Investigating the Extent to which Inverse Reinforcement Learning can Learn Rewards from Noisy Demonstrations

Charalampos Perdikis[1]

Supervisor(s): Dr. Luciano Cavalcante Siebert[1]**, Angelo Caregnato Neto**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2023

Name of the student: Charalampos Perdikis
Final project course: CSE3000 Research Project
Thesis committee: Dr. Luciano Cavalcante Siebert, Angelo Caregnato Neto, Dr. Jana Weber

Abstract

Inverse Reinforcement Learning (IRL) aims to recover a reward function from expert demonstrations in a Markov Decision Process (MDP). The objective is to understand the underlying intentions and behaviors of experts and derive a reward function based on their reasoning, rather than their exact actions. However, expert demonstrations can be influenced by various types of noise (e.g., from random behavior) which can affect their accuracy and effectiveness in solving the MDP. This research investigates the capability of IRL to recover reward functions from noisy demonstrations. Three types of noises, namely Random Action Noise, Random Bias Noise, and Sparse Noise, are introduced and modeled. Demonstrations are generated with these noises, and the corresponding reward functions are recovered. Comparisons are made between the noisy and optimal recovered rewards using various metrics. The results indicate that IRL exhibits certain tolerance level against Random Events and Sparse Noise, while being more vulnerable to Random Bias Noise.

1 Introduction

Reinforcement Learning (RL) [6] is a dynamic and promising field of machine learning that focuses on training agents using a predefined reward function to generate a policy and solve a problem. Inspired by how humans learn from trial and error, RL aims to create an optimal policy that learns from received feedback, which either rewards or punishes an action. At its core, RL involves an agent taking actions in an environment with a set of states and possible actions per state, called a Markov Decision Process (MDP), where actions and resulting states elicit positive or negative feedback.

Inverse Reinforcement Learning (IRL) [11] takes an inverse approach by inferring the underlying reward function from observed behavior. Creating a reward function, as in RL, can be challenging for complex scenarios were the path of solving a problem is not strictly defined. Hence, IRL extracts implicit knowledge from expert demonstrations to generate a reward function. The goal is to understand the reasoning behind the actions taken by experts and derive a reward function based on their intentions rather than their exact actions. This approach enables an agent to solve a problem differently from the expert but with the same objective in mind.

Typically, the expert demonstrations that will be fed in an IRL are disturbed by some kind of noise originating from human error, biased behavior, sensor measurement error, and so on [10]. These noises might cause the generation of an unreliable reward function that does not yield optimal behavior. Previous studies have explored the impact of noise on IRL algorithms, shedding light on the challenges and potential solutions. Chen et. al. [3] investigated the effects of sparse noise in the IRL process and proposed a method to handle noise by devising an Expectation-Maximization algorithm, which can automatically identify and remove behavior noise in reward learning. Another study [5] examined IRL that derives the reward from imperfect demonstrations. They introduced a unified RL algorithm that can learn robustly and outperform existing baselines.

While the aforementioned studies have explored the impact of noise on IRL algorithms, certain questions regarding noisy demonstrations in IRL remain unanswered. Firstly, the combined effects of multiple sources of noise on the IRL process require further exploration. Secondly, existing research has primarily focused on specific types of noise, such as sensor noise or environmental variability, without considering the impact of expert suboptimality. Lastly, the types of noise can occur in demonstration data and their possible impact on the learning process. How can different noise characteristics (e.g., random noise) affect the performance of IRL.

This research paper aims to investigate the extent to which IRL can learn rewards from noisy demonstrations. By evaluating and analyzing the impact of various types of noise on reward function generation, we contribute to a deeper understanding of IRL's robustness while also identifying the limitations of specific IRL algorithms when faced with noisy demonstrations.

The structure of this paper is as follows: Section 2 contains the necessary backround theory and knowledge to help understand the paper, and Section 3 provides an explanation of the chosen IRL algorithm, and the methodology used for this research. Section 4 presents an experimental setup for evaluating the impact of different types of noise on the IRL algorithm. Section 5 contains the results obtained from the experiments and analyses them. Section 6 discusses responsible research aspects pertaining to this study. Section 7 which has an in-depth discussion of the results. Finally, Section 8 summarizes the main contributions, outlines future research directions, and concludes the paper.

2 Backround

In this section, we will introduce and explain some concepts and preliminaries required for the understanding of this paper and research.

2.1 Markov Decision Process

A Markov Decision Process (MDP) [8] is a mathematical framework used to model decision-making in situations where outcomes are influenced by both stochastic (random) events and the actions taken by an agent. A MDP is represented as a tuple of components e.g., (S, A, T, R, γ):

- States (S): The set of all possible states in the system.
- Actions (A): The set of all the agent's possible actions.
- Transition probabilities (T): The transition function specifies the probability of moving from one state to another when a particular action is taken.
- Rewards (R): The reward function assigns a numerical value to each state or state-action pair.
- Discount factor (γ): Value between 0 and 1 that determines the importance of immediate versus future rewards.

The agent's goal in an MDP is to find an optimal policy, denoted as π^*, which is a set of state-action pairs (s, a), that specifies the action to take in each state to maximize the expected cumulative rewards over time. The policy can be deterministic (e.g., always choosing the same action in each state) or stochastic (e.g., selecting actions with probabilities based on a distribution).

2.2 Maximum Entropy Inverse Reinforcement Learning

Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) [18] is a method used to infer the underlying reward function in a Markov Decision Process (MDP) based on observed expert behavior using feature-expectation matching [1]. Unlike traditional IRL approaches that seek a single reward function to explain the expert's behavior, MaxEnt IRL takes a different approach. It aims to find a reward function that not only replicates the observed behavior but also maximizes the entropy or uncertainty of the expert's actions. By maximizing entropy, MaxEnt IRL allows for a broader range of possible policies that could explain the expert's demonstrated actions. This approach captures the idea that the expert's behavior might not be uniquely determined by a single reward function, but rather by a set of reward functions that exhibit similar behavior. MaxEnt IRL offers flexibility in capturing the expert's preferences and decision-making patterns, even in situations where there may be multiple valid interpretations of the observed expert behavior.

2.3 Optimal Policy Using Bellman Optimality Equation

The Bellman Optimality Equation [17] provides a recursive relationship between the value function $V(s)$ and the expected returns $Q(s, a)$ for each state-action pair, allowing for the computation of the optimal policy through iterative updates. By iteratively improving the value function $V(s)$ and selecting actions that maximize the expected return, the optimal policy π^* can be derived and used for decision-making in the MDP. This process is refer to as Value Iteration [2] and calculates the optimal policy π^*, which indicates the best action selection in each state of the MDP.

3 Methodology

To address the research question, the following procedure was established. The following is a step-by-step overview:

1. Select an appropriate IRL algorithm and MDP environment for the study.
2. Conduct a thorough review of existing literature to identify and characterize the types of noise that may be present in expert demonstrations.
3. Generate the Optimal Policy and corresponding optimal demonstrations for the chosen MDP.
4. Model each identified noise type and generate noisy demonstrations for the MDP.
5. Utilize the optimal and noisy demonstrations as input for the IRL algorithm to obtain both the optimal and noisy recovered rewards.
6. Establish metrics to facilitate a comparative analysis between the noisy recovered rewards and the optimal recovered reward.
7. Analyze the obtained metrics and data, and derive meaningful conclusions based on the results.

By following this methodology, we aim to gain insights into the impact of different noise types on the recovery of rewards from expert demonstrations.

It was crucial to select an appropriate IRL algorithm to evaluate the extent to which IRL can learn rewards from noisy demonstrations. Several potential approaches were considered: Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) [18], which utilizes the principle of maximizing entropy to determine the recovered reward, Adversarial Inverse Reinforcement Learning (AIRL) [4], employing an adversarial reward learning formulation to recover robust reward functions capable of accommodating changes in dynamics, and Nonlinear Inverse Reinforcement Learning with Gaussian Processes [8], which employs Gaussian processes to learn the reward as a nonlinear function, departing from the linear feature-based representation. Given the substantial amount of literature and supporting material available, we opted to employ MaxEnt IRL for our research, as this choice provided us with a higher confidence level in completing the study within the given time constraints.

A certain implementation of MaxEnt IRL was used in this reasearch to conduct the necessary experiments [9]. This implementation follows the maximum entropy algorithm of [18] with exponentiated gradient descent [7].

To utilize the aforementioned MaxEnt IRL implementation, we generated demonstrations that simulated expert/agent behavior in solving a Grid World MDP. Initially, we employed the Bellman Optimality Equation and Value Iteration to construct the optimal policy, which was then used to generate the optimal demonstrations. Subsequently, we fed these demonstrations into MaxEnt IRL to obtain the optimal recovered reward, allowing us to compare it with other recovered rewards from the noisy demonstrations.

Next, we proceeded to create the noisy demonstrations to simulate potential data noise. We will consider three types of noise: Random Events Noise, Random Bias Noise, and Sparse Noise. The selection of the noise types we modelled was made based on the most understandable and feasible literature references that were found.

Starting with Random Events Noise, inspired by [12], this type of noise refers to unexpected and unpredictable events that could occur during the execution of actions in each task or environment. These events introduce variations and deviations from the intended behavior, resulting in noisy demonstrations. In a Grid World MDP context, Random Event Noise could arise due to external disturbances, system failures, human error, or uncontrollable environmental conditions that cause the experts to select a random action instead of the optimal one. Accounting for noise that introduces randomness in demonstrations is crucial for robustly estimating the agent's true intentions and effectively applying IRL techniques.

Continuing with Random Bias Noise, influenced by [13], it refers to the introduction of random behavior observed in all demonstrations in a similar way, resulting in a form of bias. In the Grid World environment, this type of noise is observed in distorted choices made in specific states on the grid. The cause of this random bias can be noisy or imprecise control actions, imperfect sensing or observation, problems in the environment itself, or even cognitive bias shared among the experts attempting to solve the problem. By considering and addressing random bias noise in a Grid World environment, we can build reliable models robust to random biases from either the environment or the expert's behavior or have a threshold of biased distortion allowed.

Lastly, Sparse Noise [16] describes demonstrations where most of them are considered to have an optimal behavior, meaning they were created according to the optimal policy, while some demonstrations have significant anomalies. This noise was considered due to it being frequently observed in real-world applications [15]. It occurs when the experts are not manually filtered, therefore, some experts might make errors or attempt to solve the problem using approaches that differ from the optimal one. Examining the effects of Sparse Noise can be quite beneficial as it indicates what percentage of the population of demonstrations can have significant errors without affecting the recovered behavior and outcome.

Using the generated demonstrations as input for the Max-Ent IRL algorithm, we obtained the recovered rewards, which we compared using various metrics. Firstly, Reward Deviation measured the discrepancy between the recovered reward from the noisy demonstrations and the reward recovered from the optimal demonstrations. In addition to directly comparing rewards, we aimed to assess whether the recovered noisy reward exhibited the desired behavior. To accomplish this, we executed a deterministic policy on the noisy recovered reward, generating noisy trajectories that reflected the behavior of the noisy demonstrations in solving the MDP. To analyze these trajectories, we employed more metrics such as Goal Achieved, which checked if the goal was reached, Path Length Similarity, indicating similarity in the number of visited states compared to the optimal path, and Trajectory Similarity, which measured the resemblance between the noisy trajectories of the recovered noisy and recovered optimal rewards. These metrics will be discussed in detail in the subsequent section.

4 Experiments

4.1 Environment

To investigate the extent to which IRL can learn rewards from noisy demonstrations, we conducted experiments using an implementation of a 5x5 Grid World [9]. The Grid World represents a Markov Decision Process (MDP) in which each cell of the grid corresponds to a state with a predefined reward. The agent can take one of four possible actions (up, down, right, left) from each state. By selecting actions, the agent interacts with the environment, transitioning between states and receiving rewards based on these transitions. The objective of the agent is to discover a policy that maximizes the expected cumulative reward over a finite time horizon. In our experiments, we employed a discount factor (γ) of 0.9, which determines the trade-off between immediate and future rewards. A higher discount factor emphasizes long-term rewards, while a lower discount factor prioritizes immediate rewards. The starting state of the agent was set to (0,0), and the terminal state was (4,4).

4.2 Optimal Behavior

To establish a baseline, we defined the ground-truth reward function for the 5x5 Grid World (see Figure 1). Utilizing this ground-truth reward function in the Bellman Optimality Equation, we used value iteration to compute a stochastic policy indicating the probability of selecting each action for every state, enabling the agent to achieve the goal of the Grid World MDP. This optimal policy is represented as π^*. We then generated a set of 200 trajectories, denoted as T^*, representing the optimal behavior that the experts should exhibit when solving the problem. Subsequently, we fed these optimal trajectories T^* into our Max Ent IRL algorithm and obtained an optimal recovered reward function R^*.

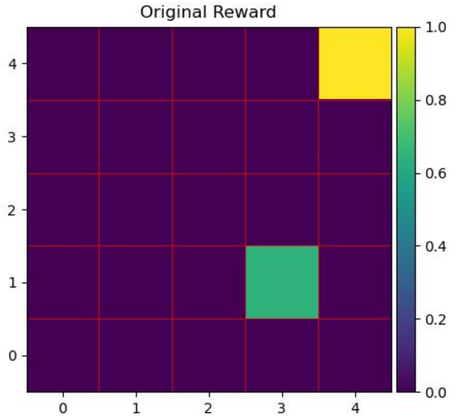

Figure 1: The ground-truth reward function of the 5x5 Grid World environment used in our experiments. The color bar indicates the reward that will be awarded to the agent for visiting that tile, with dark blue being 0 and bright yellow being 1.

4.3 Noisy Behavior

To investigate the impact of noise, we modeled the noises discussed in Section 3 and generated trajectories accordingly.

We defined Random Events Noise with probability p, indicating the probability of the expert selecting a random action. To simulate Random Events Noise, we first created the optimal policy π^* using Value Iteration. Then, using this policy, we started generating a set of 200 noisy trajectories T_n that correspond to experts solving the problem. To incorporate Random Events Noise during the generation of the trajectories, we used a uniform distribution $X \sim U(a, b)$ that generates a value of X between the range $[a, b]$. For every step in the trajectory generation procedure, if $p \geq X \sim U(0, 1)$, then the optimal policy would not be followed, and instead, a random action would be taken in the current state. This means that for every trajectory and every decision/step made in the generation of the trajectory, there is a probability p of taking a random action.

Moving to Random Bias Noise with probability p, we generated the optimal stochastic policy π^* using Value Iteration and modified it to create the noisy policy π_{n}. The modification was made based on a uniform distribution $X \sim U(a,b)$ that generates a value of X between the range $[a,b]$. For every state-action pair (s,a) in π^*, if $p \geq X \sim U(0,1)$, then the probability of (s,a) in π_{n} will be equal to $1/NumberOfPossibleActions$ (e.g., for our environment's case 1/4), otherwise, (s,a) in π_{n} will have the same probability as in π^*. In the end, we normalized the noisy policy π_{n} to make sure all probabilities of all actions per state, sum up to 1. This results in a noisy policy with some distortion in random state-action pairs. After constructing the noisy policy π_{n}, we generated a set of 200 trajectories, denoted as T_{n}, representing the behavior exhibited by experts in the presence of Random Bias Noise with probability p.

Finally, the Sparse Noise is defined with an influence factor q, indicating what percentage of the expert's trajectories will have significant error, and with probability p, indicating how severe the error will be. To generate trajectories containing this noise we started by creating the optimal policy π^* (similar to the Optimal Behaviour subsection). Then we generated 200 $*$ (1 - q) trajectories using the optimal policy π^*, and 200 $* \, q$ trajectories with significant error. For the anomalous trajectories, we used a uniform distribution $X \sim U(a,b)$ that generates a value of X between the range [a,b] and set $p = 0.5$. For every step in the trajectory generation procedure, if $p \geq X \sim U(0,1)$, then the optimal policy will not be followed and instead a random action will be made from the set of actions that does not contain the optimal one. This means that unlike the technique used for the Random Events Noise, here for every trajectory and every decision/step made in the generation of the trajectory, there is a probability p of taking a random action that will not be the optimal one. By adding the two sets of trajectories together we created the T_{n} which contains both optimal and significantly anomalous trajectories.

After obtaining the noisy trajectories T_n of every noise type we utilized them as input for our Maximum Entropy IRL algorithm, resulting in noisy recovered rewards R_{n} for reach noise. It should be noted that the trajectory generation procedure has an upper limit of iterations, specifically 25000 steps, which can be considered infinite in the context of a 5x5 Grid World scenario.

4.4 Metrics of Comparison

We defined several metrics to draw meaningful conclusions by comparing the optimal recovered reward R^* with the noisy recovered reward R_{n} of each noise type. The metric selection was determined to enable direct comparison of the recovered rewards while also assessing whether the recovered reward can effectively lead to a successful solution of the defined MDP environment. To ensure robust results, we repeated the process of creating a noisy policy π_{n}, generating trajectories T_{n}, and obtaining the recovered reward R_{n} a total of 100 times per noise. The data obtained from these 100 iterations were used to compute the following metrics:

Reward Deviation

Firstly, we employed the reward deviation metric, which measures the disparity between the recovered noisy reward R_{n} and the optimal recovered reward R^*. We modeled the reward deviation in two ways:

- We computed an element-wise absolute difference of the two recovered reward matrices, followed by summing up the differences:

$$\begin{aligned}
\text{Optimal Reward} : R^* &= \rho_0, \rho_1, \rho_2, \ldots, \rho_{23}, \rho_{24} \\
\text{Noisy Reward} : R_{\mathrm{n}} &= r_0, r_1, r_2, \ldots, r_{23}, r_{24} \\
\text{Deviation} : D_i &= |R^* - R_{\mathrm{n}}| \\
&= |\rho_0 - r_0|, \ldots, |\rho_{24} - r_{24}| \\
&= \delta_0, \ldots, \delta_{24} \\
\text{Total Deviation} : D^{\text{total}} &= \sum_{i=0}^{24} D_i = \delta_0 + \delta_1 + \ldots + \delta_{24}
\end{aligned}$$

- We computed an element-wise absolute difference of the two recovered reward matrices, resulting in an element-wise fraction when divided by the optimal recovered reward matrix. Finally, we computed the average of these fractions:

$$\begin{aligned}
\text{Optimal Reward} : R^* &= \rho_0, \rho_1, \rho_2, \ldots, \rho_{23}, \rho_{24} \\
\text{Noisy Reward} : R_{\mathrm{n}} &= r_0, r_1, r_2, \ldots, r_{23}, r_{24} \\
\text{Deviation} : D_i &= |R^* - R_{\mathrm{n}}| \\
&= |\rho_0 - r_0|, \ldots, |\rho_{24} - r_{24}| \\
&= \delta_0, \ldots, \delta_{24} \\
\text{Fractions of Deviation} : F_i &= \frac{D_i}{R_i^*} = \frac{\delta_0}{\rho_0}, \frac{\delta_1}{\rho_1}, \ldots, \frac{\delta_{24}}{\rho_{24}} \\
&= f_0, f_1, \ldots, f_{24} \\
\text{Total Deviation } (\%) : D^{\text{total}} &= \left(\frac{\sum_{i=0}^{24} F_i}{24} \right) \times 100\%
\end{aligned}$$

Failure to Achieve Goal

This metric measures whether or not the recovered noisy reward R_{n} allows reaching the final state of the Grid World. To determine this, we execute a deterministic optimal policy on the recovered noisy reward, which provides the sequence of actions to be taken for each state. Starting from the initial state of the Grid World, we iterate until we reach the terminal state. The maximum number of iterations allowed is 25000, which, as mentioned earlier, can be considered infinite for this Grid World MDP. If the terminal state is not reached within these iterations, it indicates a significant issue with the generated deterministic policy and, consequently, with the recovered noisy reward.

Path Length Similarity

Path Length Similarity metric measures the number of steps made from the initial state to the terminal state. Similarly with the Failure to Achieve Goal metric we execute a deterministic optimal policy on the recovered noisy reward R_{n}, and then starting from the initial state we count how many states we visit until we reach the terminal one. Then we compare this path length with the path obtained by following the same procedure on the optimal recovered reward R^* and log the difference. This metric is used only if the terminal state of the Grid World is reached.

Trajectory Similarity

Our final metric, Trajectory similarity, calculates the Euclidean distance between the path generated from the recovered noisy reward R_n and the recovered reward R^* (as described in the aforementioned metrics). This calculation is performed by summing the Euclidean distances between each visited state, step by step. If the length of the path generated using the noisy reward R_n is longer than that of the optimal recovered reward R^* or vice versa, we repeat the procedure using the final step of the path from the optimal recovered reward until the computation is complete. It's important to note that this metric is applicable only if the terminal state of the Grid World is reached. The following enumerated procedure outlines the calculation of the Trajectory Similarity metric:

1. Let P_1 be the optimal path consisting of states $S_1, S_2, \ldots, S_n$, and P_2 be the noisy path consisting of states $S'_1, S'_2, \ldots, S'_m$.
2. Convert the grid coordinates of each state to their corresponding Euclidean coordinates. Let $C(S_i)$ represent the Euclidean coordinates of state S_i and $C(S'_j)$ represent the Euclidean coordinates of state S'_j.
3. Calculate the Euclidean distance between points (S_i, S'_j) starting from i = 1 and j =1 of corresponding states in the two paths:
$$\sqrt{(C(S_i)_x - C(S'_j)_x)^2 + (C(S_i)_y - C(S'_j)_y)^2}$$
4. Record the distance in a variable TotalEuclideanDistance = TotalEuclideanDistance + distance(S_i, S'_j)
5. If index i $\neq$ n then increase i by 1
6. If index j $\neq$ m then increase j by 1
7. Repeat from step 3 until i = n and j = m

After following these steps TotalEuclideanDistance contains the total distance difference of the optimal and the noisy path. Then obtain the AverageEuclideanDistance by dividing with the number of iterations.

5 Results and Analysis

After conducting experiments on the three noise types described in Section 3 and defined in Section 4, we have obtained results and measurements that are crucial for addressing the research questions. These results will be presented and interpreted to shed light on the research objectives.

For each noise type, we computed both the noisy and optimal trajectories and subsequently employed the Maximum Entropy IRL algorithm to derive the optimal and noisy recovered rewards. This process was repeated 100 times per noise type to ensure reliable and representative results, allowing us to draw meaningful conclusions. To assess the disparity between the optimal and noisy recovered behaviors, we utilized the metrics outlined in Section 4. After generating the necessary graphs, we conducted a detailed analysis of the results, categorized by noise type, which are presented below. Note that Figures 8-16 are in Appendix A.

5.1 Random Events Noise

For demonstrations containing Random Events noise, we observed interesting patterns. In Figures 2 and 8, it is evident that the reward deviation increases more steadily up to noise probabilities of 0.4 and 0.5, after which the rate increase rapidly. Additionally, Figure 8 shows that the percentage of reward deviation is relatively low in the early stages of the noise, but significantly higher in the later stages reaching 120% of deviation for the largest noise probability.

Considering the standard deviation among the 100 iterations, depicted by the red lines in Figure 2, we find that the deviation between iterations remains relatively insignificant even at high probabilities of the noise. This suggests that the total reward deviation metric provides a representative measure of the deviations.

Moving on to the analysis of the Failure to Achieve Goal metric, as illustrated in Figure 3, we observed that the goal is reached for every noise probability up to 0.7. This indicates that although the reward may deviate from the optimal recovered reward, it is still sufficient to successfully reach the terminal state of the Grid World problem. For probabilities 0.7 there are only 2 cases in which re goal was not reached and for 0.8 they increase to 17.

Next, we examined the Path Length and the Euclidean Distance, presented in Figures 9 and 10, respectively. Interestingly, all recovered paths have the same length as the optimal path, indicating that the noisy recovered trajectories maintain a similar trajectory length to the optimal one. However, a slight deviation in the path is observed when the noise reaches its highest value of 0.8.

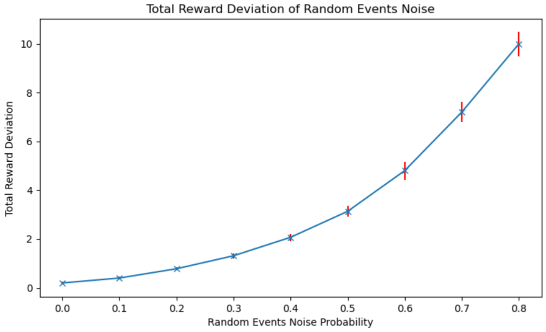

Figure 2: Total reward deviation of recovered rewards with Random Events Noise compared to optimal recovery, with standard deviation for 100 iterations.

5.2 Random Bias Noise

Moving on to Random Bias Noise, we analyzed the results presented in Figures 4 and 11. It is evident that the reward deviation caused by this noise is relatively small, with the highest percentage being around 35%, displaying an upward

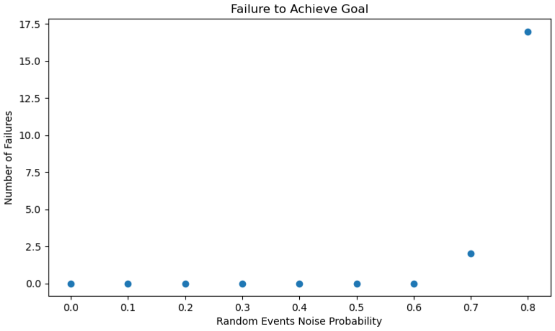

Figure 3: Frequency of noisy recovered rewards with Random Events Noise failing to reach terminal state with deterministic optimal policy, for 100 iterations.

trend. However, it is worth noting that the standard deviation depicted in Figure 4 is large, indicating that the deviations from the optimal reward were scattered. This suggests that relying solely on the reward deviation metric may not be sufficient to draw conclusive insights from these results.

Examining the Failure to Achieve Goal metric in Figure 5, we observed a significant impact of Random Bias Noise on the problem-solving ability in the Grid World scenario. Even at the lowest introduced probability of 0.05, there were instances where the terminal state and the goal were not reached. This highlights the severe impact of this noise source. As the noise probability increased to 0.15 and beyond, the number of times the optimal deterministic policies failed to achieve the goal increased, reinforcing the significance of the reward deviation observed in Figures 4 and 11.

Lastly, we analyzed the difference in Path Length and Euclidean Distance, as depicted in Figures 12 and 13, respectively. These results indicate that when the recovered reward was able to successfully solve the problem and reach the terminal state, the path used resembled the optimal path, with some difference in the states visited when the noise reached its peak at 0.4. An average Euclidean distance of 1.4 is not significant but suggests that a slightly different route was used to reach the terminal state.

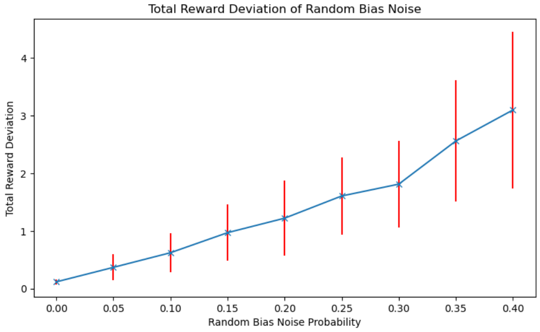

Figure 4: Total reward deviation of recovered rewards with Random Bias Noise compared to optimal recovery, with standard deviation for 100 iterations.

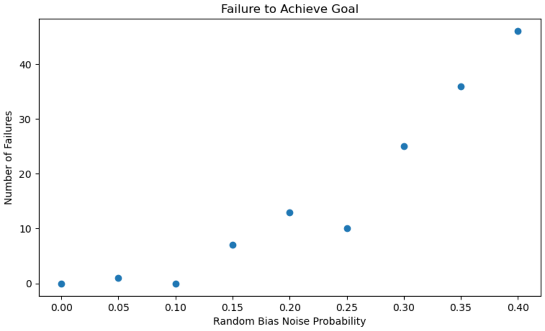

Figure 5: Frequency of noisy recovered rewards with Random Bias Noise Failing to reach terminal state with deterministic optimal policy, for 100 iterations.

5.3 Sparse Noise

After analyzing the graphs generated for demonstrations affected by Sparse Noise, we observed a clear linear upward trend in the reward deviation, as depicted in Figures 6 and 14. Interestingly, even at the highest influence factor, the percentage of deviation is not excessively large, hovering just above 60%. Additionally, considering the standard deviation among the 100 iterations, represented by the red lines in Figure 6, we found that the deviations between iterations remained relatively insignificant, even at high noise probabilities. This indicates that the total reward deviation metric effectively captures the extent of the deviations.

Of particular interest is the fact that for every influence factor, the terminal state is consistently reached, as illustrated in Figure 7. This implies that despite the presence of reward deviation, the recovered reward still leads to successful solutions to the Grid World problem.

Finally, examining Figures 15 and 16, which showcase the Path Length and Euclidean Distance from the optimal path, respectively, we can conclude that the paths generated from the noisy recovered reward are nearly identical to the optimal path, except when the influence factor of Sparse Noise is relatively high (e.g., 0.8).

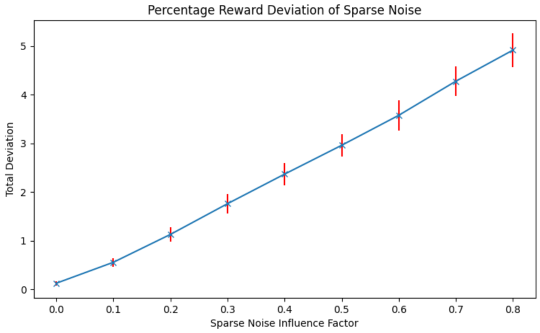

Figure 6: Total reward deviation of recovered rewards with Sparse Noise compared to optimal recovery, with standard deviation for 100 iterations.

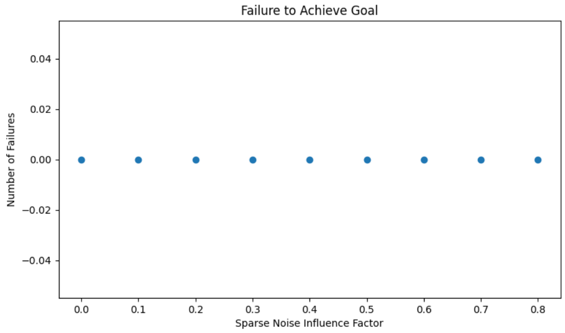

Figure 7: Frequency of noisy recovered rewards with Sparse Noise failing to reach terminal state with deterministic optimal policy, for 100 iterations.

6 Discussion

After conducting a thorough analysis of the results for each type of noise, several noteworthy conclusions can be drawn and further discussion can be undertaken.

Let us begin with Random Events Noise, where it is evident that MaxEnt IRL shows resilience and is not significantly impacted by noise probabilities below 0.7. Even when higher probabilities are introduced, only 17 iterations out of the total failed to reach the terminal state, indicating that the effect is not particularly detrimental. Consequently, based on our findings, it can be asserted that a certain level of randomness (below 70% per action) in expert demonstrations is acceptable without compromising the retrieval of a representative reward function and the attainment of the terminal state in a Markov Decision Process (MDP). This finding holds significant implications for MaxEnt IRL, as disturbances and random behavior are often unavoidable to some extent due to human errors, sensor limitations, and interactions with other agents in the MDP.

In contrast, Random Bias Noise yields unsatisfactory results. Even at low probabilities, there remains a chance of not achieving the goal and reaching the terminal state, signifying poor performance of MaxEnt IRL in the presence of this type of noise. Although the recovered rewards do not deviate significantly from the optimal recovered reward, like with the other types of noise, they fail to produce the correct behavior required to solve the given MDP. Furthermore, the high standard deviation of the recovered rewards suggests considerable variation in rewards across different iterations for the same noise probabilities. This instability in reward recovery is to be expected since randomness is introduced in the policy, however, there was hope that the recovered rewards would still enable MDP solutions, which is not the case. Thus, it can be concluded that the presence of random bias in expert policies leads to trajectories that hinder MaxEnt IRL from recovering a reward that facilitates successful solutions of the MDP. These findings underscore the importance of considering and addressing biased suboptimal behaviors when utilizing MaxEnt IRL in practice.

Finally, MaxEnt IRL demonstrates remarkable resilience to Sparse Noise. Regardless of the proportions of anomalous trajectories introduced, MaxEnt IRL consistently recovers a reward with some deviation that solves the MDP in a similar manner to the optimal reward. Notably, the extent of reward deviation appears to have a linear relationship with the proportion of anomalous trajectories. Consequently, Sparse Noise does not significantly impede the reward recovery capability of MaxEnt IRL, and even having only 20% of expert trajectories being optimal is sufficient. This finding is particularly encouraging since Sparse Noise is prevalent in real-world scenarios, as discussed in [16]. However, it is important to interpret these results cautiously since the Sparse Noise we modeled assumes a constant error probability for the anomalous trajectories. Therefore, further research could be conducted to increase the error and obtain more comprehensive results for more confident conclusions.

In addition to considering reward deviation and goal achievement in our MDP, the quality of the generated path based on the recovered reward also warrants discussion. It is noteworthy that across all three types of noise, when reaching the terminal state of the MDP, the path taken closely resembles the optimal one. The only exception occurs when the highest introduced probability is utilized to generate noisy demonstrations, where Random Bias Noise exhibits a slightly different path to reach the goal.

7 Responsible Research

In this research, the collection of primary data or the involvement of human subjects was not conducted. Instead, the expert demonstrations were synthesized rather than gathered from actual human experts. Therefore, no approvals were necessary for this study. However, we have made efforts to ensure the reproducibility [14] of our research by providing commented code [1] and instructions for generating expert demonstrations and calculating meaningful metrics. The code can be utilized for future investigations focused on the topic of noisy demonstrations in IRL, enabling researchers to replicate and build upon our findings.

Furthermore, it is important to acknowledge the limitations inherent in this study. The main limitation stems from the use of constructed demonstrations instead of real data from human experts. While this approach eliminates ethical concerns, it introduces a potential limitation in terms of the representativeness and reliability of the results. Constructed demonstrations may not fully capture the noisy behaviors and decision-making processes of real human experts. As a result, the generalizability and practical applicability of the findings may be reduced.

Additionally, another limitation relates to the choice of the environment. The utilization of the 5x5 Grid World MDP as the experimental environment facilitated obtaining initial results. However, it is important to note that increasing the complexity or realism of the environment could potentially reveal additional behavioral issues that the constructed expert demonstrations may not fully account for. Therefore, it is important to interpret the findings within the context of

[1] https://gitlab.tudelft.nl/lcavalcantesie/rp_irl_human_behavior/-/tree/NoisyDemonstrations

the chosen environment and consider the potential impact of environment-specific factors.

By acknowledging these limitations, we provide a comprehensive assessment of the study's scope and potential implications. It is important for future research to address these limitations by incorporating real human expert data and considering more diverse and challenging environments. This would enhance the reliability, validity, and applicability of the findings, and contribute to the advancement of the field of study.

8 Conclusion and Future Work

This study aimed to investigate the influence of noisy demonstrations on the learning capability of Inverse Reinforcement Learning (IRL), specifically focusing on the recovery of rewards using the Maximum Entropy IRL (MaxEnt IRL) algorithm in the presence of Random Events Noise, Random Bias Noise, and Sparse Noise.

Our findings demonstrate the robustness of MaxEnt IRL in the face of Random Events Noise with probabilities below 0.7. Additionally, the algorithm proves resilient to Sparse Noise, accommodating up to 80% anomalous expert trajectories. However, it demonstrates instability and inadequacy in handling Random Bias Noise, even at low probabilities. Therefore, biased suboptimal behavior exhibited in the expert policy has a negative influence on the learning capability of MaxEnt IRL. These challenges in reward recovery highlight the need for caution and further investigation when applying MaxEnt IRL in real-world scenarios.

For future research, there are several potential avenues for improvement and expansion. Firstly, incorporating additional noise types found in expert demonstrations would provide further insights into the behavior of MaxEnt IRL. Furthermore, mixing noise types might wield interesting results, of noise canceling out or amplifying one another. Additionally, exploring alternative IRL algorithms such as AIRL or Nonlinear Inverse Reinforcement Learning with Gaussian Processes could behave differently with noisy demonstrations. Another suggestion is to introduce greater complexity to the environment by incorporating negative rewards and obstacles, thereby creating a more realistic setting to investigate the effects of noise on IRL.

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 1, New York, NY, USA, 2004. Association for Computing Machinery.

[2] Richard E. Bellman. *Dynamic Programming (reprint ed.)*. Princeton University Press, 2010.

[3] Jitong Chen, Yuxuan Wang, and DeLiang Wang. Noise perturbation for supervised speech separation. *Speech Communication*, 78:1–10, 2016.

[4] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

[5] Yang Gao, Huazhe Xu, Ji Lin, Fisher Yu, Sergey Levine, and Trevor Darrell. Reinforcement learning from imperfect demonstrations. *ArXiv*, abs/1802.05313, 2018.

[6] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *J. Artif. Int. Res.*, 4(1):237–285, May 1996.

[7] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

[8] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[9] Maximilian Luz. Maximum entropy and maximum causal entropy inverse reinforcement learning implementation in python. https://github.com/qzed/irl-maxent, 2019. Accessed: May, 2023.

[10] Kun-Peng Ning and Sheng-Jun Huang. Reinforcement learning with supervision from noisy demonstrations. *ArXiv*, abs/2006.07808, 2020.

[11] Stuart J. Russell. Learning agents for uncertain environments (extended abstract). In *COLT' 98*, 1998.

[12] Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2021.

[13] Rohin Shah, Noah Gundotra, Pieter Abbeel, and Anca Dragan. On the feasibility of learning, rather than assuming, human biases for reward inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5670–5679. PMLR, 09–15 Jun 2019.

[14] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, and et al. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016.

[15] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. Ibat: Detecting anomalous taxi trajectories from gps traces. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, page 99–108, New York, NY, USA, 2011. Association for Computing Machinery.

[16] Jiangchuan Zheng, Siyuan Liu, and Lionel M. Ni. Robust bayesian inverse reinforcement learning with sparse behavior noise. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), Jun. 2014.

[17] Jianjun Zheng and Akbar Siami Namin. A markov decision process to determine optimal policies in moving target. In *Proceedings of the 2018 ACM SIGSAC*

Conference on Computer and Communications Security, CCS '18, page 2321–2323, New York, NY, USA, 2018. Association for Computing Machinery.

[18] Brian Ziebart, Andrew Maas, J. Bagnell, and Anind Dey. Maximum entropy inverse reinforcement learning. pages 1433–1438, 01 2008.

A Appendix A

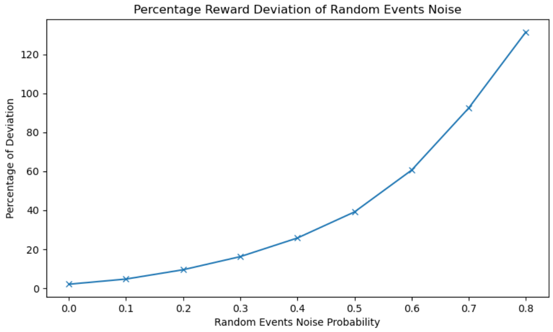

Figure 8: Percentage deviation of recovered rewards with Random Events Noise compared to optimal recovery, for 100 Iterations.

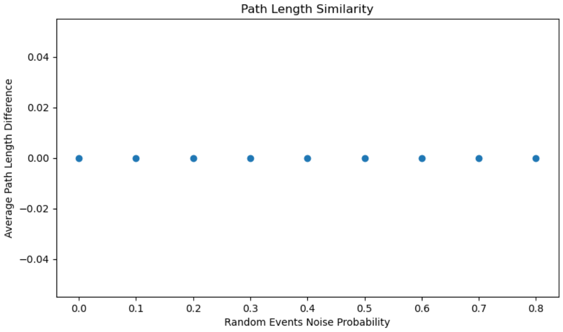

Figure 9: Average path length of deterministic optimal policy on noisy recovered reward with Random Events Noise for each noise probability, for 100 iterations.

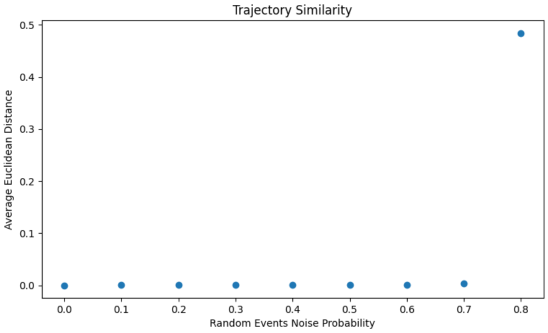

Figure 10: Average Euclidean distance between paths of noisy recovered reward with Random Events Noise and optimal recovered Reward, for 100 iterations.

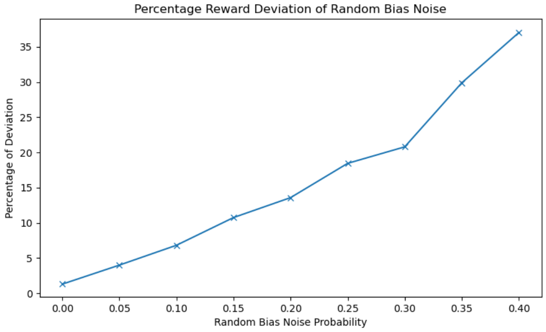

Figure 11: Percentage deviation of recovered rewards with Random Bias Noise compared to optimal recovery, for 100 iterations.

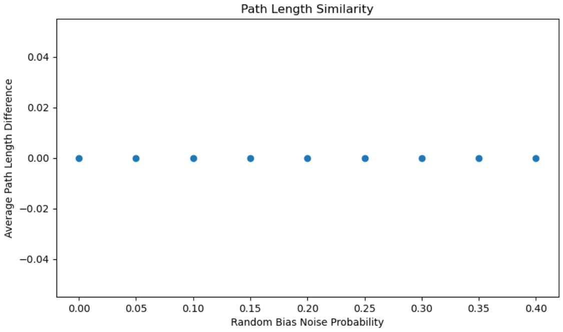

Figure 12: Average path length of deterministic optimal policy on noisy recovered reward with Random Bias Noise for each noise probability, for 100 iterations.

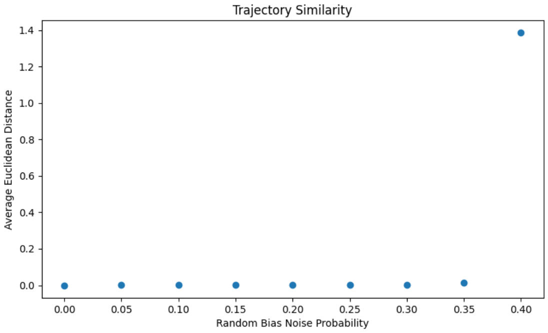

Figure 13: Average Euclidean distance between Paths of noisy recovered reward with Random Bias Noise and optimal recovered reward, for 100 iterations.

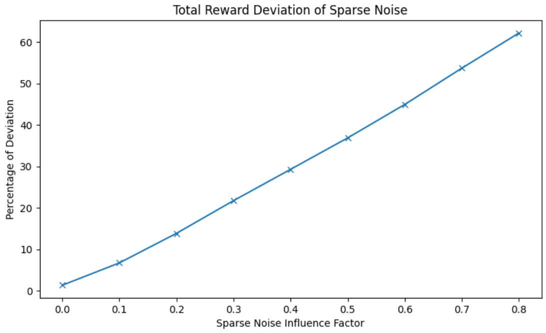

Figure 14: Percentage deviation of recovered rewards with Sparse Noise compared to optimal recovery, for 100 iterations.

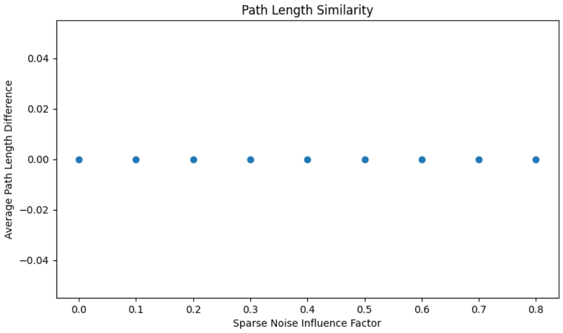

Figure 15: Average path length of deterministic optimal policy on noisy recovered reward with Sparse Noise for each noise probability, for 100 iterations.

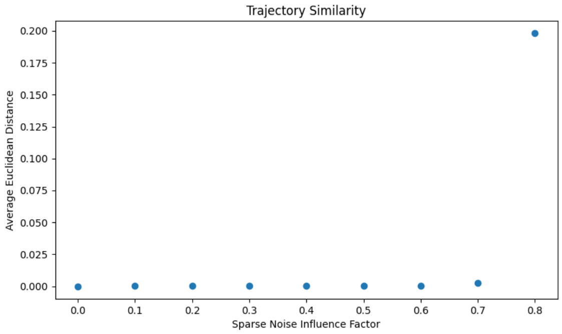

Figure 16: Average Euclidean distance between paths of noisy recovered reward with Sparse Noise and optimal recovered reward, for 100 iterations.