eq. (1) w.r.t. the generated reward function, until the trajectory reaches the terminal state. To introduce behavior anomalies, we randomly choose 10% of the trajectories and draw their α's randomly from the Beta distribution $Beta(10, 90)$ in generating them. To mimic realistic setting, we further add small random noise to all other reliable trajectories by drawing their α's randomly from $Beta(90, 10)$. In implementing our model, to facilitate the task of noisy trajectory detection, we assign a single α to all actions in one trajectory. This only requires a slight change to our model, i.e., the likelihood term in the posterior of γ is replaced by the product of the likelihoods of all actions in one trajectory. For parameter setting, we set β^2 to 1.5 and η to 0.5 after careful tuning.

We compare our RBIRL with two other methods: GBIRL and BIRL. GBIRL is the same as our model except that the prior on γ is changed to Guassian distribution, which makes the computation easier. This model is still noise-aware, but is less robust than RBIRL, which we shall show in the results. For fairness of comparison, we set the variance of Gaussian prior in a way to make it achieve the same density as Laplace prior at 0. BRIL is the method proposed in (Ramachandran and Amir 2007), which assumes all demonstrations as trustworthy without any mechanism to deal with the potential noise. We use four metrics for comparison: (a) the mean square error between the learned and the true reward functions, (b) a ranking-based measure, i.e., the *discounted cumulative gain* (DCG) (Järvelin 2002) of the optimal decision (the sequence in which actions at one state are ranked according to their Q values under the true rewards) w.r.t. the action probabilities computed using Eq. (1) based on the learned rewards, averaged over all states, (c) the ratio of the average of $E_{q(\alpha)}[\alpha]$ over all reliable trajectories to that over all anomalous ones, and (d) the *area under curve* (AUC) (Fawcett 2004) of the task of classifying trajectories to be reliable or anomalous, using the learned $E_{q(\alpha)}[\alpha]$ as the probability of one trajectory being reliable. These metrics cover various aspects in the evaluation of a robust IRL model, namely reward learning accuracy (metric (a)), policy imitation performance (metric (b), a higher value suggests the model imitates the optimal policy better), and anomaly detection accuracy (metrics (c) and (d), a higher ratio or AUC suggests the model can better differentiate noisy behaviors from normal ones). Note that BIRL is noise-unaware in its design. To compare with it on metrics (c) and (d), we calculate the average action likelihood for each trajectory [8] and deem it as the reliability for that trajectory.

Figure 2 shows the comparison results. As can be seen, our robust model consistently outperforms other methods, both in terms of IRL and noisy behavior detection. Due to its unawareness of the anomalous behaviors in the data, BIRL performs poorly on all metrics. Although GBIRL can also distinguish noisy behaviors from reliable ones in reward learning, its Gaussian prior does not faithfully model the properties of sparse behavior noise, and hence it gives poorer performance compared with RBIRL. Particularly, Figure 3 compares RBIRL and GBIRL in how their noise detection performance change as EM proceeds for a specific state configuration. The results show that RBIRL is significantly better. This is because the Laplace prior used strongly penalizes small noises while tolerates big noises, which fits the sparse noise assumption better than the Gaussian prior.

For the efficiency of RBIRL, we find in the experiments that the importance sampling procedure takes less than 80 samples on average to reach a stable estimation of $E_{q_1(\gamma)}[\gamma^2]$, and that the variational inference converges in less than 10 iterations most of the time. Hence, the E-step does not render itself as a severe bottleneck of RBIRL.

Experiments on Real Data

We apply our framework to a real-world taxi trajectory data set collected from Shenzhen, China, to model routing preferences of taxi drivers. The aim is to recover the latent cost on each road segment reflective of various unseen factors that determine drivers' routing policies, which can facilitate various smart-city applications such as route recommendation. We assume that taxi drivers who are carrying passengers are

[7] Grid world is an MDP widely used in reinforcement learning research. Please refer to (Sutton and Barto 1998) for more details.

[8] The likelihood of each action is calculated by Eq. (1) with α set to 1 using the learned reward function.

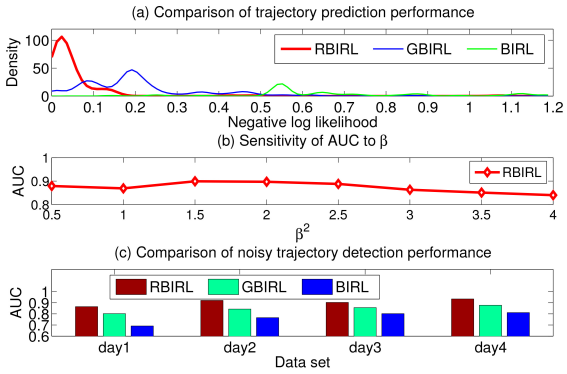

Figure 4: Comparison and sensitivity results on real data

attempting to reach their destinations as efficiently as possible by optimizing such costs. To test the robustness of our method, we create a training data set in which most trajectories are passenger-taking while a few (10%) are unoccupied (with no passengers) or are apparently detouring labeled by humans. Since taxi drivers normally adopt a drastically different routing policy when carrying no passengers (i.e., to pick passengers in a short time rather than to reach a certain destination efficiently), such "vacant" trajectories can be deemed as anomalous for our purpose.

For illustration, we restrict the state space in a region containing about 500 road segments and select over 3000 trajectories, with 15% containing no unoccupied trajectories withheld as the testing set. We apply each method to the training data [9], which contains a few anomalous trajectories. The parameter settings are the same as used in the synthetic experiment. Since no ground-truth of road latent costs is available, we compare each model's confidence in predicting the occupied (reliable) trajectories observed in the testing set. Such confidence is quantified by the predictive likelihood of each testing trajectory, which is defined as the average of the likelihood of each action in it (see Footnote 8). A higher predictive likelihood suggests that a model is more robust against anomalous trajectories in the training set. Figure 4(a) plots the distributions of negative log likelihood (NLL) of testing trajectories for each model. The strongly concentrated mass of RBIRL around 0.04 suggests our method outperforms the other two significantly (smaller NLL is better). This shows that sparse behavior noise is quite common in real-world vehicle routing data, and by explicitly modeling such noise, the performance of IRL can be greatly improved. Additionally, in Figure 4(c), we create four different training sets picked from 4 days and compare the performance of unoccupied trajectory detection (AUC) among three methods. The results show that our model is superior in de-noising behavior data compared with other methods, due to its robust nature. To study the sensitivity of our method to the hyperparameter β in the Laplace prior, we vary β^2 from 0.5 to 4.0 and plot the change of AUC in Figure 4(b), using the same training data as in Figure 4(a). It can be seen that although our method is not so sensitive to β, its performance may degrade when β gets relatively large. This is because a larger β makes the Laplace prior less tolerant to noises, and hence renders the algorithm more vulnerable to noisy behaviors.

[9] Trajectories ending at different destinations correspond to different MDPs as their terminal states differ. Yet multiple MDPs share the same rewards except for the terminal state, so for each destination we treat it as the terminal state of an MDP and perform RL using the shared rewards, the results of which are used to calculate the likelihood of all trajectories ending at that destination.

Related Work

IRL was initially proposed by (Ng and Russell 2000) and their solution exploits the explicit specification of optimal policy. Later (Ramachandran and Amir 2007) and (Neu and Szepesvári 2007) tackle the more practical case where the optimal policy is presented implicitly as expert demonstrations, via sampling and gradient methods, respectively. Instead of learning the reward function over the entire state space, (Abbeel and Ng 2004), (Ziebart et al. 2008), (Kalakrishnan, Theodorou, and Schaal 2010) and (Ratliff, Bagnell, and Zinkevich 2006) represent rewards as linear combinations of state features and learn those feature weights via maximum margin or maximum entropy principle. What combines these two lines of work is (Levine, Popovic, and Koltun 2011), which learns the reward of each state while exploiting the similarity in state features by imposing a Gaussian process prior on the reward function. There are also other variants of IRL models such as active learning-based IRL (Lopes, Melo, and Montesano 2009) and multi-task IRL (Dimitrakakis and Rothkopf 2012). However, all such work ignores the potential presence of anomalous demonstrations, and hence lacks robustness. Only a few work deals with imperfect demonstrations, such as (Silva, Costa, and Lima 2006) and (Grollman and Billard 2011). Yet they assume the "goodness" of each demonstration can be obtained by the access to some external routines, and learn how to avoid such behaviors. In contrast, our method is superior in that it automatically distinguishes noisy behaviors from reliable ones in reward learning. For work on routing preference modeling from trajectory data, (Zheng and Ni 2013) restricts the latent cost of road segment as speed limit while ignoring other latent factors. (Liu et al. 2013) and (Ziebart et al. 2008) consider various latent factors by framing the problem as IRL, yet they ignore the anomalous trajectories in real-world vehicle data, which makes their methods not robust. (Zhang, Yeung, and Xing 2012) proposed a method to model sparse noise in images, while we transfer the technique to IRL scenario, which is more challenging due to its more complex learning structure.

Conclusion and Future Work

We propose a robust IRL framework which can estimate the reward function and optimal policy accurately from expert behavior data with anomalous demonstrations. Our framework is particularly suitable for data with sparse behavior noise, which is commonly encountered in practice. The sparsity of behavior noise is modeled by using Laplace prior, and the learning is based on an effective variational EM algo-

rithm. Experiments on synthetic and real data demonstrate the superiority of our framework in policy learning accuracy and de-noising performance in face of imperfect data. Future work shall investigate how to extend our model to incorporate state features usually observed in real applications.

Acknowledgements

This research was supported by Hong Kong, Macao and Taiwan Science & Technology Cooperation Program of China under Grant No. 2012DFH10010, Nansha S&T Project under Grant No. 2013P015, NSFC under Grant No. 61300030, 973 Program under Grant No. 2014CB340303, Huawei Corp. under Contract YBCB2009041-27, the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA) and the Pinnacle Lab at Singapore Management University. We sincerely thank the anonymous reviewers for their insightful comments to help improve this paper.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the* 21^{st} *International Conference on Machine Learning*.

Bishop, C. M., and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 1. Springer New York.

Bouchard, G. 2007. Efficient bounds for the softmax function, applications to inference in hybrid models. In *Workshop for Approximate Bayesian Inference in Continuous/Hybrid Systems*.

Dimitrakakis, C., and Rothkopf, C. A. 2012. Bayesian multitask inverse reinforcement learning. In *Recent Advances in Reinforcement Learning*. Springer. 273–284.

Fawcett, T. 2004. Roc graphs: Notes and practical considerations for researchers. *Machine Learning* 31:1–38.

Grollman, D. H., and Billard, A. 2011. Donut as i do: Learning from failed demonstrations. In *Proceedings of IEEE International Conference on Robotics and Automation*, 3804–3809. IEEE.

Järvelin, K. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)* 20(4):422–446.

Kalakrishnan, M.; Theodorou, E.; and Schaal, S. 2010. Inverse reinforcement learning with pi^2. In *The Snowbird Workshop*.

Lange, K., and Sinsheimer, J. S. 1993. Normal/independent distributions and their applications in robust regression. *Journal of Computational and Graphical Statistics* 2(2):175–198.

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

Liu, S.; Araujo, M.; Brunskill, E.; Rossetti, R.; Barros, J.; and Krishnan, R. 2013. Understanding sequential decisions via inverse reinforcement learning. In *Proceedings of the* 14^{th} *IEEE International Conference on Mobile Data Management*, volume 1, 177–186. IEEE.

Lopes, M.; Melo, F.; and Montesano, L. 2009. Active learning for reward estimation in inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 31–46.

Neu, G., and Szepesvári, C. 2007. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proceedings of the* 23^{rd} *Conference on Uncertainty in Artificial Intelligence*, 295–302.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the* 17^{th} *International Conference on Machine Learning*, 663–670.

Ramachandran, D., and Amir, E. 2007. Bayesian inverse reinforcement learning. In *Proceedings of the* 20^{th} *International Joint Conferences on Artificial Intelligence*, 2586–2591.

Ratliff, N. D.; Bagnell, J. A.; and Zinkevich, M. A. 2006. Maximum margin planning. In *Proceedings of the* 23^{rd} *International Conference on Machine Learning*, 729–736.

Shuster, J. 1968. On the inverse gaussian distribution function. *Journal of the American Statistical Association* 63(324):1514–1516.

Silva, V. F. d.; Costa, A. H. R.; and Lima, P. 2006. Inverse reinforcement learning with evaluation. In *Proceedings of IEEE International Conference on Robotics and Automation*, 4246–4251.

Sutton, R. S., and Barto, A. G. 1998. *Introduction to reinforcement learning*. MIT Press.

Zhang, D.; Li, N.; Zhou, Z.-H.; Chen, C.; Sun, L.; and Li, S. 2011. ibat: detecting anomalous taxi trajectories from gps traces. In *Proceedings of the* 13^{th} *International Conference on Ubiquitous Computing*, 99–108. ACM.

Zhang, Y.; Yeung, D.-Y.; and Xing, E. P. 2012. Supervised probabilistic robust embedding with sparse noise. In *Proceedings of the* 26^{th} *AAAI Conference on Artificial Intelligence*, 1226–1232.

Zheng, J., and Ni, L. M. 2013. Time-dependent trajectory regression on road networks via multi-task learning. In *Proceedings of the* 27^{th} *AAAI Conference on Artificial Intelligence*, 1048–1055.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the* 23^{rd} *AAAI Conference on Artificial Intelligence*, 1433–1438.