Open Problem: Reward Estimation in Inverse Bandit Problems

Michela Petriconi
University of Tübingen and Tübingen AI Center
michela.petriconi@uni-tuebingen.de

Claire Vernade
University of Tübingen
claire.vernade@uni-tuebingen.de

Abstract

Inverse Reinforcement Learning (IRL) is the problem of inferring a reward function from an observed optimal behavior in a Markov Decision Process (MDP) [12]. However, the idea of deriving a reward function from a *learning agent* has not been extensively explored. This paper examines this question within the Inverse Bandit context [6], which focuses on estimating the expected rewards of a multi-armed bandit (MAB) instance by observing the actions of a demonstrator algorithm. Despite some recent efforts to address this issue [6, 5], many questions remain open, and we aim to highlight some of them in this work.

1 Introduction

Inverse Reinforcement Learning (IRL) is the problem of inferring a reward function from an observed optimal behavior in a Markov Decision Process (MDP) [12], such as expert's trajectories (or demonstrations). Over the past decade, IRL has attracted much interest in the machine learning and control theory communities, and different methods have been proposed, e.g. the ones employing Bayesian inference [14] or the principle of *Maximum-entropy* [17].

IRL is an ill-posed problem: the reward function that explains the expert's behavior is not unique in general and it is possible to show that the *feasible* reward set contains infinitely many reward functions [10]. Consequently, research on reward identifiability [11, 3, 9, 16] has emphasized the importance of incorporating more exploration in the expert's trajectories.

Despite this, most of the IRL literature considers demonstrations generated by an *optimal* policy. Although some work has incorporated sub-optimal experts [13, 2], there has been little research on considering *learning experts*, i.e. whose policy evolves over time, as they collect data.

Some initial steps in this directions have been made on the *Inverse Bandit* problem, which was formally introduced by Guo et al. [6]. This concerns estimating the expected rewards of a multi-armed bandit (MAB) instance solely from observing the actions of a demonstrator algorithm. It is important to note that in MAB, the identifiability issue from optimal demonstrations is particularly acute. Suppose you can observe someone playing UCB [1] on a K-armed bandit problem. It is well known that, after some time, higher value arms should be played more often [8], making it relatively easy to identify which are the sub-optimal arms. However, can the sub-optimality gaps be recovered? This question is partially positively answered by Guo et al. [6], and the subsequent work by Guha et al. [5] in the stochastic linear bandits setting. Yet, there are still unresolved questions, and the purpose of this paper is to raise some of them.

Guo et al. [6] propose a procedure that estimates rewards for each sub-optimal arm and they provide theoretical guarantees on the estimation error. However, this procedure requires to know the exact algorithm that generated the demonstration. Sadly, for technical reasons, the latter is not a widely used algorithm but a modified version of standard UCB that results in a higher regret.
Question 1: *Can we define a reward estimation procedure that yields good results even with*

38th Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

trajectories generated by commonly used algorithms, e.g. vanilla UCB?
Question 2: *Is it possible to design a procedure that estimates rewards without knowing the algorithm that generated the trajectories?*
Other natural questions concern the sample complexity of the problem that we will formalize in Section 4.2 together with preliminary results. Ultimately, our goal is to highlight the remaining challenges and stimulate interest within the community.

2 Background and notation

Let us recall the standard notation for the MAB problem [8]. We can define a *stochastic bandit* as a collection of distributions $\nu = (P_i : i \in [K])$, where $[K] := \{1, 2, \dots, K\}$ is the set of available actions (or bandit arms). The learner and the environment interact over a *horizon* of T rounds. At each round $t \in \{1, \dots, T\}$ the learner chooses an action $A_t \in [K]$, then the environment samples a reward $X_t \in \mathbb{R}$ from the distribution P_i and reveals it to the learner. For each arm $i \in [K]$ we denote $\mu_i := \mathbb{E}_{X \sim \nu_i}[X]$ its expected reward. Define $T_i(t) := \sum_{s=1}^{t} \mathbb{1}(A_s = i)$, as the number of times arm i has been pulled by the learner. Assume that there is a unique best arm or *optimal arm* that can be identified as $i^* := \textit{arg max}_{i \in [K]} \mu_i$. Therefore, we refer to the remaining arms, i.e. $i \in [K] \backslash \{i^*\}$, as sub-optimal. For each sub-optimal arm the *sub-optimality gap* is defined as $\Delta_i := \mu^* - \mu_i$, where $\mu^* := \mu_{i^*}$ is the expected reward of the optimal arm. The learner's goal is to minimize the *regret*, i.e. the difference between the best possible reward she could obtain, if she knew which is the optimal arm, and the actual accumulated reward. We say that a *no-regret algorithm* is one whose regret scales sub-linearly with the horizon T. One of the most common algorithm that obtain sub-linear regret is *Upper-Confidence-Bound* (UCB) [1] or its *anytime* version *Asymptotically Optimal UCB* [7].

2.1 Problem statement

Inverse Bandits [6] is the problem of estimating the rewards of a MAB instance from observing only the actions of a demonstrator algorithm. More formally, we can define a trajectory (or demonstration) as a sequence of actions $\{A_t\}_{t=1}^{T}$, i.e. the sequence of arms pulled by the algorithm. Therefore, a reward estimation procedure is a mapping from demonstrations to mean estimates $\hat{\mu}_i$ for each arm $i \in [K]$. The goal is to minimize the *expected estimation error* for each arm i, i.e. $\mathbb{E}[|\hat{\mu}_i - \mu_i|]$. We will assume that the procedure knows the value of μ^* but not the index i^* as in [6]. Note that this assumption does not change the problem. In fact, here the goal is the *estimation* of reward values as opposed to identifying the best arm, that is a relatively straightforward problem instead.

3 State of the art

Guo et al. [6] studied Inverse Bandits and showed, under certain assumptions (see Theorem 1 in [6]), the following information-theoretic lower bound on the estimation error given *any* reward estimation procedure:

$$\mathbb{E}[|\hat{\mu}_i - \mu_i|] \geq \frac{1}{16} \left(\frac{1}{\sqrt{\mathbb{E}[T_i(T)]}} \wedge 1 \right). \tag{1}$$

It shows that the procedure's efficacy in estimating μ_i is fundamentally limited by $\sqrt{\mathbb{E}[T_i(T)]}$, highlighting the necessity of exploration, as observed in the IRL case.

However, not every demonstrator algorithm, that perhaps includes "enough" exploration, allow us to match this lower bound. For instance, consider the Explore-Then-Commit algorithm [15]. It suffers from a regret of $O(T^{2/3})$, thus constituting a no-regret algorithm. Despite this, it is impossible to perform reward estimation by only observing trajectories generated by this algorithm. In practice, it explores each arm a fixed number of times and then exploits the estimated best arm. Therefore, by observing a trajectory, we can only identify the best arm, but it provides no information about the sub-optimal ones. Hence, we can say that reward estimation is only possible when the learning algorithm exhibits some type of instance-dependent behaviour. For this reason, the authors focused on two families of algorithms with this characteristic: successive-arm-elimination (SAE) [4] and UCB [1]. Perhaps surprisingly, they use a slightly modified version of the algorithms to control exploration. From now on, we will focus on the UCB family of algorithms, although similar conclusions could be drawn for the SAE family. The version of UCB employed in Guo et al. [6] is obtained by defining

the confidence intervals for each arm i at time $t \in \{1, \dots, T\}$ as $C_{i,t} := \sqrt{\frac{2(T^\alpha - 1)}{\alpha T_i(t)}}$ for $\alpha \in [0, 1)$. This algorithm suffers from a regret of $O(T^\alpha)$. Note that for $\alpha \to 0$ the typical choice of confidence intervals (vanilla UCB) is recovered. The authors justify this by stating that a higher value of α inflates the confidence intervals, allowing for greater exploration. Although result (1) demonstrates the necessity of exploration, in practice it is more common to observe trajectories generated by vanilla UCB rather than this modified version.

3.1 The reward estimator

The main contribution of the paper [6] is the design of a *reward estimation procedure* (see Procedure 2 in [6]), along with providing theoretical guarantees on the expected estimation error. The procedure takes as input *one* trajectory $\{A_1, \dots, A_T\}$, a scalar μ^* and the $\alpha \in [0, 1)$ employed by the demonstrator algorithm. It estimates the best arm as the most pulled one, i.e. $\hat{i} = \arg\max_i T_i(T)$, then it computes the *switching round* for every sub-optimal arm as follows:

$$\tau_i := \max\{t : A_t = i \quad \text{and} \quad A_{t'} = \hat{i} \quad \text{for some} \quad t' > t\}. \tag{2}$$

Finally, it estimates the rewards as:

$$\hat{\mu}_i := \mu^* - (C_{i,\tau_i} - C_{\hat{i},\tau_i}), \quad \forall i \in [K] \backslash \{i^*\}. \tag{3}$$

Note that this procedure requires knowledge of the demonstrator algorithm, particularly the value of α used. Under certain assumptions (see Theorem 2 in [6]) the authors prove that it exists an universal constant C such that for any sub-optimal arm the procedure satisfies:

$$\mathbb{E}[|\hat{\mu}_i - \mu_i|] \leq C \sqrt{\frac{\log(\mathbb{E}[T_i(T)]\sqrt{K})}{\mathbb{E}[T_i(T)]}}. \tag{4}$$

Furthermore, $\mathbb{E}[T_i(T)] \geq c \frac{T^\alpha - 1}{\alpha \Delta_i^2}$, for some other universal constant $c > 0$. While the theorem indicates that a higher value of $T_i(T)$ leads to a smaller estimation error, it is important to acknowledge that increased exploration of sub-optimal arms results in higher regret, a consequence that is undesirable in practical applications.

4 Open problems

While an initial step has been taken in addressing the Inverse Bandit problem, numerous questions remain unanswered. Here we will highlight some of them.

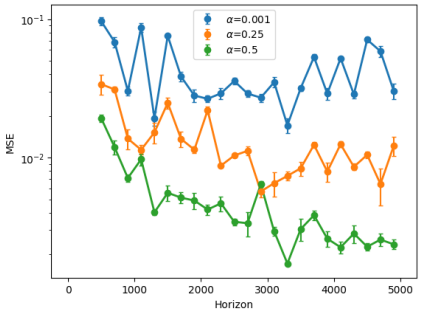
4.1 Demonstrator algorithm

The first open question concerns the algorithm employed to generate the demonstrations. In the modified version of UCB used in [6], increasing the exploration rate (i.e., using a larger α) reduces the estimation error (Figure 1a) obtained by the reward estimation procedure. However, this also increases the regret for the demonstrator, as shown in Figure 1b. The estimation error here is measured as the Mean Squared Error (MSE) $\mathbb{E}[|\mu_i - \hat{\mu}_i|^2]$ for each sub-optimal arm. In practice, this modified version of UCB is not as commonly used as the vanilla UCB, and the results from the reward estimation procedure for α values closer to those in vanilla UCB are not as good as those for larger values of α (see Figure 1a). Therefore, it would be useful to explore the following questions:

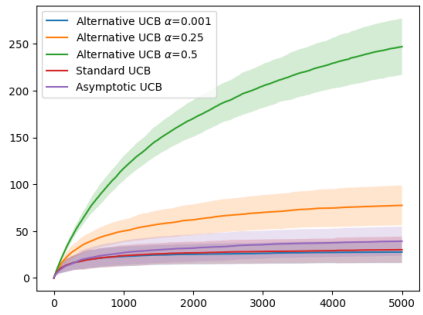
Question 1: *Is it possible to implement a reward estimation procedure (or modify the existing one) that achieves good results for the vanilla UCB, given that it is a commonly used algorithm for generating trajectories?*
Question 2: *Would it be possible to develop a reward estimation procedure that does not require prior knowledge of the generating algorithm but is adaptable to a set of different algorithms?*

4.2 Sample complexity

Both Guo et al. [6] and Guha et al. [5] consider the process of estimating rewards by observing *only a single* trajectory. Although they achieved strong theoretical bounds on the estimation error, they



(a) Reward estimation (sub-optimal arm) for various α. Results for 50 runs of simulations.



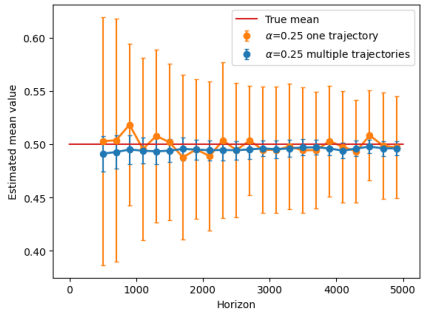
(b) Regret curves for different implementations of UCB. Results for 100 runs of simulations.

Figure 1: In both cases we simulate a $K = 2$ armed bandit instance with Gaussian rewards distribution $X \sim \mathcal{N}(\mu_i, 1)$, where $\mu_1 = 1, \mu_2 = 0.5$. Algorithms with smaller errors (left plot) result in much higher regret (right plot).

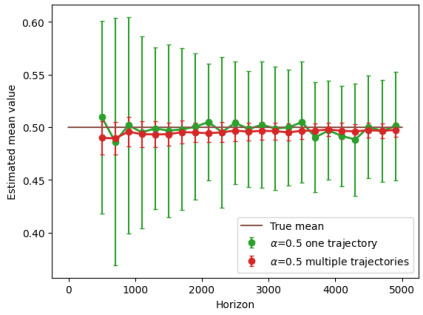
also observed that a smaller error is linked with higher regret (see Corollary 1 in [6]).
Question 3: *Can we mitigate this issue by incorporating additional trajectories? If so, what is the trade-off between regret, estimation error and number of demonstrations?*

We designed a straightforward procedure to include more trajectories: consider n trajectories $\mathcal{D} = \{\tau_1, \ldots, \tau_n\}$, then for each trajectory estimate the best arm as $\hat{i}_j = \arg\max_i \{T_i(T)\}_{\tau_j}$. Set $\hat{i}$ as the arm that is considered the best in the majority of these trajectories. Next, remove from $\mathcal{D}$ the trajectories for which $\hat{i}_j \neq \hat{i}$. Using the same method as Guo et al. [6] and considering $\hat{i}$ as the optimal arm, estimate the rewards for each remaining trajectory. Finally, average these results to obtain $\hat{\mu}_i$ for all $i \in [K]$. In Figure 2 we show the results of this procedure compared to those obtained by observing a single trajectory. Note that increasing the number of trajectories leads to a better estimation of the arm means, even within a shorter horizon, significantly reducing the variance.

Finally, we wonder how to efficiently collect data by designing an algorithm that queries only the necessary number of trajectories for accurate reward estimation, in the spirit of Best Arm Identification [4] and Active Exploration in IRL [9].
Question 4: *Is it possible to design an algorithm that adaptively queries trajectories until the estimation error is below a given threshold ($\epsilon > 0$)?*



(a) Mean value estimation for demonstrator algorithm with $\alpha = 0.25$.



(b) Mean value estimation for demonstrator algorithm with $\alpha = 0.5$.

Figure 2: In both cases we simulate a $K = 2$ armed bandit instance with Gaussian rewards distribution $X \sim \mathcal{N}(\mu_i, 1)$, where $\mu_1 = 1, \mu_2 = 0.5$. The results are obtained over 100 simulations for different values of α. The comparison is between the reward estimation procedure with 1 trajectory and 50 trajectories for the sub-optimal arm.

Acknowledgments and Disclosure of Funding

We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for their support. M. Petriconi and C. Vernade are supported by the German Research Foundation (DFG)

under both the project 468806714 of the Emmy Noether Programme and under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645.

References

[1] P. AUER, N. CESA-BIANCHI, and P. FISCHER. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

[2] D. Brown, W. Goo, P. Nagarajan, and S. Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 783–792. PMLR, 09–15 Jun 2019.

[3] H. Cao, S. Cohen, and L. Szpruch. Identifiability in inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12362–12373, 2021.

[4] E. Even-Dar, S. Mannor, Y. Mansour, and S. Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

[5] E. Guha, J. James, K. Acharya, V. Muthukumar, and A. Pananjady. One shot inverse reinforcement learning for stochastic linear bandits. *Proceedings of Machine Learning Research*, 1: 34.

[6] W. Guo, K. K. Agrawal, A. Grover, V. K. Muthukumar, and A. Pananjady. Learning from an exploring demonstrator: Optimal reward estimation for bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

[7] J. Honda and A. Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *COLT*, pages 67–79. Citeseer, 2010.

[8] T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[9] D. Lindner, A. Krause, and G. Ramponi. Active exploration for inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5843–5853, 2022.

[10] A. M. Metelli, G. Ramponi, A. Concetti, and M. Restelli. Provably efficient learning of transferable rewards. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7665–7676. PMLR, 18–24 Jul 2021.

[11] A. M. Metelli, F. Lazzati, and M. Restelli. Towards theoretical understanding of inverse reinforcement learning. In *International Conference on Machine Learning*, pages 24555–24591. PMLR, 2023.

[12] A. Ng and S. Russell. Algorithms for inverse reinforcement learning. *ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning*, 05 2000.

[13] R. Poiani, G. Curti, A. M. Metelli, and M. Restelli. Inverse reinforcement learning with sub-optimal experts. *arXiv preprint arXiv:2401.03857*, 2024.

[14] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

[15] H. Robbins. Some aspects of the sequential design of experiments. 1952.

[16] P. Rolland, L. Viano, N. Schürhoff, B. Nikolov, and V. Cevher. Identifiability and generalizability from multiple experts in inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 35:550–564, 2022.

[17] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.