Investigating Inverse Reinforcement Learning from Human Behavior

Effect of Demonstrations with Temporal Biases on Learning Rewards using Inverse Reinforcement Learning

Mateja Zatezalo[1]

Supervisor(s): 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 25, 2023

Name of the student: Mateja Zatezalo
Final project course: CSE3000 Research Project
Thesis committee: Luciano Cavalcante Siebert, Angelo Caregnato Neto, Jana Weber

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Inverse Reinforcement Learning (IRL) is a machine learning technique used for learning rewards from the behavior of an expert agent. With complex agents, such as humans, the maximized reward may not be easily retrievable. This is because humans are prone to cognitive biases. Cognitive biases are a form of deviation from rationality that affects everyday human decision-making. Time inconsistent decision-making is a type of a temporal cognitive bias where planning of future actions may vary at different points of time. Existing research in this field explores using IRL algorithms in numerous real-life situations. However, few works examine the effects of temporal biases on the recovered reward function. Hence in this research, we propose a methodology to generate synthetic demonstrations that emulate human data with this bias. An existing method, Maximum Entropy IRL (MEIRL) algorithm is used to recover reward functions from expert models containing aforementioned biases and compare them to the performance of unbiased models. The demonstrations are in a form of Markov Decision Process (MDP), implemented in a GridWorld environment. Temporal biases will be implemented within the expert demonstrations as different types of agents that portray a specific behavior. Our findings show that all biases affect reward learning to a considerable extent, with that effect having different magnitudes depending on different comparisons.

Keywords: *Inverse Reinforcement Learning, Cognitive Bias, Time Inconsistency, Maximum Entropy, Markov Decision Process, Temporal*

1 Introduction

Machine learning (ML) is a branch of artificial intelligence that enables models to acquire knowledge and learn from the presented data. Demonstrations are often performed to teach specific policies or actions to the models. One technique within the field of ML is Inverse Reinforcement Learning (IRL) (Russell, 1998), which uses learning from expert demonstrations in order to learn the reward function of a Markov Decision Process (MDP) (Sigaud & Buffet, 2013). One goal of IRL is to understand human behavior from the demonstrations and use the knowledge to obtain the maximized reward function (Adams et al., 2022). The reward function is an important component of the learning task and, at times, can be hardly determined in certain applications.

Most important applications of IRL involve learning how to mimic actions of the expert, attempting to learn the reward function in order to improve interaction with other systems and learning about them (Adams et al., 2022). Hence, this technique has been used in various realms, ranging from interaction between autonomous cars and other cars and pedestrians (Sun et al., 2018), to controlling genetic regulatory networks (Imani & Braga-Neto, 2018). IRL maintains a high reputation because of its potential to improving capabilities of machines and aligning them to human values (Peschl et al., 2021).

Challenges still remain regarding further research of this field, including reducing computational complexity, and adapting to inevitable human systematic biases. These challenges have been explored in many works such as (Ziebart et al., 2008), (Fu et al., 2017) and (Levine et al., 2011), who created different IRL algorithms.
This is where the aim of our research is centered. The topic is as follows:

- To what extent can IRL learn rewards from demonstrations that contain some form of temporal cognitive bias?

Human behavior is imperfect and humans are naturally prone to having systematic biases (Kahneman et al., 1982). Alignment of these human characteristics to machines presents a large challenge in technology. Some of the characteristics are related to time.

Time inconsistency is a type of cognitive bias that refers to possessing some kind of inconsistent (varying) behavior through the specific time period (Frederick et al., 2002). Decision-making on which actions are taken at certain points of time can change and cause the inconsistency within humans (Wong, 2008).

On the other hand, time consistent decision-making implies that present plans match future plans. (Frederick et al., 2002). However, humans are prone to present bias (Chakraborty, 2021), which describes humans tend to prefer a smaller present reward over a bigger future reward. This behavior can also be classified as a time consistent bias as the rewards are consistently discounted over time, but it may affect future decision-making.

The goal of this research is to investigate the effects of these cognitive biases on the recovered rewards. This will be performed using an already developed algorithm, Maximum Entropy IRL (MEIRL) (Ziebart et al., 2008), since it addresses both imperfect behavior and sub-optimal expert demonstrations. We will evaluate and compare the model's performance of learning reward functions with the distinct temporal biases and compare them to models with unbiased models, in order to show the effect of the imperfect behavior.

This paper is organized as follows. Further information regarding MDPs, MEIRL and time inconsistency biases will be explained in Section 2, as well as previous related work. Methodology of the utilized environment (model) and the implementation of different time-related biases will be elaborated in Section 3. Furthermore, the results of experimenting with temporal biases will be shown in Section 4. Reflection and discussion of the results will be elaborated in Section 5, including improvements in the approach and missing instances. Section 6 proposes potential future work on this topic. Ethical aspects and the reproducibility of our methods will be discussed in Section 7. Finally, in section 8, the research will be concluded by summarizing the work.

2 Background and Related Work

This section presents the background information on Markov Decision Processes (MDPs), the Maximum Entropy IRL al-

gorithm (MEIRL), time inconsistency biases, and related work in this field.

2.1 Markov Decision Process

MDP (Sigaud & Buffet, 2013) provides a model for sequential decision-making under uncertainty. These stochastic models capture the dynamics of systems where the future state depends solely on the current state, independent of the past.

MDP is defined as a 5-tuple $\{S, A, P, \gamma, R\}$:

- S represents the set of states
- A represents the set of actions
- $P = P(s' \mid s, a)$ represents the probability that action a in state s goes to state s'
- γ represents the discount factor which quantifies the importance of short-term/long-term rewards
- R represents the reward function

MDPs enable agents to navigate uncertain environments by explicitly modeling transition probabilities and rewards. Policies map states to actions. Using policy π, there are different probabilities of choosing a particular action a in state s. The objective of the agent is to learn optimal policies that maximize long-term cumulative rewards within MDPs. The mathematical foundations of Markov processes empower researchers to develop advanced techniques for solving complex decision-making problems and exploring the boundaries of reinforcement learning theory.

2.2 Maximum Entropy IRL algorithm

Maximum Entropy IRL (MEIRL) (Ziebart et al., 2008) algorithm is used to train a set of agents. MEIRL addresses both imperfect behavior and sub-optimal expert demonstrations, hence it represents the best fit in our research, according to Ziebart et al. (2008), where this algorithm was firstly presented. The features define that the desired states of our environment, with the expectation that both the agent following our refined reward function, and the expert demonstrating near-optimal behavior, visit these states with equal frequency.

Matching the expected feature-visitation frequency with the best reward function proposes an ill-posed issue, since there are multiple rewards that match it (Ng et al., 1999). MEIRL proposes a solution to this problem by taking the solution with maximum entropy.

Entropy represents a measure of uncertainty. Feature-expectation matching has multiple solutions that satisfies our constraints. The only information which our solution contains for this problem are the feature-expectations we want to replicate. For this reason, choosing the solution with minimal information ensures lower probability of bias within that information. We achieve this by choosing the solution with maximum entropy.

2.3 Time inconsistency biases

As already stated, time inconsistency biases are types of cognitive biases. Cognitive bias (Kahneman et al., 1982) is a systematic process which makes people assess information and make decisions based on personal experience and knowledge. They represent the brain's shortcuts to navigating plans and decisions. Decision-making is often affected unconsciously, and the result can be positive or negative.

Time inconsistency biases lead people to make decisions that change over a period of time. Human characteristics which can explain this bias, according to Ainslie and George (2001), include pre-commitment and temptation, among others. A well-known example of this bias in human behavior is *hyperbolic discounting* (Evans et al., 2016).

Hyperbolic discounting holds that individuals exhibit a preference for immediate, smaller rewards over delayed, larger rewards. An example of this model is that if a person has the choice of receiving €100 now or €120 tomorrow, and they choose the smaller reward, but would rather choose €120 in e.g. 21 days than €100 in 20 days. The switch of the preference yields the inconsistency here.

This bias can have significant implications for self-control. For instance, someone might choose to eat unhealthy food now rather than adhere to a long-term healthy nutrition plan. People often succumb to immediate temptations rather than making choices that align with their long-term best interests.

Present or time consistent bias is explained in Section 3.3. Implementing and experimenting with these time-related biases, including exponential and hyperbolic discounting, are explained in Sections 3.3 and 3.4.

2.4 Related work

Time-related biases have been explored in many works, relating to multiple fields. Hyperbolic discounting and change of preference over time has been explored by Sozou (1998) and Zauberman et al. (2009). They have been explored in psychology, through modeling behavioral reinforcement learning (Sutton, Barto, et al., 1998). The computations relied on exponential discounting (Ainslie, 1992; Mazur, 1997), which was shown not to align with behavior of humans and animal, regarding time inconsistency. Hyperbolic discounting was later proposed as a more precise model (Frederick et al., 2002).

In computer science, the most researched factor regarding biases related to time are time discounting and temporal preferences (Green et al., 1994; Lattimore & Hutter, 2014). Discounting of rewards through time and the estimation of the discount factor in an IRL framework have been explored by Giwa and Lee (2021). While they focus on the estimation of the discount factor, we will show the behavior of agents with different types of discounting rewards, and the recovery of the reward using IRL.

As mentioned before, hyperbolic discounting is the most precise type of discounting rewards for humans, and has been explored in various works (Evans et al., 2016; Fedus et al., 2019; Nascimento, 2019). In these works, the bias in agents is implemented to emulate hyperbolic discounting, and the behavior of the agents are investigated. Fedus et al. (2019) show that a deep RL agent can emulate hyperbolic discounting using the Q-learning method (Watkins & Dayan, 1992).

However, to our knowledge, there is no research on how IRL algorithms, such as MEIRL, can recover reward functions from those demonstrations of biased models. The cre-

ation of synthetic demonstrations that emulate human data with temporal biases are inspired by multiple referenced works, and thoroughly elaborated in Sections 3 and 5. However, it is important to emphasize that we aim to expand the research by experimenting with MEIRL and explain how it affects the recovery of the reward function.

3 Methodology

This section will elaborate on the overall approach and techniques used in the research. Firstly, we will introduce the our approach for the utilized environment and how we use expert demonstrations, followed by the explanation of learning rewards using MEIRL. To continue, time-related biases in expert demonstrations will be explained backed by motivations of the implementation approach.

3.1 Environment and Research Approach

The environment we utilized was inspired by the work of (Ziebart et al., 2008), where they used this environment to introduce MEIRL. We believe this is the environment which successfully integrates expert demonstrations containing biases and the utilization of the MEIRL algorithm, while it also offers clear visualizations of policies, recovered rewards and agent behavior. The programming language we utilized is Python.

The environment consists of expert demonstrations implemented in a MDP by using a 6x6 state grid-world environment. This environment contains different rewards from which we obtain reward functions. These rewards are situated in the terminal states. Expert demonstrations are generated in form of different expert agents. Actions involve the possibility of the agent moving to all four adjacent states. However, in case the agent chooses an action that results beyond the edges of the grid, it will remain in the same state.

Figure 1 shows our grid-world environment, with three rewards the agent can obtain. The closest small reward (S) is colored dark blue, the medium reward (M) is colored lighter blue and the delayed big reward (B) is colored yellow.

Every expert agent represents characteristics of a specific temporal bias. Expert agents will be trained through value iteration (Poole & Mackworth, 2010) and adapted to include temporal biases. These agents represent the use of synthetic data which replaces human demonstrations. Similar agents have been used in various demonstrations within the field of artificial intelligence. (Peschl et al., 2021)

The stochastic policy is created given the value derived from value iteration, which describes a probability distribution of selecting an action depending on the state. This policy is then used to generate trajectories of the agent, forming the expert demonstrations. In this environment, it is important to mention that the decision-making (choosing of actions) possesses a 20% chance of choosing a random action, which incorporates stochasticity and adds variability to the agent's actions.

After we generate trajectories using the policy, we use MEIRL to recover (learn) rewards. The algorithm generates new trajectories based on the trajectories of the expert agents, and determines the value of the recovered reward.

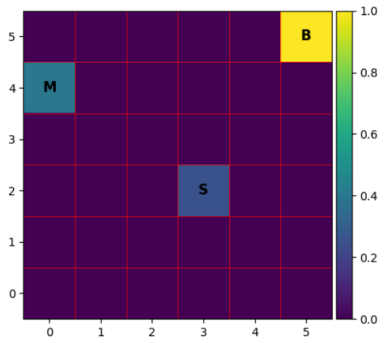

Figure 1: Original reward function

In our experiments, we will implement four types of expert agents: unbiased(optimal), time consistent (present) biased, and two time inconsistent biased agents. The agents will be elaborated in the following subsections.

3.2 Unbiased Agent

The unbiased agent represents the unbiased or optimal behavior. Its policy is optimal, which means the agent always chooses actions that lead to biggest rewards. The discount factor value, which decreases future rewards, should ideally be 1 for this agent, but is set to 0.995 in order to achieve convergence in value iteration. This setting emulates the lack of bias in this agent and ensure that the decision-making maintains the same over time.

3.3 Agent with Time Consistent (Present) Bias

The discount factor has an important role in our experiments as it represents how much rewards hold value in a certain time step t. It participates in generating the policy π in value iteration, while converging to calculate optimal state-action probabilities. The choices humans make are related to the value γ of the discount factor. If the value is higher, $0.5 < \gamma < 1$, the decision-maker is more likely to choose larger rewards and maintain their initial plans. On the other hand, when the value is lower, $0 < \gamma < 0.5$, the decision-maker is more 'forgetful' and more likely to choose actions with smaller rewards.

A way of discounting future rewards is exponential discounting, which is often used in demonstrations. However, Fedus et al. (2019) and Mazur (1997) argue that a single discount factor used in exponential discounting, does not reflect on the measured value preferences in humans with perfect accuracy. It does, though, represent a time consistent bias by decreasing the value of rewards by a constant factor. This fact motivated the further investigation of this bias in this research. Equation 2 describes exponential discount function:

$$D = \gamma^t \tag{1}$$

where γ represents the discount factor, and t represents the time-step. This discount function ensures that rewards decrease exponentially as the time goes, which is how present bias is defined in (Chakraborty, 2021). The agent knows this and stays consistent with its actions, hence this bias is considered time consistent with regards to planning future actions.

Agent implementation For this biased agent, the chosen value of distance factor is 0.9, and this means higher preference for the big, long-term reward. The value of the reward, relative to the state, is described in Equation 2. Value iteration is adapted to correctly calculate values and feature exponential discounting.

$$V_s = \max_a \{R(s) + \gamma * p(s, s', a)V_{s'}\} \tag{2}$$

where γ is the discount factor, and the action *a* and state *s'* are chosen from the maximum value from the state-action probability matrix. The values of all rewards will exponentially decrease.

3.4 Agents with Time Inconsistent Biases

From Section 2.3, we concluded that the agents with time inconsistency bias have a different discount function. Since humans discount their future rewards according to a hyperbolic curve (Fedus et al., 2019), we use a hyperbolic discount function described in Equation 1:

$$D = \frac{1}{1 + kd} \tag{3}$$

where *k* is the discount factor, and *d* presents a delay of time which both control how we discount future rewards. This discount function is considered time inconsistent, since there is a delay, that represents the time in the future where decisions of the agent can be changed.

Time inconsistency bias in agents are implemented in forms of a naive and a sophisticated agent. (O'Donoghue & Rabin, 1999). These agents represent different human behaviors related to time inconsistency. Naive agent represents temptation, and sophisticated agent represents precommitment (Ainslie & George, 2001). Agents differ in the way how they model their future actions.

- Naive: Models its future self with the same actions and values it has now. When it makes a decision for itself at time *t* + *d*, it discounts immediate rewards at 1 / (1 + *kd)*.
- Sophisticated: Knows all its future actions and models its planning accordingly. It makes the decision correctly at rate 1 / (1 + 0).

The existence of time inconsistency within these agents can be explained with an example. The objective of both agents is to obtain the maximum reward, the big reward (B).

The Naive agent moves along the shortest path. As it reaches the medium reward (M), the discounted reward value becomes larger than the delayed value for the bigger reward (B), and the agent cannot resist temptation, continuing to choose to obtain the medium reward. Along the path and with certain passed time, the plan of action changed, which displays time inconsistency in decision-making.

The Sophisticated agent has the same preference of reward as the Naive agent. However, it knows its future plans, so it chooses to go along the possibly longer route, to avoid temptation from the medium reward (M). On the longer route, it still might opt to obtain a smaller reward over the big reward if the difference in discount factor values is not significant. Otherwise, the initial plan stays, it reaches the big reward (B), which displays the characteristic of pre-commitment. In this way, the Sophisticated agent behaves similarly to the agent with time consistent bias, but the discounting of rewards is different.

Implementation of agents The implementation of these agents is as follows. The decision-making of the agent is dependent of three components: state *s*, action *a* and delay *d*, instead of the state-action decision-making in value iteration. Calculation of the value of the reward is inspired by (Evans et al., 2017), but is adapted to be similar to value iteration, as it best fits to our setting. The value of the reward *V* is consisted of the current reward *R* and the future reward. The current reward is discounted as described in Equation 3. The final reward is computed as described in Equation 4:

$$V_s = \max_{a,d} \left\{ \frac{1}{1 + kd} R(s, a) + p(s, s', a)V_{s'} \right\} \tag{4}$$

Where *s'* is the next state and is determined by the maximum value from the state-action probability matrix. The action *a* of the agents are determined by the maximum value of probability, according to the delay *d* for the Naive agent and the delay $d = 0$ for the Sophisticated agent. In this manner, the Naive agent computes the action it would take in state *s'* given that reward is obtained as with a delay *d*. The Sophisticated agent computes the action that will successfully happen, with the delay of 0.

The results obtained with these experiments and metrics used to evaluate the performance of learning rewards are elaborated in Section 4, while the further discussion on the implementation and the reflection on results can be found in Section 5.

4 Results

4.1 Agent with Time Consistent Bias

With the implementation of exponential discounting that represent the time consistent bias, we will display some results.

Figure 2 shows the behavior of the unbiased agent and its actions on the grid. The recovered reward using MEIRL is shown in Figure 3. These results can be compared to the behavior and recovered reward of the biased agent, where the reward is discounted exponentially with a discount factor of 0.9, shown respectively in Figure 4 and Figure 5. The value of the discount factor for the unbiased agent was set to 0.995, as explained in Section 3.2.

From Figure 2, we can clearly see that the policy of the unbiased expert agent leads it to claim the big reward (B) in most cases. With the recovered reward shown in Figure 3, we see MEIRL successfully learns this reward and that the big reward is the highest recovered among the terminal reward states.

In Figure 4, we see that the optimal policy of the biased agent changes by a certain margin comparing to the policy of the unbiased agent. However, the recovered reward did result in a solid replication of behavior. Hence, it is observable that MEIRL can learn rewards from demonstrations with this bias where the reward is decreased by exponential discounting.

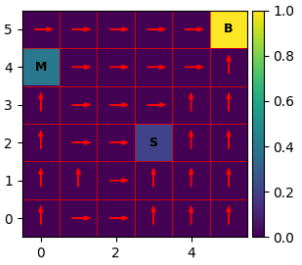

Figure 2: Optimal policy of unbiased agent

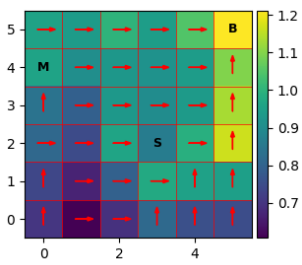

Figure 3: Recovered reward of unbiased agent

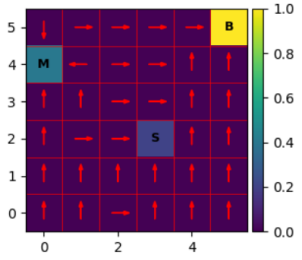

Figure 4: Optimal policy of biased agent

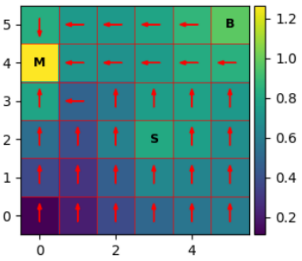

Figure 5: Recovered reward of biased agent

These initial results lead us to experimenting with different values of the discount factor. Distribution of obtained rewards with generated trajectories further shows the behavior of the biased agent. When started experimenting with different values of the discount factor, we observed that with discount factors smaller than 0.6, the rewards get discounted quickly over time and the agent always chooses to obtain the closest reward, the medium one. As we observe from Figure 6, the trajectories mainly end up in the medium reward state, in around 97% of them. With larger values of the discount factor, the behavior begins to change. By raising the discount factor, the agent chooses the small and the big reward more often. With the discount factor of 0.9, the agent obtains the medium reward in 73% of trajectories, the small one in 18% of trajectories, and the big one in 9% of trajectories. As the discount factor gets closer to the value of 1, the agent tends to obtain the big reward in most cases, but not all of them due to stochasticity and the minimal discounting of rewards. The optimal agent gets the small reward in 38% of the trajectories, the medium one in 16%, and the big one in 46%.

At this point we examined the distribution of trajectories in relation with the recovered reward using MEIRL algorithm, to investigate the affect of bias on reward learning. The results revealed consistent distribution, where the medium reward exhibited the highest recovery rate, with an average of 75.7%. The small reward is recovered with an average of 24.3%, while the big reward is solely recovered only when the discount factor is higher than 0.95. Besides the rare deviation when the discount factor value is 0.85, this distribution pattern shows that learning rewards from demonstrations with the time consistent bias, is not significantly influenced

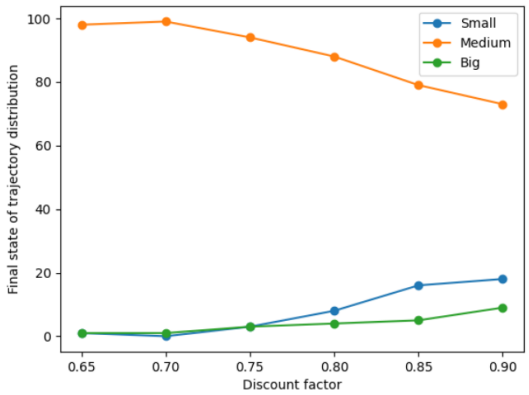

Figure 6: Trajectory distribution of the biased agent

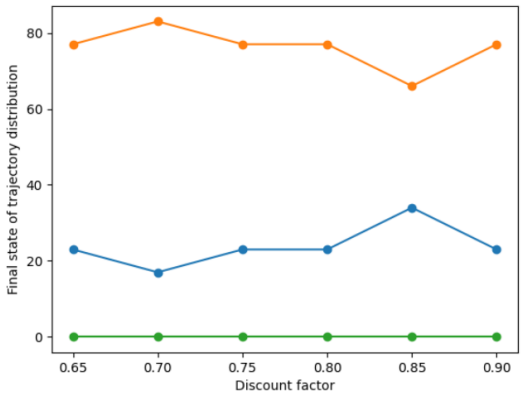

Figure 7: Trajectory distribution with MEIRL

by the value of the discount factor. Moreover, we can conclude by comparing results of the biased agent and recovering the reward, that MEIRL does not achieve the highest degree of success, mostly by differences of the recovery of the small reward.

Finally, we compared the similarity between trajectories generated from the biased agent with the recovered reward, and those trajectories generated from the optimal agent with the recovered reward. To calculate this similarity, we employed the Euclidean distance similarity to measure the closeness of the points in two trajectories. We calculated the similarity of trajectories in percentages, where points were considered close if their Euclidean distance similarity exceeded a predefined threshold of 0.3. The Euclidean distance similarity of two points is obtained as in Equation 5, where $d(p_1, p_2)$ is the Euclidean distance between points p_1 and p_2.

$$D = \frac{1}{1 + d(p_1, p_2)} \tag{5}$$

As the two trajectories were generated with the recovered reward using the optimal policy, we repeated the generation process 10 times and took the average value of the Euclidean distance similarity as a result.

Results of this analysis revealed an interesting pattern. The

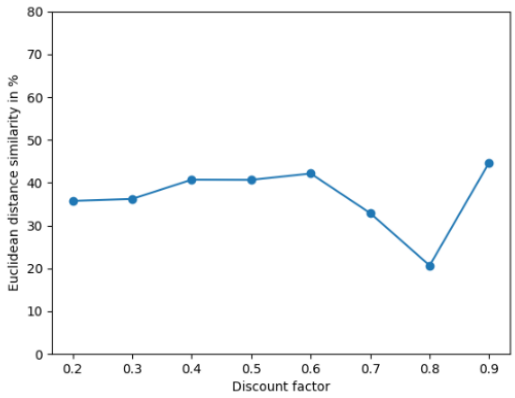

Figure 8: Similarity of trajectories generated from biased agent and the optimal agent with recovered reward

trajectories share 35.8% of similarity when the biased agent has a discount factor of 0.2. As the discount factor increased to 0.6, the similarity between the trajectories also gradually increased, reaching 42.2%. An unexpected drop in similarity was encountered for discount factor values of 0.7 and 0.8, where the similarity dropped to 20.7%. The reason for this could potentially be the same reason there was a deviation in the trajectory distribution of MEIRL for the similar discount factor value. With the discount factor of 0.9, the similarity peaked with a value of 44.7%.

These findings provide an insight into the affect of the bias on learning rewards, with the similarity of agent's trajectories. Observed results suggest that the discount factor has a nuanced impact on the similarity between generated trajectories.

4.2 Results of Time Inconsistent Agents

Optimal policy and recovered reward of the Naive agent are shown in Figure 9 and 10, respectively. In the same manner, the optimal policy and recovered reward of the Sophisticated agent are shown in Figure 11 and 12.

The figures display that the Naive agent imitates an inconsistent behavior, especially with the optimal policy with the recovered reward shown in Figure 10. The Sophisticated agent shows some inconsistent behavior, as it sometimes chooses the path to a small reward when reaching specific states, as well as the longest path to the big reward. However, we decided to focus on the performance of learning the rewards using the MEIRL algorithm.

In order to discover to what extent MEIRL learns rewards from demonstrations that contain time inconsistency biases, we again employed the Euclidean distance similarity metric to calculate the similarity between the trajectory of the optimal agent and the trajectory of the biased agents (Figure 13). Same threshold was applied as for the time consistent biased agent.

In our experiments, we used hyperbolic discounting to demonstrate time inconsistency, varying the discount factors from 0.02 to 0.2. Results showed that values larger than 0.2

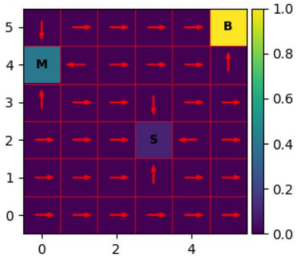

Figure 9: Optimal policy of Naive agent

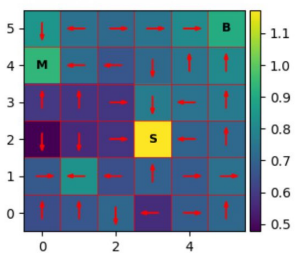

Figure 10: Recovered reward of Naive agent

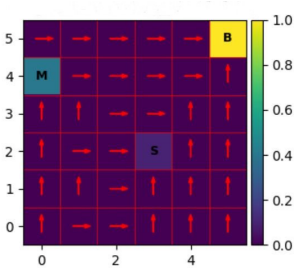

Figure 11: Optimal policy of Sophisticated agent

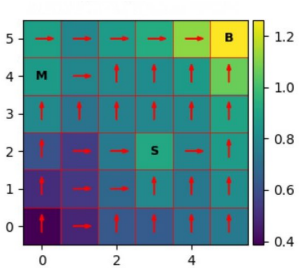

Figure 12: Recovered reward of Sophisticated agent

discounted rewards very rapidly, and therefore did not accurately represent naive and sophisticated behaviors of the agents.

The findings suggested that the similarity of trajectories was higher for the sophisticated agents than the naive ones, peaking at 42.6% for the discount factor of 0.02, while being reduced to 30.6% when the discount factor is increased to 0.2. Trajectories exhibited lower similarity for naive agents, peaking at 39% when the factor is 0.02, and declining to 24.7% for the discount factor value of 0.1.

These results showed that for both Naive and Sophisticated agents, MEIRL did not learn rewards to the perfect extent, comparing to the reward learning of the unbiased agent.

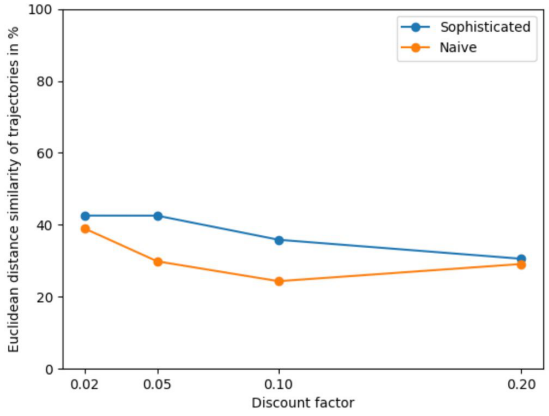

Figure 13: Similarity of trajectories generated from biased agent and the optimal agent with recovered reward

Though, it is noteworthy that the performance of MEIRL did not vary greatly between demonstrations with time consistent biases and time inconsistent biases.

5 Discussion

This section will discuss the final analysis on the obtained results mention some improvements and metrics we could have made to obtain different, and possibly better, results.

5.1 Result Analysis

Regarding the time consistent biased agent, the retrieved results were similar to what we expected. The biased agent showed the expected behavior, as it chose the medium reward in most trajectories. After applying MEIRL to the trajectories of the biased agent, we found that the rewards were learned in a similar fashion, with an exception. MEIRL recovered the small reward around 25% more than the big one consistently for all values of the discount factor, while the expert agent claimed big and small rewards at a similar amount of trajectories. The recovery of the medium reward was a bit lower than for the expert agent, but remained fairly consistent for all discount factors.

Regarding the comparison of performance of learning rewards between the biased agent and the unbiased optimal agent, from Figure 8 we concluded that the biased agent yielded worse recovery of rewards, at an average of 30.5% for the different discount factors. We claim it showed worse performance of learning rewards since from Figure 3, we found that the unbiased agent learns the rewards with the highest success rate.

On the other hand, regarding time inconsistent agents, we cannot conclude with absolute certainty that we received the best possible results and flawlessly shown the learning of rewards from the demonstrations generated by our versions of time inconsistent agents. The possible reasons are elaborated in the following subsection, while we will now reflect on the findings of our experiment.

Though the Sophisticated agent behaved in a similar fashion to the time consistent agent (as explained in Section 3.4), we obtained results between those two agents comparing to the optimal agent, that show a superiority of performance of the Sophisticated agent (around 10%). Hence, the way in which humans discount rewards (hyperbolic) performs slightly better than exponential discounting on learning the rewards.

The Naive agent showed inconsistent recovery of rewards comparing to the optimal agent, and with a lower success rate than the Sophisticated agent. This is caused presumably by the way the Naive agent is implemented, but also the inconsistent behavior it is supposed to imitate. On the other hand, we believe that MEIRL manages to recover the rewards to a substantial level. From Figure 10, we see that the most obtained reward is the small one, followed by the medium and big, which adheres to its characteristics according to its original description from O'Donoghue and Rabin (1999).

5.2 Improvements

To begin with, it is stated in Section 3.1 that the implementation of time consistent agents was performed in an environment inspired by the work of Ziebart et al. (2008). In the context of our second part of the research, time inconsistent agents, we built upon the work by Evans et al. (2017) to implement these specific types of agents. In their research, this is employed by the WebPPL tool (Goodman & Stuhlmüller, 2014), which offered them a suitable framework. After some experimenting, our objective was to integrate this approach into our original setting, and not fully abandon it. This decision was motivated by the fact that our original grid-world environment already adheres to the MEIRL algorithm, which is a crucial factor in our research. Furthermore, it also provides clear display of policies and recovered rewards, as explained in Section 3.1.

However, due to the time constraints of the research and the difference of implementation in the work of Evans et al. (2017), we could not manage to completely adapt it to our environment (setting). Nevertheless, we believe our implementation of time inconsistent agents (explained in Section 3.4), presents the correct adaptation to a significant extent, which is also shown by our findings, mainly from Figures 9 to 12. The key difference lies in the way our approach involves training agents with specific policies and utilizing those policies to generate trajectories, same as MEIRL algorithm uses the policy the calculate the final reward. In contrast, their implementation is not centered around IRL, nor it uses policies to generate trajectories or rewards. Instead, their implementation relies on a recursive decision rule in methods that calculate expected values (utilities in their work) and perform actions. While this approach certainly can be used to implement time inconsistent agents, it would be intriguing to compare our findings.

Finally, we are missing more results and different metrics to show the performance of reward recovery for time inconsistent agents. Due to the time constraints, the results were more focused on performance in form of the euclidean distance similarity of trajectories. In the potential future expansion of this research, more metrics should be incorporated, to help in discovery of more thorough findings. Future work involving this research is further discussed in the following section.

6 Future Work

This section discusses the possible next steps in the research of this topic.

The next step in this research topic would likely be the further adaptation of the grid-world environment, and using the same recursive approach in the tool WebPPL (Goodman & Stuhlmüller, 2014) to construct time inconsistent agents, as implemented in (Evans et al., 2017). As discussed in the previous section, this may result in getting more accurate results in the behavior of the agents, but also may improve the performance of learning the rewards.

If the research is to be continued using the same environment, it can be enhanced by incorporating walls in the grid, in the form of inaccessible states, which reduce the number of potential different trajectories of the agent. This would improve the supposed behavior of the agents, especially the time inconsistent ones, as they would construct more optimal

paths.

Due to the time constraints of this research, real-life data from humans could not have been collected. It would be interesting to compare real-life data to synthetic data used in this research, and compare the results of recovered rewards.

While we believe the Naive and Sophisticated agents correctly emulate human characteristics regarding time inconsistency, there is one more mentioned in (Ainslie & George, 2001), which is procrastination.

Procrastination means delaying the important task, by focusing on smaller, immediate tasks. This can also be replicated in the expert demonstrations, though the implementation is different to the implementation of agents in our MDP. The idea behind this agent is elaborated by Evans et al. (2017), where they use the Partially Observed MDP (POMDP) (Spaan, 2012). Learning rewards using IRL algorithms can then be investigated with demonstrations containing this bias.

We believe these advancements would certainly help expand the research in this field and determine the ability of IRL to learn rewards from agents with temporal biases.

7 Responsible Research

Ethical responsibility is a very important factor in every research. As already explained, no real-life data was used in this research, hence approval and consent was not needed in our case. The implementation of all agents (expert demonstrations) are inspired by previous works, as explained in Section 3. Maximum Entropy IRL (MEIRL) algorithm is a well-known, published algorithm. Changes and alterations in code were done to adhere to our MDP environment and the use of MEIRL, in order to obtain proper results and show the results of our research.

With this, all researchers can replicate the investigation in order to observe results or expand onto ours. This research paper will also be available online in TU Delft research repositories, so it will be available for all future researchers.

The limitations of this research include creating synthesized demonstrations, instead of using real-life data from humans. In addition, the adapted implementation of the Naive and Sophisticated agents may not replicate the behavior of time inconsistent agents to perfection. Unfortunately, with the use of the 6x6 grid-world MDP, the existing MEIRL algorithm and the form of displaying results in Section 4, the time inconsistency bias in agents could not be replicated in the same way as inspired from the work of Evans et al. (2016), as explained in Section 5. Therefore, this may hinder the further applicability of our findings.

8 Conclusion

As Inverse Reinforcement Learning continues to emerge as a valuable framework in the world of Artificial Intelligence, more research is conducted in order to expand its applications and enrich this field. Understanding human behavior from demonstrations in order to retrieve maximized rewards describes the aim of IRL algorithms. Humans tend to possess cognitive biases which hinders their decision-making and everyday life. One type of those cognitive biases include temporal or time-related biases, which affect people's planning over time.

The objective of this paper is to investigate the effect of expert demonstrations containing temporal biases on reward learning using Inverse Reinforcement Learning. From numerous types of temporal biases, we managed to investigate this effect by implementing three different types of biases in our expert demonstrations. which represent different human characteristics (present, pre-commitment and temptation). These biases were incorporated in the demonstrations in a form of agents (time consistent, sophisticated and naive, respectively). The existing algorithm, Maximum Entropy IRL algorithm, was utilized to recover (learn) rewards from the behavior of these agents. Our findings show that all biased agents have a substantial effect on learning rewards, especially when compared to learning rewards from an optimal agent. Moreover, we have also shown that IRL learns rewards from time consistent and sophisticated agents at a higher rate then the naive agent.

As we conducted this research, we have come across potential improvements and additional temporal biases which can be utilized to further expand this topic. It will be intriguing to compare our results to the results using different tools, as well as the effect of demonstrations with different temporal biases on reward learning.

References

Adams, S., Cody, T., & Beling, P. A. (2022). A survey of inverse reinforcement learning. *Artificial Intelligence Review*, *55*(6), 4307–4346.

Ainslie, G. (1992). *Picoeconomics: The strategic interaction of successive motivational states within the person*. Cambridge University Press.

Ainslie, G., & George, A. (2001). *Breakdown of will*. Cambridge University Press.

Chakraborty, A. (2021). Present bias. *Econometrica*, *89*(4), 1921–1961.

Evans, O., Stuhlmüller, A., & Goodman, N. (2016). Learning the preferences of ignorant, inconsistent agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1).

Evans, O., Stuhlmüller, A., Salvatier, J., & Filan, D. (2017). Modeling agents with probabilistic programs. *URL: http://agentmodels. org*.

Fedus, W., Gelada, C., Bengio, Y., Bellemare, M. G., & Larochelle, H. (2019). Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*.

Frederick, S., Loewenstein, G., & O'donoghue, T. (2002). Time discounting and time preference: A critical review. *Journal of economic literature*, *40*(2), 351–401.

Fu, J., Luo, K., & Levine, S. (2017). Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*.

Giwa, B. H., & Lee, C.-G. (2021). Estimation of discount factor in a model-based inverse reinforcement learning framework.

Goodman, N. D., & Stuhlmüller, A. (2014). The Design and Implementation of Probabilistic Programming Languages [Accessed: 2023-6-25].

Green, L., Fry, A. F., & Myerson, J. (1994). Discounting of delayed rewards: A life-span comparison. *Psychological science*, *5*(1), 33–36.

Imani, M., & Braga-Neto, U. M. (2018). Control of gene regulatory networks using bayesian inverse reinforcement learning. *IEEE/ACM transactions on computational biology and bioinformatics*, *16*(4), 1250–1261.

Kahneman, D., Slovic, P., & Tversky, A. (1982). *Judgment under uncertainty: Heuristics and biases*. Cambridge university press.

Lattimore, T., & Hutter, M. (2014). General time consistent discounting. *Theoretical Computer Science*, *519*, 140–154.

Levine, S., Popovic, Z., & Koltun, V. (2011). Nonlinear inverse reinforcement learning with gaussian processes. *Advances in neural information processing systems*, *24*.

Mazur, J. E. (1997). Choice, delay, probability, and conditioned reinforcement. *Animal Learning & Behavior*, *25*(2), 131–147.

Nascimento, J. C. d. (2019). Rational hyperbolic discounting. *arXiv preprint arXiv:1910.05209*.

Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *Icml*, *99*, 278–287.

O'Donoghue, T., & Rabin, M. (1999). Doing it now or later. *American economic review*, *89*(1), 103–124.

Peschl, M., Zgonnikov, A., Oliehoek, F. A., & Siebert, L. C. (2021). Moral: Aligning ai with human norms through multi-objective reinforced active learning. *arXiv preprint arXiv:2201.00012*.

Poole, D. L., & Mackworth, A. K. (2010). *Artificial intelligence: Foundations of computational agents*. Cambridge University Press.

Russell, S. (1998). Learning agents for uncertain environments. *Proceedings of the eleventh annual conference on Computational learning theory*, 101–103.

Sigaud, O., & Buffet, O. (2013). *Markov decision processes in artificial intelligence*. John Wiley & Sons.

Sozou, P. D. (1998). On hyperbolic discounting and uncertain hazard rates. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *265*(1409), 2015–2020.

Spaan, M. T. (2012). Partially observable markov decision processes. *Reinforcement learning: State-of-the-art*, 387–414.

Sun, L., Zhan, W., & Tomizuka, M. (2018). Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2111–2117.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning* (Vol. 135). MIT press Cambridge.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*, 279–292.

Wong, W.-K. (2008). How much time-inconsistency is there and does it matter? evidence on self-awareness, size, and effects. *Journal of Economic Behavior & Organization*, *68*(3-4), 645–656.

Zauberman, G., Kim, B. K., Malkoc, S. A., & Bettman, J. R. (2009). Discounting time and time discounting: Subjective time perception and intertemporal preferences. *Journal of Marketing Research*, *46*(4), 543–556.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. (2008). Maximum entropy inverse reinforcement learning. *8*, 1433–1438.