INVERSE REINFORCEMENT LEARNING FOR HELPER-AI DESIGN

April-August 2017

Raphaël Duroselle

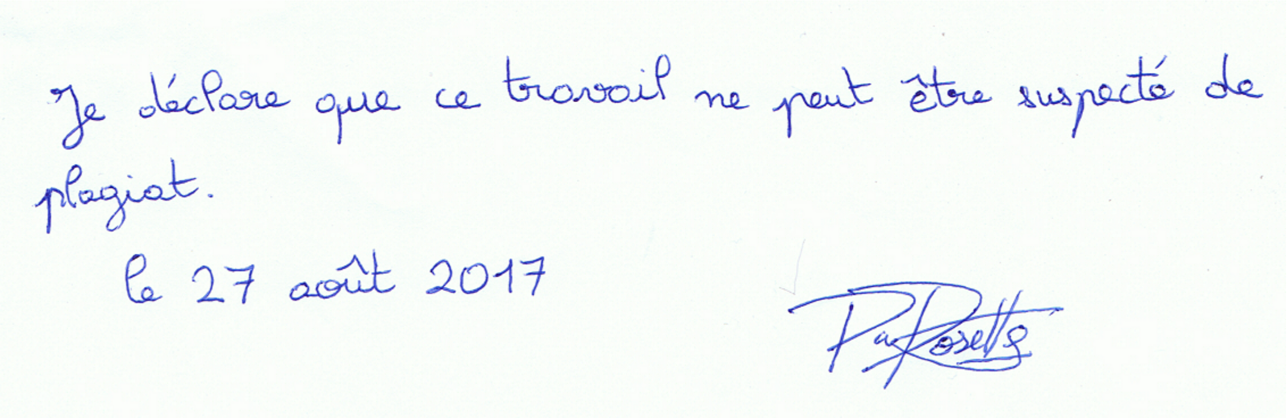
Advisor: Christos Dimitrakakis

HARVARD
School of Engineering and Applied Sciences

DÉCLARATION D'INTÉGRITÉ RELATIVE AU PLAGIAT

Je soussigné Raphaël Duroselle certifie sur l'honneur:

1. Que les résultats décrits dans ce rapport sont l'aboutissement de mon travail.
2. Que je suis l'auteur de ce rapport.
3. Que je n'ai pas utilisé des sources ou résultats tiers sans clairement les citer et les référencer selon les règles bibliographiques préconisées.



Je déclare que ce travail ne peut être suspecté de plagiat.

le 27 août 2017

Abstract

We consider a two-player game played by two cooperative agents. They disagree on the underlying Markovian model of the world and thus achieve a suboptimal equilibrium. The Helper-AI team at Harvard University has shown that, if one player knows the difference between the models used by the agents, the loss can be reduced. In this work we focus on the estimating problem of the model of the second agent by the first one.

We first propose a simple way of choosing the likeliest model among a finite set and then show the failure of the classical Inverse Reinforcement Learning methods to predict future behavior of the second agent. Based on a small number of observations and on general hypothesis about the agent's behavior, we provide an exact description of the set of models consistent with the observations. When the number of observations increases, our parametrization is not efficient and we have to use some heuristics to find a realistic model of the world.

Finally we show empirically that some models are strictly equivalent: they produce exactly the same behavior in every situation. As a consequence, we propose to use a classification method among the set of equivalence classes. We can predict the future behavior of the agent without knowing its underlying model, whose estimation is impossible.

ACKNOWLEDGMENTS

I would like to thank Christos Dimitrakakis, my supervisor for this internship. He made me work on an interesting problem and gave me good insights. He let me the freedom to explore the directions I chose but always helped me when I was lost.

Thank you to David Parkes, head of the EconCS group at the School of Engineering and Applied Science of Harvard University, for welcoming me to his group. I am very grateful to all members of the group for the fantastic atmosphere of work in this laboratories, particularly to the members of the Helper-AI team with whom I have been working for a few months: Goran Radanovic and Paul Tylkin. I also want to mention Thibaut Horel with whom I had very inspiring discussions even though he was not part of this project.

Thank you to Aristide Tossou and Hannes Eriksson, PhD students at Chalmers University, who welcomed me to Sweden at the beginning of my internship. Finally I thank all the people that helped me with the administrative work, before and during the internship: Rebecca Cyren, Gina Scribner and Ann Marie King.

TABLE OF CONTENTS

1
FORMULATION OF THE PROBLEM

1.1 Motivation: the Helper-AI framework

We consider systems where an artificial intelligence (AI) interacts with an human being. Both AI and human are able to make decisions that affect the world. Each of them will be in consequence affected by the succession of states of the world. If they behave rationally, they will attempt to lead the world to a satisfying state (for themselves). In a game framework, for instance when a human plays chess against a computer program, the human player wants the system to converge to a winning position whereas the computer tries to defeat him.

We want to design a system where the AI helps the human. Consequently the AI adopts the same goal as the human. It aims at reaching the same 'satisfying states'. Then if AI and human agree on their goals, why do we need an artificial intelligence? Both agents should make the same decisions. We observe that in a lot of practical cases, human beings do not behave optimally according to their own goals. In other words, they have a wrong belief about the consequences of their decisions. They may use a simplified model of reality because of their lack of perception or of computing power. They might also be fundamentally wrong. For instance in an autonomous plane, the on-board computer could have more information than the pilot about the weather and the state of the motor (temperature of different parts, oil consumption, resistances). Moreover the belief over the system of an experienced pilot would surely have some bias.

Knowing that fact, if we want the system to evolve optimally according to the model of the world used by the AI (that we assume to be accurate), the AI should make all decisions. Nevertheless the danger of such a system is clear. It is almost certain that the human would stop a self-driving car if it behaved very strangely in comparison with a human driver. By definition our intelligent system will use a model of the world that could be wrong and the human being must be free to behave in a different way.

Consequently, from an AI designer point of view, we want to create an helper-AI that would take into account the fact that it collaborates with a human who uses a different model of the world. The first step in the design of the system is the estimation of the model of the world used by the human.

1.2 Model and definitions

We use a classical model in Reinforcement Learning: Markov Decision Processes ([Put05]). We first define a Markov Decision Process for one player and then show how it can be generalized to a two-player game. Finally we define precisely the estimation problem we want to solve.

1.2.1 • Markov Decision Process

A Markov Decision Process (MDP) controls the evolution of an agent within a set of states S. In each state $s \in S$, there is a set of allowed actions A_s. For any state $s \in S$, the agent chooses an action $a \in A_s$, there is a probability distribution $P_{s,a}$ over the set S such that, from the state s and taking the action a, the agent will go to the state $s' \in S$ with probability $P_{s,a}(s')$. In addition, he will receive a reward $R_{s,a,s'}$ (that could depend only on s, (s,a) or (s,a,s')).

In this work, we focus on finite MDPs where both sets of states and actions are finite. We also make the assumption that the set of allowed actions A_s does not depend on the state $s \in S$. This hypothesis is not restrictive and only aims at simplifying the notations.

Definition 1. *A Markov Decision Process (MDP) over a finite states set S and a finite actions set A is a tuple $(P, R) \in \mathcal{M}_{S\times A,S}(\mathbb{R}) \times \mathbb{R}^{S\times A\times S}$.*

P is the transition matrix. In particular, we have:

$$\forall (s,a,s') \in S \times A \times S, 0 \leq P_{s,a}(s') \leq 1$$

$$\forall (s,a) \in S \times A, \sum_{s' \in S} P_{s,a}(s') = 1$$

R is the rewards vector.

1.2.2 • Strategies and state values

Definition 2. *Let (P,R) be a MDP over the states set S and actions set A. A sequence $(s_t, a_t)_{t\in\mathbb{N}} \in (S \times A)^{\mathbb{N}}$ is called a path in the MDP (P,R).*

Definition 3. *Let (P,R) be a MDP and $(s,a)_{t\in\mathbb{N}}$ a path. Given a discount factor $\gamma \in [0,1[$, the cumulative reward associated with the path $(s,a)_{t\in\mathbb{N}}$ is given by:*

$$R_C = \sum_{t\in\mathbb{N}} \gamma^t R_{s_t,a_t,s_{t+1}}$$

A rational agent evolving in a given MDP, tries to maximize the cumulative reward he gets over time. The discount factor represents the preference of the agent for an immediate over a future reward. The agent only acts through the choice of the actions. In this work, we assume that the agent behaves like if he knew perfectly the parameters of the MDP (P,R). As a consequence, its behavior does not change with time, he uses a Markovian strategy.

Definition 4. *Given a state set S and an action set A, a Markovian strategy (or policy) is the list of $|S|$ probability distributions over A.*

In the state $s \in S$, the agent chooses the action $a \in A$ with probability $\pi_s(a)$

Π is the set of all Markovian strategies over the sets (S,A).

Π^D is the subset of Π that contains all deterministic policies.

Definition 5. *Given a strategy π, the value $V_\pi(s)$ of a state s is the total expected reward of an agent starting from the state s and playing the strategy π.*

$$\forall s \in S, V_\pi(s) = \mathbb{E} \sum_{t\in\mathbb{N}} \gamma^t R_{s_t,a_t,s_{t+1}}$$

where the expectation is taken over all paths with the constraint $s_0 = s$. The probability of a given path depends on the policy π. More precisely:

$$\forall s \in S, V_\pi(s) = \sum_{t\in\mathbb{N},(s_1,\dots s_{t+1})\in S^t,(a_0,\dots a_t)\in A^t} \gamma^t R_{s_t,a_t,s_{t+1}} \prod_{t'=1}^{t} P_{s_{t'},a_{t'}}(s_{t'})\pi_{s_{t'}}(a_{t'})$$

Definition 6. *Let π be a policy of the agent. The associated expected reward vector $R_\pi \in \mathbb{R}^S$ is given by:*

$$\forall s \in S, R_\pi(s) = \sum_{(a,s')\in A\times S} \pi_s(a) P_{s,a}(s') R_{s,a,s'}$$

We remark that if R does not depend on the action and the arrival state, then for every policy π, $R_\pi = R$.

Definition 7. *Let π be a policy of the agent. The associated transition matrix P_π is given by:*

$$\forall (s,s') \in S^2, P_\pi(s,s') = \sum_{a\in A} \pi_s(a) P_{s,a}(s')$$

Theorem 1. *Given a Markovian strategy π, the associated value function is the unique solution of the system known as Bellman equation:*

$$V_\pi = R_\pi + \gamma P_\pi V_\pi$$

It gives:

$$V_\pi = (I - \gamma P_\pi)^{-1} R_\pi$$

Proof. Since P_π is a transition matrix, all its eigenvalues' modules are 1. As a consequence the matrix $(I - \gamma P_\pi)$ is invertible for $|\gamma| < 1$. □

Definition 8. *The Q-function associated to the policy π is the vector $Q \in \mathbb{R}^{S\times A}$ that gives the expected cumulative reward obtained from the policy π after playing action a in state s.*

$$\forall (s,a) \in S \times A, Q_s(a) = \sum_{s'\in S} P_{s,a}(s')(R_{s,a,s'} + \gamma V_\pi(s'))$$

1.2.3 • Optimal strategy

Definition 9. *In order to compare value functions, we take the classical definition of order between two vectors. Let $n \in \mathbb{N}^*$, $(v, v') \in \mathbb{R}^n$. v is said superior to v' if:*

$$\forall i \in \{1, \dots n\}, v_i \geq v'_i$$

Then we use the notation $v \geq v'$.

Theorem 2. *Let (P, R) be a MDP over a finite states set S and a finite actions set A. Let γ be a discount factor. Then there exists an optimal strategy π^* such that:*

$$\forall \pi \in \Pi, V_{\pi^*} \geq V_\pi$$

We use the notation $V^* = V_{\pi^*}$.

V^* is the unique solution of the system:

$$\forall s \in S, V(s) = \max_{a\in A} \sum_{s'\in S} P_{s,a}(s')(R_{s,a,s'} + \gamma V(s'))$$

and:

$$\forall (s,a) \in S \times A, \pi^*_s(a) \neq 0 \Leftrightarrow a \in \operatorname{argmax}_{a'\in A} \sum_{s'\in S} P_{s,a'}(s')(R_{s,a',s'} + \gamma V(s')$$

Theorem 3. *An approximation of V^* can be computed with arbitrary precision $\epsilon > 0$ thanks to the following value iteration algorithm:*

1. *Initialize the vector $V \in \mathbb{R}^S$ to some arbitrary value.*
2. *$\forall s \in S, V'(s) = \max_{a \in A} \sum_{s' \in S} P_{s,a}(s')(R_{s,a,s'} + \gamma V(s'))$*
3. *if $||V - V'||_\infty > \frac{\epsilon(1-\gamma)}{2\gamma}$, $V = V'$ and go back to step 2.*
4. *Return V'.*

Proof. The proof of the two previous theorems can be found in [Put05]. They are based on an attractive fixed point argument. □

Definition 10. *The Q-value vector $Q^* \in \mathbb{R}^{S\times A}$ is defined by:*

$$\forall (s,a) \in S \times A, Q^*_s(a) = \sum_{s' \in S} P_{s,a}(s')(R_{s,a,s'} + \gamma V^*(s'))$$

1.2.4 • Markov Decision Process for two cooperative players

A Markov Decision Process (MDP) controls the evolution of the system within a set of states S. The first agent will be called the AI, and the second agent the human. In each state $s \in S$, there is a set of allowed actions A^{AI} for the AI and A^H for the human.. For any state $s \in S$ and any pair of actions $(a,b) \in A^{AI} \times A^H$ there is a probability distribution $P_{s,a,b}$ over the set S such that, from the state s and taking the actions (a,b), the system will go to the state $s' \in S$ with probability $P_{s,a,b}(s')$. In addition, the human and AI receive a reward $R_{s,a,b,s'}$.

In this work, we focus on finite MDPs where both sets of states and actions are finite. Both agents know perfectly the reward vector R but they have potentially different beliefs over the transition matrix P, respectively P^{AI} and P^H.

Definition 11. *A multi-view MDP over the finite states set S and the finite actions sets A^{AI} and A^H is the tuple $(R, P^{AI}, P^H, \gamma) \in \mathbb{R}^{S\times A\times S} \times \mathcal{M}_{S\times A^{AI}\times A^H, S}(\mathbb{R}) \times \mathcal{M}_{S\times A^{AI}\times A^H, S}(\mathbb{R}) \times [0,1[$.*

R is the rewards vector, shared by both agents.

P^{AI} is the transition matrix of the AI. We assume it is the real transition matrix of the system.

P^H is the belief of the human over the transition matrix, potentially different from P^{AI}.

γ is the discount factor.

1.2.5 • Human point of view

When both agents know the transition matrix of the system (i.e. $P^{AI} = P^H = P$), since they receive the same reward, they can achieve an optimal joint policy. Nevertheless, when one of the agent has a wrong belief about the transition matrix of the system, he may use a suboptimal policy. We assume that the human uses a wrong model of the system, ie $P^H \neq P$.

More precisely, the AI will commit to a policy π^{AI}, i.e. in the state s the AI will choose action $a \in A^{AI}$ with probability $\pi^{AI}_s(a)$. Then the human will face a new MDP P' where he knows exactly the behavior of the AI. We have:

$$\forall (s,b,s') \in S \times A^H \times S, P'_{s,b}(s') = \sum_{a \in A^{AI}} \pi^{AI}_s(a) P^H_{s,a,b}(s')$$

In this MDP, he will behave according to his belief about the world. We observe his answer policy π^H.

1.3 Previous results in the Helper-AI problem

The Helper-AI team at Harvard University has shown bounds over the loss in term of value functions that is created by a difference between the models of the AI and the human, when both players play optimally according to their own model. The results have not been published yet ([PDRT]).

Theorem 4. *In a MDP (R, P^{AI}, P^H, γ), there exists an optimal joint policy (that maximizes the values vector).*

Proof. Consider the MDP $\mathcal{M} = (R', P')$ over the sets $S' = S$ and $A' = A^{AI} \times A^H$ with:

$$\forall (s, a, b, s') \in S \times A^{AI} \times A^H \times S, R'_{s,(a,b),s'} = R_{s,a,b,s'}$$

and

$$P'_{s,(a,b)}(s') = P_{s,a,b}(s')$$

The theorem 2 gives the existence of an optimal policy π'^* for $\mathcal{M}$. Let $(\pi^*_{AI}, \pi^*_H) = \pi'^*$ is the optimal joint policy for the original two-player MDP. □

Definition 12. *An optimal uninformed policy of the AI is a policy π_{AI} such as there exists an optimal joint policy of the MDP π' and a policy of the human π_H such that $(\pi_{AI}, \pi_H) = \pi'$.*

Definition 13. *One optimal answer for the human to the policy of the AI π_{AI} is one optimal policy in the MDP $(R_{\pi_{AI}}, P^H_{\pi_{AI}}, \gamma)$. We use the notation $\pi^*_H(\pi_{AI}, P^H)$.*

For each policy of the AI π^{AI}, the human plays the optimal answer according to his own model: $\pi^*_H(\pi_{AI}, P^H)$. Nevertheless the AI expects the human to play optimally according to the real model, it expects the policy $\pi^*_H(\pi_{AI}, P^{AI})$. Consequently he chooses its policy $\pi^*_{AI} \in \operatorname{argmax}_{\pi_{AI}} V_{\pi_{AI},\pi^*_H(\pi_{AI},P^{AI})}$: he chooses an optimal uninformed policy.

It generates a loss:

$$L = V_{\pi^*_{AI},\pi^*_H(\pi^*_{AI},P^{AI})} - V_{\pi^*_{AI},\pi^*_H(\pi^*_{AI},P^H)}$$

Nonetheless, if the AI knew the transition matrix of the human, it could choose $\pi^*_{AI} \in \operatorname{argmax}_{\pi_{AI}} V_{\pi_{AI},\pi^*_H(\pi_{AI},P}$ and consequently reduce the loss L.

In the example shown in the figures 1 and 2, both agents receive a reward of +1 in state 1 and 0 in state 0. The optimal joint policy is then the action a for both AI and human in state 0 and action a for the AI and b for the human in state 1. Nevertheless, according to the human's belief, the trajectory does not depend on AI's actions. As a consequence the human will always play the same policy: action b in state 0 and action a in state 1. As a consequence, the optimal policy of the AI that takes into account the model of the human is action b in state 0 and action b in state 1. As shown in the table 1, it leads to an important increase of the expected cumulative reward.

Policy of the AI	Value of state 0	Value of state 1
Optimal uninformed policy	0	1
Optimal informed policy	1.3	2.9

Table 1: Value gain brought by the knowledge of human's model with a discount factor of 0.9.

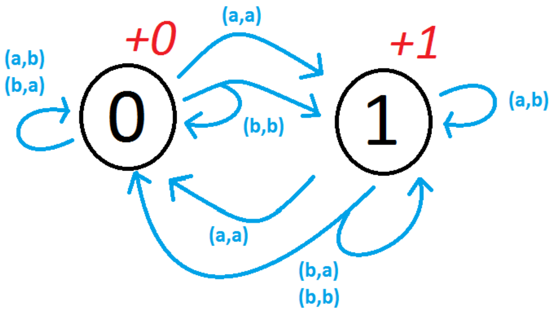


Figure 1: Real MDP, known by the AI.

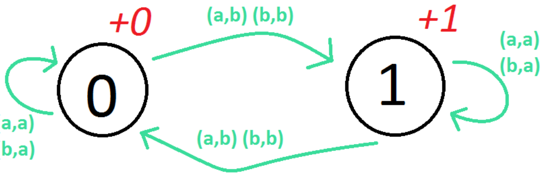


Figure 2: Model used by the human. A pair (a, b) indicates that AI plays action a and human plays action b. When the arrow points two different states, it means that the joint action will lead to one of the two states with a probability $\frac{1}{2}$

1.4 Estimating problem

In order to choose the best policy for the AI, we want to be able to predict the answer policy of the human.

How much information does the tuple (π^{AI}, π^{H}) give about the human belief? Is it possible to determine with a given precision what is the transition matrix used by the human? If so, how many observations do we need, and otherwise what useful information can we extract?

The set of observations is a list of tuples (π^{AI}, π^{H}) where π^{H} is the human policy answer to the policy π^{AI} of the AI. We aim at describing the set of transition matrix that are consistent with these observations. Such an admissible transition matrix or any other way of predicting the behavior of the human will be evaluated by the measure of the difference for other policies of the AI between the predicted policy of the human and his actual policy.

In the second section we show that we can efficiently pick the best model among a finite set. Afterwards, we try to generate a good model thanks to the classical Inverse Reinforcement Learning method of estimation of the transition matrix in the third section. In the fourth part we give a parametrization of the set of admissible models for one observations and in the fifth section we give some heuristics to find an element of the intersection of this sets for several observations. Finally, we discuss in the last section the limits to the information given by observations and propose a prediction method without estimating the model of the human.

2
CHOICE OF A MODEL IN A FINITE SET

We first assume that the transition matrix of the human P belongs to a finite set of models $\mathcal{P}$. We observe the behavior of the human for several policies of the AI. The assumption of a softmax behavior for the human allows to define a likelihood and then to choose the likeliest model.

2.1 Observations

The observations are a finite list of tuples (π_{AI}, π_H) where π_H is the answer policy of the human to the policy π_{AI}. In practice, π_H will be given in this part by a finite set of tuples $(s, a) \in S \times A$ where a has been chosen by the human according to the policy π_H.

Definition 14. *The observation set is a finite set $\mathcal{O}$ where every element of $\mathcal{O}$ is a tuple (π_{AI}, s, a) where a is the action played by the human in state s when the AI commits to π_{AI}.*

2.2 Idea of the estimating process

Definition 15. *The optimal probability of an observation $(\pi_{AI}, s, a) \in \mathcal{O}$ for the model $P^H \in \mathcal{P}$ is the real number $\pi^*_{\pi_{AI},P^H,s}(a)$ where $\pi^*_{\pi_{AI},P^H}$ is the optimal answer such that*

$$\forall (s,a) \in S \times A^H, Q^*_s(a) = V(s) \Rightarrow \pi^*_{\pi_{AI},P^H,s}(a) = \frac{\mathbb{1}_{Q^*_s(a)=V(s)}}{\#\{a' \in A^H | Q^*_s(a') = V(s)\}}$$

It is the probability of observing action a in state s assuming that the human plays optimally.

Definition 16. *The optimal likelihood $\mathcal{L}_{P^H}$ of the $P^H \in \mathcal{P}$ for the observation set $\mathcal{O}$ is the product of the likelihoods of the elements of $\mathcal{O}$.*

$$\mathcal{L}_{P^H} = \prod_{(\pi_{AI},s,a)\in\mathcal{O}} \frac{\pi^*_{\pi_{AI},P^H,s}(a)}{\sum_{P^{H\prime}\in\mathcal{P}} \pi^*_{\pi_{AI},P^{H\prime},s}(a)}$$

By definition, the human plays optimally according to the model P^H if and only if $\mathcal{L}_{P^H} > 0$. As a consequence, we compute $\mathcal{L}_{P^H}$ for every element of $\mathcal{P}$. A good guess for the model of the human would be any $P^H \in \mathcal{P}$ such that $\mathcal{L}_{P^H} > 0$.

Nevertheless, in all our experiments, when $\mathcal{P}$ is generated randomly, we get $\mathcal{L}_{P^H} = 0$ for all the elements of $\mathcal{P}$. Then we need a subtler way of choosing the best model in $\mathcal{P}$.

2.3 A softmax model for the human

Since we are generally not able to randomly generate a transition matrix that would explain the observations, we assume that the policy of the human obeys to a softmax model. This assumption allows to compare the likelihoods of two models that are not entirely satisfying and to pick the best one.

We assume that the human is not totally certain of the validity of his model. As a consequence in state $s \in S$, he chooses an action $a \in A$ according to its Q-value but does not want to eliminate a priori suboptimal actions. If we assume that he is driven by some exploration-exploitation dilemma, he will be more likely to play the best actions. We assume that he uses a softmax policy.

Definition 17. *The softmax policy $\pi_{P^H,\beta}$ of parameter $\beta \in \mathbb{R}_+$ of the transition matrix $P^H \in \mathcal{P}$ is given by: t*

The softmax parameter β is the certainty of the human. $\beta = 0$ generates an uniformly random policy whereas $\beta \to +\infty$ produces an optimal policy according to the model.

Definition 18. *The softmax likelihood of parameter β and observation $(\pi_{AI}, s, a) \in \mathcal{O}$ for the model $P^H \in \mathcal{P}$ is the real number:*

$$l = \frac{\pi_{\pi_{AI},P^H,\beta,s}(a)}{\sum_{P^{H\prime} \in \mathcal{P}} \pi_{\pi_{AI},P^{H\prime},\beta,s}(a)}$$

where $\pi_{\pi_{AI},P^H,\beta}$ is the answer softmax policy of parameter β of the model P^H to the policy π_{AI}.

Definition 19. *The softmax likelihood of parameter β $\mathcal{L}_{P^H,\beta}$ for the observation set $\mathcal{O}$ of the model $P^H \in \mathcal{P}$ is the product of the likelihoods for all the elements of $\mathcal{O}$.*

$$\mathcal{L}_{P^H,\beta} = \prod_{(\pi_{AI},s,a)\in\mathcal{O}} \frac{\pi_{\pi_{AI},P^H,\beta,s}(a)}{\sum_{P^{H\prime} \in \mathcal{P}} \pi_{\pi_{AI},P^{H\prime},\beta,s}(a)}$$

The estimated model will then be chosen among the transition matrices with the highest softmax likelihood.

2.4 Results

This process always allows to choose a transition matrix with the highest likelihood.

We verify that when we know the softmax parameter and when the actual transition matrix of the human belongs to the set of models $\mathcal{P}$, our estimating procedure generally chooses this model.

The actual model of the human is the likeliest in most cases (figure 3). The relative likelihood grows with the softmax parameter that increases the difference between two transition matrices. We can compute the success rate of our estimating procedure as the percentage of instances when the algorithm chooses the right model (figure 4).

Moreover, the success rate is still satisfying when the AI does not know the softmax parameter of the human (figure 5). This allows to use this estimating procedure in a wide range of situations.

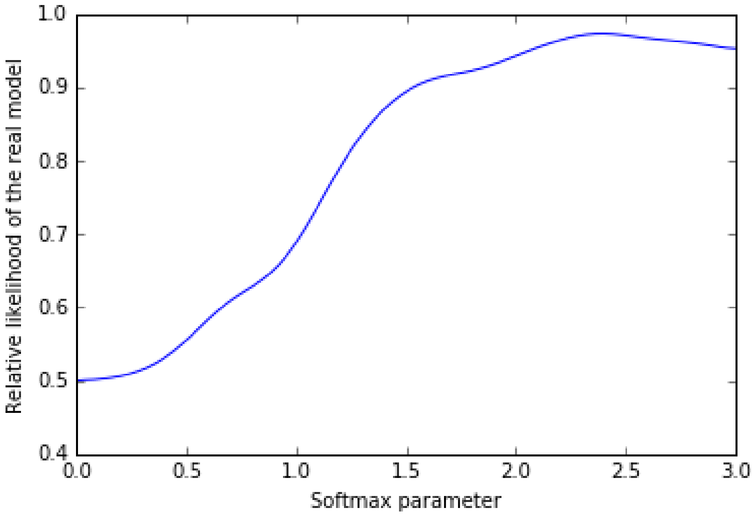


Figure 3: Relative likelihood of the model of the human over the known softmax parameter. The relative likelihood is the ratio between the likelihood of the actual model of the human and the sum of all likelihoods. Data generated for $\#S = \#A^{AI} = \#A^{H} = \#\mathcal{P} = 2$ and $\#\mathcal{O} = 200$, average over 200 simulations.

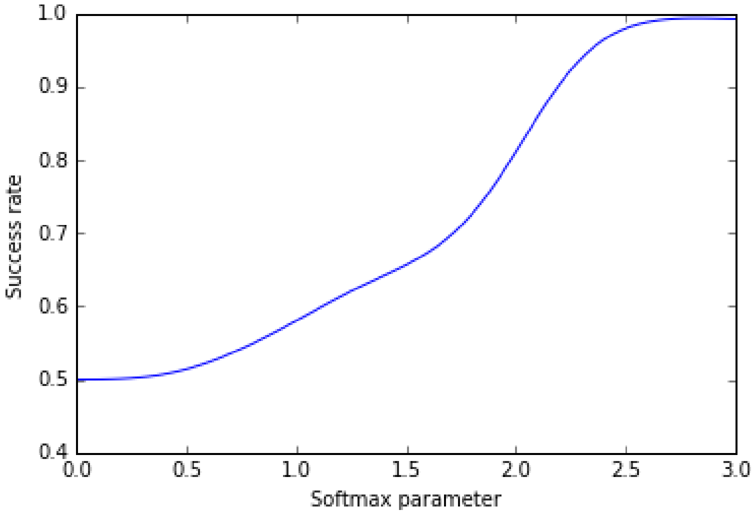


Figure 4: Success rate over the softmax parameter. Data generated for $\#S = \#A^{AI} = \#A^{H} = \#\mathcal{P} = 2$ and $\#\mathcal{O} = 200$, average over 200 simulations.

2.5 Conclusion

We designed a choice procedure based on a maximum likelihood principle that allows to pick the best transition matrix among a finite set. This procedure chooses the actual model of the human when it is in the set of possible models. Otherwise it chooses another model that best suits the observations set.

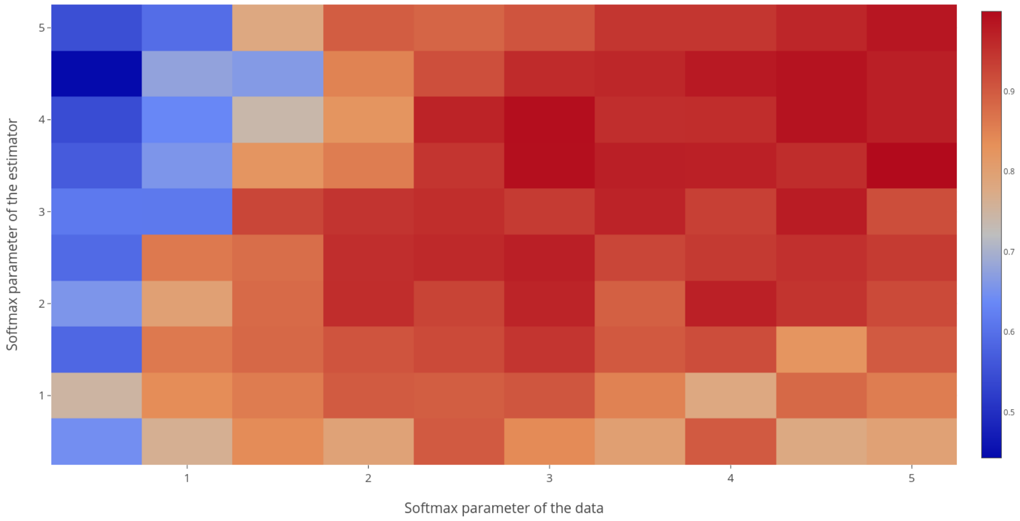


Figure 5: Success rate. We generate random sets of MDP and observations based on one MDP in the set. The success rate is the proportions of instances when the relative likelihood of the actual MDP is the highest. Data generated for $\#S = \#A^{AI} = \#A^{H} = \#\mathcal{P} = 2$ and $\#\mathcal{O} = 200$, average over 50 simulations.

However we have no control over the quality of the transition matrices in our finite set of possible models. That's why we would like to generate a model directly from the observations.

3
CLASSICAL INVERSE REINFORCEMENT LEARNING

In literature, we are used to consider that the transition matrix of the system is known. We can not predict the behavior of the human because we do not know the rewards vector. This point of view is orthogonal to ours. However it is clear that we can construct two tuples (P_1, R_1) and (P_2, R_2) that produce the same behavior with $P_1 \neq P_2$. As a consequence we evaluate the classical inverse reinforcement learning pipeline according to its power of prediction.

First we describe the set of admissible rewards vectors for one observation (π_{AI}, π_H) thanks to the results in [NR00]. We then consider that the human is playing in a classical MDP P that linearly depends on P^H and π_{AI} In this MDP, he plays with the policy $\pi = \pi_H$.

3.1 Observation of the optimal policy of the human

We observe one optimal policy of the human π^*.

3.1.1 • Expression of the values and Q-values

We first consider that the reward is only a function of the current state $s \in S$ and does not depend on the action $a \in A$ chosen by the human. As a consequence, the value vector $V \in \mathbb{R}^S$ is given by:

$$V = (I - \gamma P^*)^{-1} R \tag{1}$$

where $P^* \in \mathcal{M}_{S,S}(\mathbb{R})$ is the transition matrix associated with the optimal policy π^*:

$$\forall (s, s') \in S^2, P^*_s(s') = \sum_{a \in A} \pi^*_s(a) P_{s,a}(s') \tag{2}$$

The Q-vector associated with the action $a \in A$, $Q_a \in \mathbb{R}^S$, gives for every state $s \in S$ the expected optimal cumulated rewards when playing action a in state s.

$$\forall a \in A, Q_a = R + \gamma P^a V \tag{3}$$

where $P^a \in \mathcal{M}_{S,S}(\mathbb{R})$ is the transition matrix associated with the action a:

$$\forall a \in A, \forall (s, s') \in S^2, P^a_s(s') = P_{s,a}(s') \tag{4}$$

3.1.2 • Set of admissible reward vectors

Theorem 5. *Given the observation of one optimal policy π^*, the set of admissible rewards vectors R_{π^*} is given by:*

$$R_{\pi^*} = \{R \in (R)^S | \forall a \in A, ((I - \gamma P^a)(I - \gamma P^*)^{-1} - I)R \geq 0\}$$

Proof. The policy π^* is optimal if and only if for all action a, for all state s, taking the action a in the state s leads to an expected cumulated reward inferior to the action decided by the policy π^*, i.e.:

$$\forall a \in A, Q_a = (I + \gamma P^a (I - \gamma P^*)^{-1}) R \leq V = (I - \gamma P^*)^{-1} R$$

□

Hence R_{π^*} is defined by a system of $S \times A$ linear constraints. This system generally admits an infinite set of solutions.

3.1.3 • Choice of a rewards vector

The usual way ([NR00]) of choosing among this set is the maximization of the 'explaining power' of the rewards vector R. In other words, we maximize the gap between the expected gain of the optimal policy π^* and the second best action for every state. Note that we assume that π^* is deterministic.

Definition 20. *The chosen rewards vector $R^C_{\pi^*}$ is given by:*

$$R^C_{\pi^*} \in \underset{R \in \mathbb{R}^S}{\operatorname{argmax}} \min_{\pi \in \Pi_d \{\pi^*\}} ((I - \gamma P^\pi)(I - \gamma P^*)^{-1} - I)R$$

where Π_d is the finite set of deterministic policies.

This is a linear programming problem. If it admits an admissible point, we can add some bounds for R, depending on the context of the problem, to ensure the existence of a finite solution.

3.2 Observation of a softmax policy

3.2.1 • Stochastic human policies

Nevertheless this way of choosing an admissible rewards vector is not satisfying. In fact, almost always, observations reveal that humans do not commit to a deterministic policy. Then do they play optimally?

Let π^* be an optimal policy. Then:

$$\forall s \in S, \forall (a_1, a_2) \in A^2 | a_1 \neq a_2 :$$

$$\pi^*_s(a1) > 0 \wedge \pi^*_s(a_2) > 0 \Rightarrow Q^*_s(a_1) = Q^*_s(a_2)$$

This is of course possible, but the set of MDPs that allow a stochastic optimal policy is of measure 0 in the set of all MDPs. Do humans always choose a model in this reduced set? It is very unlikely since there is no way to distinguish this models without actually solving the MDP. We prefer a more natural assumption: the human uses the softmax policy of parameter $\beta \in \mathbb{R}_+$.

3.2.2 • Constraints over the rewards vector

If we observe the softmax policy π of the human and if we know the parameter β, we can restrict the set of admissible rewards vectors.

Theorem 6. *One optimal policy π^* can be constructed from a softmax policy π of parameter $\beta > 0$.*

Proof.

$$\forall s \in S, \forall (a_1, a_2) \in A^2, \pi_s(a_1) < \pi_s(a_2) \Leftrightarrow Q^*_s(a_1) < Q^*_s(a_2)$$

Then for $s \in S$, we define:

$$a(s) \in \underset{a \in A}{\operatorname{argmax}}\, Q^*_s(a)$$

Let π^* be the deterministic policy defined by:

$$\forall(s,a) \in S \times A, \pi^*_s(a) = 1 \Leftrightarrow a = a(s)$$

π^* is an optimal policy. □

Theorem 7. *Let π be the softmax policy of parameter β. The set of admissible rewards vectors R_π is given by:*

$$R_\pi = \{R \in \mathbb{R}^S | \forall(s,a_1,a_2) \in S \times A \times A, \gamma[(P^{a_1} - P^{a_2})(I - \gamma P^*)^{-1}R]_s = \frac{1}{\beta}\ln(\frac{\pi_s(a_1)}{\pi_s(a_2))}\}$$

where P^* is the transition matrix associated with one optimal policy constructed from π

Proof. Let $(s,a_1,a_2) \in S \times A \times A$. Like previously we have:

$$Q(a_1) = R + \gamma P^{a_1}(I - \gamma P^*)R$$

and:

$$\frac{\pi_s(a_1)}{\pi_s(a_2))} = e^{\beta(Q^*_s(a_1) - Q^*_s(a_2))}$$

i.e.

$$Q^*_s(a_1) - Q^*_s(a_2) = \frac{1}{\beta}\ln(\frac{\pi_s(a_1)}{\pi_s(a_2)}) = \gamma[(P^{a_1} - P^{a_2})(I - \gamma P^*)^{-1}R]_s$$

This set of admissible rewards vectors is the intersection of $S \times A - 1$ affine hyperplanes in a space of dimension S. Generally it is empty. □

3.2.3 • Enlargement of the rewards vector space

One could argue that the variation of transition matrix of dimension $S \times A \times S$ can not be explained by a rewards vector of dimension S. Then we give the same dimension to the rewards space with a reward $R_{s,a,b,s'}$ that depends on the departure state $s \in S$, the action of the AI $a \in A^{AI}$, the action of the human $b \in A^H$ and the arrival state $s' \in S$. This generally allows a non empty set of admissible rewards for a single observation (π^{AI}, π^H).

Nonetheless, when we increase the number of observations and take the intersection of respective admissible rewards sets, we reach an empty intersection after a reduced number of policies of the AI: figure 6. In other words, we can find a rewards vector R that would explain the behavior of the human for a reduced number of policies of the AI, if we assume that he uses the transition matrix P^H. Nevertheless, in every simulation we made, we were able to find a finite set of policies of the AI such that observations generated from this policies were not consistent with the initial transition matrix of the human.

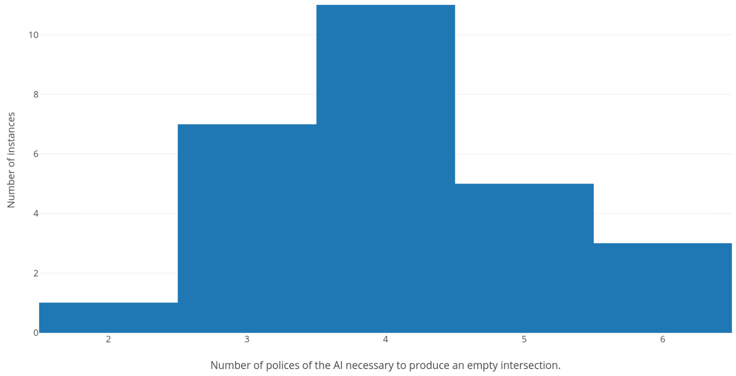


Figure 6: Number of random policies of the AI in the observation set before we reach an empty intersection. Data generated from 27 MDPs with $(\#S, \#A^{AI}, \#A^{H}) \in \{2, 3, 4\}^3$.

3.3 Conclusion

The classical point of view in Inverse Reinforcement Learning focuses on the estimation of the rewards vector. This process is linear and is not robust when the data is generated by a different transition matrix. Generally, for a rewards vector R and two transition matrices P and P', for every rewards vector R', there exists a policy π_{AI} of the AI such as the associated softmax policies of the human for (P, R) and (P', R') are different.

As a consequence, to predict the behavior of the human for future policies of the AI, we need to estimate the transition matrix.

4 ESTIMATION OF THE TRANSITION MATRIX FOR ONE OBSERVATION

Now we assume that we know the rewards vector R, the discount factor γ and the softmax parameter β. We want to estimate the transition matrix P. In this part, we have only one observation (π^{AI}, π^{H}) then we consider that we are in a classical one player MDP and that we observe the softmax policy π.

4.1 Partition function and value

The softmax policy of parameter β is defined as follow:

$$\forall (s,a) \in S \times A, \pi_s(a) = \frac{e^{\beta Q^*(s,a)}}{\sum_{a' \in A} e^{\beta Q^*(s,a')}}$$

Definition 21. *By analogy with statistical physics the partition function Z_s of a state $s \in S$ is given by:*

$$Z_s = \sum_{a' \in A} e^{\beta Q^*(s,a')}$$

As a consequence:

$$\forall (s,a) \in S \times A, \quad Q^*(s,a) = \frac{1}{\beta}[\ln(Z_s) + \ln(\pi_s(a))]$$

Definition 22. *The entropy H_s^{π} of the policy π at state s is defined by: $H_s^{\pi} = \max_{a \in A} \ln(\pi_s(a)$*

Then:

$$\forall s \in S, V^*(s) = \max_{a \in A} Q^*(s,a) = \frac{1}{\beta}(\ln Z_s + H_s^{\pi}).$$

And partition function and value function represent the same notion:

$$\forall s \in S, lnZ_s = \beta V^{\pi}(s) - H_s^{\pi}$$

$$\forall (s,a) \in S \times A, Q^{\pi}(s,a) = V_s^{\pi} + \frac{1}{\beta}(ln(\pi_s(a)) - H_s)$$

This also holds for the optimal value function, we have the following result.

Theorem 8. *The pair $(P, V^*) \in \mathcal{M}_{S \times A, S}(\mathbb{R}) \times \mathbb{R}^S$ is consistent with the observed policy π for the softmax parameter β if and only if:*

$$\forall (s,a) \in S \times A, V_s^* + \frac{1}{\beta}(\ln(\pi_s(a)) - H_s) = \sum_{s' \in S} P_{s,a}(s')(R_{s,a,s'} + \gamma V_{s'}^*)$$

π and H can be estimated thanks to the observations $\mathcal{O}$, R and β are known. The admissible transition matrices for the problem are the normalized matrices P such as $\exists V^* \in \mathbb{R}^S$ such as the tuple $(P, V)^*$ satisfies the condition of theorem 8. By definition of the value function, such a V^* associated with a given admissible P is unique and is the value function of P.

4.2 Parametrization of the set of admissible transition matrices by the value function

Definition 23. *The polytope of the transition matrices of $\mathcal{M}_{S\times A,S}(\mathbb{R})$ is:*

$$\mathcal{B}=\{P\in mathcalM_{S\times A,S}(\mathbb{R})|\forall(s,a,s')\in S\times A\times S,0\leq P_{s,a}(s')\leq 1,\forall(s,a)\in S\times A,\sum_{s'\in S}P_{s,a}(s')=1\}$$

Theorem 9. *Let $V\in\mathbb{R}^S$. The two following propositions are equivalent.*

$$\exists P\in\mathcal{B},\forall(s,a)\in S\times A,V_s+\frac{1}{\beta}(ln(\pi_s(a))-H_s)=\sum_{s'\in S}P_{s,a}(s')(R_{s,a,s'}+\gamma V_{s'})\tag{5}$$

$$\forall(s,a)\in S\times A,\min_{s'\in S}(R_{s,a}+\gamma V_{s'})\leq V_s+\frac{1}{\beta}(ln(\pi_s(a))-H_s)\leq\max_{s'\in S}(R_{s,a,s'}+\gamma V_{s'})\tag{6}$$

And for a given V that verifies the condition (6), the associated admissible P are given by the possible coefficients of the barycentre in equation (5). Notice that a transition matrix admits by definition only one value function. We now want to describe the set of admissible value functions, ie the value functions that satisfy (6).

For a given V verifying the condition (6), the associated set of admissible transition matrices $\mathcal{P}(V)$ is the cartesian product of $S\times A$ polytopes of $\mathbb{R}^S$ defined by the intersection of $\mathcal{B}$ with the solutions of equation (5). As a consequence, it is a polytope of $\mathbb{R}^{S\times A\times S}$ and the orthogonal projection of any transition matrix over this set according to any scalar product can be computed with linear programming.

4.3 Admissible value functions

Definition 24. *A value function $V\in\mathbb{R}^S$ is admissible if it verifies the condition 6. It means that there exists a transition matrix P such that V is the value function associated with P and P generates the policy π.*

4.3.1 • Auxiliary variables

The condition (6) can be written:

$$\forall(s,a)\in S\times A,\min_{s'\in S}V_{s'}\leq\frac{V_s}{\gamma}+\frac{1}{\gamma}(\frac{1}{b}(ln(\pi_s(a))-H_s)-R_{s,a})\leq\max_{s'\in S}V_{s'}\tag{7}$$

Definition 25. *We introduce the following auxiliary variables, $\forall s\in S$:*

$$m_s=\min_{a\in A}\frac{1}{\gamma}(\frac{1}{b}(ln(\pi_s(a))-H_s)-R_{s,a})$$

$$M_s=\max_{a\in A}\frac{1}{\gamma}(\frac{1}{b}(ln(\pi_s(a))-H_s)-R_{s,a})$$

Then (6) is equivalent to:
$\forall s\in S$:

$$\begin{cases}\min_{s'\in S}V_{s'}\leq\frac{V_s}{\gamma}+m_s\\ \max_{s'\in S}V_{s'}\geq\frac{V_s}{\gamma}+M_s\end{cases}$$

4.3.2 • Parametrization by the worst and the best states

Let V be a value function. Then let $(s_{min}, s_{max}) \in S^2$, $s_{min} \neq s_{max}$, such as:

$$s_{min} \in \operatorname*{argmin}_{s \in S} V_s$$

$$s_{max} \in \operatorname*{argmax}_{s \in S} V_s$$

Then (7) is equivalent to:
$\forall s \in S$:

$$\begin{cases} V_{s_{min}} \leq \frac{V_s}{\gamma} + m_s \\ \frac{V_s}{\gamma} + M_s \leq V_{s_{max}} \\ V_{s_{min}} \leq V_s \leq V_{s_{max}} \end{cases}$$

In particular, it is verified by $V_{s_{min}}$ and $V_{s_{max}}$, so:

$$\begin{cases} V_{s_{min}} \geq \frac{-\gamma}{1-\gamma} m_{s_{min}} \\ V_{s_{max}} \leq \frac{-\gamma}{1-\gamma} M_{s_{max}} \end{cases}$$

Definition 26. *We define:*

$$m = \min_{s \in S} m_s$$

$$M = \max_{s \in S} M_s$$

$$D = \max_{s \in S} M_s - m_s$$

Then (7) is possible if and only if:

$$\begin{cases} V_{s_{min}} \geq \frac{-\gamma}{1-\gamma} m_{s_{min}} \\ V_{s_{max}} \leq \frac{-\gamma}{1-\gamma} M_{s_{max}} \\ V_{s_{min}} + D \leq V_{s_{max}} \\ V_{s_{min}} \leq \frac{V_{s_{max}}}{\gamma} + m \\ V_{s_{min}} \leq \gamma(V_{s_{max}} - M) \end{cases}$$

is verified by definition of m.

Theorem 10. *Let $(s_{min}, s_{max}) \in S^2$, there exists an admissible $V \in \mathbb{R}^S$ that verifies $V_{s_{min}} = \min_{s \in S} V_s$ and $V_{s_{max}} = \max_{s \in S} V_s$ if and only if:*

$$m_{s_{min}} \geq \max[M_{s_{max}} + \frac{1-\gamma}{\gamma} D, -\frac{1}{\gamma} M_{s_{max}} - \frac{1-\gamma}{\gamma} m, \gamma M_{s_{max}} + \frac{1}{1-\gamma} M]$$

If this condition holds we say that (s_{min}, s_{max}) is admissible.

Theorem 11. *If (s_{min}, s_{max}) is admissible then the admissible value functions associated with the tuple (s_{min}, s_{max}) form the polytope $\mathcal{V}(s_{min}, s_{max})$ defined by:*

$$\begin{cases} -\frac{\gamma}{1-\gamma} m_{s_{min}} \leq V_{s_{min}} \leq \min[-\frac{\gamma}{1-\gamma} M_{s_{max}} - D, \frac{1}{1-\gamma} M_{s_{max}} + m, -\frac{\gamma^2}{1-\gamma} M_{s_{max}} - \gamma M] \\ \max[V_{s_{min}} + D, \gamma(V_{s_{min}} - m), \frac{1}{\gamma} V_{s_{min}} + M] \leq V_{s_{max}} \leq -\frac{\gamma}{1-\gamma} M_{s_{max}} \\ \forall s \in S/\{s_{min}, s_{max}\}, \max[V_{s_{min}}, \gamma(V_{s_{min}} - m_s)] \leq V_s \leq \min[V_{s_{max}}, \gamma(V_{s_{max}} - M_s)] \end{cases}$$

4.4 Set of admissible models

Theorem 12. *The set of admissible value functions $\mathcal{V}$ is given by:*

$$\mathcal{V} = \bigcup_{(s_{min}, s_{max}) \in S^2, s_{min} \neq s_{max}} \mathcal{V}(s_{min}, s_{max}) \tag{8}$$

Where $\mathcal{V}(s_{min}, s_{max})$ is not empty if and only if

$$m_{s_{min}} \geq \max[M_{s_{max}} + \frac{1-\gamma}{\gamma} D, -\frac{1}{\gamma} M_{s_{max}} - \frac{1-\gamma}{\gamma} m, \gamma M_{s_{max}} + \frac{1}{1-\gamma} M]$$

Theorem 13. *The set of admissible transition matrices $\mathcal{P}$ is given by:*

$$\mathcal{P} = \bigcup_{V \in \mathcal{V}} \mathcal{P}(V) \tag{9}$$

where:

$$\forall V \in \mathcal{V}, \mathcal{P}(V) = \{P \in \mathcal{B} | \forall (s,a) \in S \times A, V_s + \frac{1}{\beta}(ln(\pi_s(a)) - H_s) = \sum_{s' \in S} P_{s,a}(s')(R_{s,a,s'} + \gamma V_{s'})\}$$

Remark that $\forall (V, V') \in \mathcal{V}^2, \mathcal{P}(V) \cap \mathcal{P}(V') \neq \emptyset \Rightarrow V = V'$ by definition of the value function of a transition matrix.

4.5 Choice of the estimated transition matrix

We go back to the fundamental problem. We know (R, γ, β) and we observe the policy π. Which transition matrix $P \in \mathcal{P}$ should we use to model the behavior of the agent?

4.5.1 • Closest transition matrix

Let P_0 be a transition matrix. In the case of the Helper-AI, P_0 is the real transition matrix of the system. We make the assumption that the transition matrix used by the agent is close to P_0. Thus we model the agent by the closest admissible matrix. But which distance should we use?

In fact, $\mathcal{P}$ is not convex. It is logical to use a kind of projection that could be associated with the parametrization by the value functions. We will proceed in two steps.

First, we compute the projection V_P of the value function V_0 of P_0 over $\mathcal{V}$. Then we project P_0 over $\mathcal{P}(V_P)$ and get the estimated admissible matrix P_P. For this two projections, we use the L2-norms associated with the canonical scalar products of $\mathbb{R}^S$ and $\mathcal{M}_{S \times A, S}(\mathbb{R})$.

4.5.2 • Prior on the value function

Assume that the prior on the value function is a set of linear constraints that reduce the authorized value function to a polytope $\mathcal{B}_{prior}$. Since for each tuple (s_{min}, s_{max}), $\mathcal{V}(s_{min}, s_{max})$ is a polytope, a simplexe algorithm allows to check whether $\mathcal{V}(s_{min}, s_{max}) \cap \mathcal{B}_{prior}$ is empty, like in the two phases method.

A linear prior on the transition matrix reduces in the same way the set of admissible matrices once the value function has been chosen. Nevertheless the prior on the transition matrix may reduce the set of admissible value functions. When the constraint on the transition matrix can be translated into a linear constraint on the value function, we can refer to the previous paragraph.

4.5.3 • Partially known transition matrix

If we know the transition matrix $P_{s,a}$ for a pair $(s,a) \in S \times A$, we get an additional constraint for the definition of $\mathcal{V}$:

$$\sum_{s' \in S} P_{s,a}(s')V_{s'} = \frac{1}{\gamma}(V_s + \frac{1}{b}(ln(\pi_s(a)) - H_s) - R_{s,a})$$

4.5.4 • Lower bound on the transition matrix

The uncertainty on the transition matrix may be limited. Here we assume that we have a function $f \in [0,1]^{S \times A \times S}$ such as:

$$\forall (s,a,s') \in S \times A \times S, P_{s,a}(s') \geq f(s,a,s')$$

where:

$$\forall (s,a) \in S \times A, \sum_{s' \in S} f(s,a,s') < 1$$

or we are in the precedent case.

Then we define:

$$\forall (s,a,s') \in S \times A \times S, p_{s,a}(s') = \frac{P_{s,a}(s') - f(s,a,s')}{1 - \sum_{s' \in S} f(s,a,s')} \in [0,1]$$

It verifies:

$$\forall (s,a) \in S \times A, \sum_{s' \in S} p_{s,a}(s') = 1$$

Then (5) becomes:

$$\exists p \in \mathcal{B}, \forall (s,a) \in S \times A, \sum_{s' \in S} p_{s,a}(s')V_{s'} = \frac{V_s + \frac{1}{b}(ln(\pi_s(a)) - H_s) - R_{s,a}}{\gamma(1 - \sum_{s' \in S} f(s,a,s'))} - \sum_{s' \in S} \frac{f(s,a,s')}{(1 - \sum_{s' \in S} f(s,a,s'))} V'_s \tag{10}$$

That is equivalent to:

$$\forall (s,a) \in S \times A, V_{s_{min}} \leq \frac{V_s + \frac{1}{b}(ln(\pi_s(a)) - H_s) - R_{s,a}}{\gamma(1 - \sum_{s' \in S} f(s,a,s')} - \sum_{s' \in S} \frac{f(s,a,s')}{(1 - \sum_{s' \in S} f(s,a,s'))} V'_s \leq V_{s_{max}} \tag{11}$$

which are new linear constraints on the value function.

4.5.5 • Upper bound on the transition matrix

Here we assume that we have a function $g \in [0,1]^{S \times A \times S}$ such as:

$$\forall (s,a,s') \in S \times A \times S, P_{s,a}(s') \leq g(s,a,s')$$

where:

$$\forall (s,a) \in S \times A, \sum_{s' \in S} g(s,a,s') > 1$$

We get:

$$\forall (s,a) \in S \times A, V_{s_{min}} \leq -\frac{V_s + \frac{1}{b}(ln(\pi_s(a)) - H_s) - R_{s,a}}{\gamma(\sum_{s' \in S} g(s,a,s') - 1)} + \sum_{s' \in S} \frac{g(s,a,s')}{(\sum_{s' \in S} g(s,a,s')) - 1} V'_s \leq V_{s_{max}} \tag{12}$$

4.6 Conclusion

In a single player MDP, the observation of a softmax policy of the agent of a known parameter allows an exact description of the set of admissible transition matrices that could explain this behavior. The set of admissible value functions is the union of at most $S \times (S-1)$ polytopes. For each admissible value function, there is an associated polytope of admissible transition matrices.

5
THE HELPER-AI FRAMEWORK

We have solved the problem for a single player MDP. Now we want to use this result in the Helper-AI context. We show that our solution allows to find an admissible model consistent with a reduce number of observations. For more observations, our previous work can not be used and we need to develop other methods. We define some loss measures that depend on the difference between the behaviors predicted by the models and the real ones. We then minimize this losses thanks to gradient descents.

5.1 SEVERAL OBSERVATIONS

5.1.1 • APPLICATION OF THE RESULTS FOR THE SINGLE PLAYER MDP

Let (π_{AI}, π_H) be one observation. Then we can apply the precedent result to the single player MDP $(P^{\pi_{AI}}, R^{\pi_{AI}}, \gamma)$, the observed policy is π_H.

Then $\mathcal{V}_{\pi_{AI}}$ is defined exactly like $\mathcal{V}$ in the last part and:

Theorem 14. *The set of admissible matrices for the observation (π_{AI}, π_H) is:*

$$\mathcal{P}_{\pi_{AI}} = \bigcup_{V \in \mathcal{V}_{\pi_{AI}}} \mathcal{P}_{\pi_{AI}}(V)$$

where:

$$\forall V \in \mathcal{V}_{\pi_{AI}}, \mathcal{P}_{\pi_{AI}}(V) = \{P \in \mathcal{B} | P^{\pi_{AI}} \in \mathcal{P}(V)\}$$

5.1.2 • INDEPENDENT OBSERVATIONS

Theorem 15. *For an obervation set $\mathcal{O} = \{(\pi^1_{AI}, \pi^1_H), \ldots (\pi^n_{AI}, \pi^n_H)\}$, the set of admissible transition matrices is given by:*

$$\mathcal{P}_{\mathcal{O}} = \bigcap_{(\pi_{AI}, \pi_H) \in \mathcal{O}} \mathcal{P}_{\pi_{AI}} = \bigcup_{(V^1, \ldots V^n) \in \mathcal{V}_{\pi^\infty_{\mathcal{AI}}} \times \ldots \times \mathcal{V}_{\pi^\backslash_{\mathcal{AI}}}} \bigcap_{i=1}^{n} \mathcal{P}_{\pi^i_{AI}}(V^i)$$

Definition 27. *Two policies π and π' are linearly independent if:*

$$\nexists (s,a) \in S \times A, \pi_s(a) > 0 \wedge \pi'_s(a) > 0$$

Theorem 16. *If $\mathcal{O} = \{(\pi^1_{AI}, \pi^1_H), \ldots (\pi^n_{AI}, \pi^n_H)\}$, if for all $(i,j) \in \{1, \ldots n\}$ if $i \neq j$, π_{AI} and π'_{AI} are linearly independent, then:*

$$\forall (V^1, \ldots V^n) \in \mathcal{V}_{\pi^\infty_{\mathcal{AI}}} \times \ldots \times \mathcal{V}_{\pi^\backslash_{\mathcal{AI}}}, \bigcap_{i=1}^{n} \mathcal{P}_{\pi^i_{AI}}(V^i) \neq \emptyset$$

This theorem allows, for observations generated from a linearly independent family of policies of the AI, to project a prior belief for the transition matrix over the set of admissible transition matrices.

Proof. If π_{AI}^1 and π_{AI}^2 are linearly independent, then $P^{\pi_{AI}^1}$ and $P^{\pi_{AI}^2}$ are generated from different components of P. A constraint over $P^{\pi_{AI}^1}$ lets $P^{\pi_{AI}^2}$ totally free. □

A family of linearly independent policies of the AI contains at most $|A^{AI}|$ elements. An example of such a family is $\{\pi_{AI}^a | a \in A^{AI}\}$ where for $a \in A^{AI}$, π_{AI}^a is the deterministic policy that always plays action a.

5.1.3 • Larger number of observations

If the policies of the AI are not linearly independent, then it is difficult to find an element of $\mathcal{P}_{\mathcal{O}}$ since 16 does not hold.

Moreover this setting corresponds to the practical case. Indeed, to find an informed policy, the AI tries several stochastic strategies and quickly generates a non linearly independent family of policies.

We do not have a general description of the admissible set of matrices in this setting. In the following parts, we expose some practical methods to find an admissible transition matrix for a non linearly independent family of observations.

5.2 Error measure

Since we provide algorithmic heuristics to find an admissible transition matrix, we must be able to evaluate the quality of a matrix. With this objective, we go back to the probabilistic methods we used to choose a model among a finite set.

The likelihood $\mathcal{L}$ of a transition matrix P_0 given a set of observations: $O = \{(s_i, a_i) | i \in [1, n]\} \subset (S \times A)^n$ (n is the number of observations) is given by:

$$\mathcal{L} = \prod_{i \in [1,n]} \pi_{s_i}^{P_0}(a_i)$$

where π^{P_0} is the softmax policy of the human for the transition matrix P_0.

In fact, we have:

$$\mathcal{L} = \prod_{s \in S} [\prod_{a \in A} \pi_s^{P_0}(a)^{\hat{\pi}_s(a)}]^{n\hat{\pi}(s)}$$

where $\hat{\pi}(s)$ is the empirical frequency of the state s in O and $\hat{\pi}_s(a)$ the empirical probability of the action a in state s. We can assume that $\hat{\pi}(s)$ is constant over S and that we know perfectly $\hat{\pi}_s(a)$ for all s and a. Then the log-likelihood is given by:

$$l = \frac{n}{|S|} \sum_{(s,a) \in S \times A} \hat{\pi}_s(a) \log \pi_s^{P_0}(a)$$

In reality we directly observe $\hat{\pi}$. As a consequence we define a natural error.

Definition 28. *For a finite observations set $\mathcal{O} = \{(\pi_{AI}^1, \pi_H^1), \dots (\pi_{AI}^n, \pi_H^n)\}$, the negative log-likelihood of the transition matrix $P \in \mathcal{M}_{S \times A^{AI} \times A^H, S}$ is:*

$$l(P) = \sum_{i=1}^{n} \sum_{(s,a) \in S \times A} \pi_{H,s}^i(a) \log \frac{\pi_{H,s}^i(a)}{\pi_s^{P^{\pi_{AI}^i}}(a)}$$

where $\pi^{P^{\pi_{AI}^i}}$ is the softmax policy of parameter β for the MDP $(P^{\pi_{AI}^i}, R^{\pi_{AI}^i}, \gamma)$.

This negative log-likelihood is the sum of the negative Kullback-Leibler divergence between the distributions $\pi^i_{H,s}$ and $\pi_s^{P^{\pi^i_{AI}}}$.

Property 1.

$$\forall P \in \mathcal{M}_{S \times A^{AI} \times A^H, S}, l(P) \geq 0$$

Property 2. $l(P) = 0$ *if and only if* P *is admissible.*

5.3 Successive projections

For each observation, we can compute the associated set of admissible transition matrices. Some projection of one given transition matrix over this set can be computed, taking the projection of the associated value function over the set of admissible value functions and then taking the projection of the matrix over the set of transition matrices associated with the projected value function.

The less clever way to find one matrix in the intersection of the sets of admissible matrices associated with each observation is to take one arbitrary matrix and to perform successive projections over this sets until the distance to all sets seems acceptable.

Unhappily in most of our experiments, this method does not converge and does not reduce significantly the negative log-likelihood.

5.4 Gradient descent

We want to minimize the negative log-likelihood over the set of transition matrices. Even if this function is not convex, we use a gradient descent algorithm.

Here we give some partial derivative functions that are useful to deploy this method.

Property 3.

$$\forall i \in \{1, \dots n\}, \forall (s,a) \in S \times A^H, \frac{\partial l}{\partial \pi_s^{P^{\pi^i_{AI}}}(a)} = -\frac{\pi^i_{H,s}(a)}{\pi_s^{P^{\pi^i_{AI}}}(a)}$$

$$\forall i \in \{1, \dots n\}, \forall (s,a,a') \in S \times A^H \times A^H, \frac{\partial \pi_s^{P^{\pi^i_{AI}}}(a)}{\partial Q_s^{P^{\pi^i_{AI}}}(a')} = \beta \pi_s^{P^{\pi^i_{AI}}}(a)(\mathbb{1}_{a'=a} - \pi_s^{P^{\pi^i_{AI}}}(a'))$$

$$\forall i \in \{1, \dots n\}, \forall (s,a,s') \in S \times A^H \times S, \frac{\partial Q_s^{P^{\pi^i_{AI}}}(a)}{\partial P_{s,a}^{\pi^i_{AI}}(s')} = R_{s,a,s'}^{\pi^i_{AI}} + \gamma V_{s'}^{P^{\pi^i_{AI}}}$$

$$\forall i \in \{1, \dots n\}, \forall (s,a,s') \in S \times A^H \times S, \frac{\partial Q_s^{P^{\pi^i_{AI}}}(a)}{\partial V_{s'}^{P^{\pi^i_{AI}}}} = \gamma P_{s,a}^{\pi^i_{AI}}(s')$$

If π_i^* *is an optimal policy for* $(P^{\pi^i_{AI}}, R^{\pi^i_{AI}}, \gamma)$*:*

$$\frac{\partial V^{P^{\pi^i_{AI}}}}{\partial P^{(\pi^i_{AI}, \pi_i^*)}} = -(I - \gamma P^{(\pi^i_{AI}, \pi_i^*)})^{-1} P^{(\pi^i_{AI}, \pi_i^*)} V^{P^{\pi^i_{AI}}}$$

$$\forall i \in \{1, \dots n\}, \forall (s, a_0, a, s') \in S \times A^{AI} \times A^{H} \times S, \frac{\partial P_{s,a}^{\pi^i_{AI}}(s')}{\partial P_{s,a_0,a}(s')} = \pi^i_{AI,s}(a_0)$$

$$\forall i \in \{1, \dots n\}, \forall (s, a_0, a, s') \in S \times A^{AI} \times A^{H} \times S, \frac{\partial P_s^{(\pi^i_{AI},\pi^*_i)}(s')}{\partial P_{s,a_0,a}(s')} = \pi^i_{AI,s}(a_0)\pi^*_{i,s}(a)$$

If P is the current transition matrix of the algorithm, we use the following notation for the gradient:

$$\nabla = \frac{\partial l}{\partial P}$$

Algorithm 1. *Gradient descent for the negative log-likelihood.*

1. *Initialize the transition matrix P and the total number of iterations n. Initialize the iteration counter $i = 0$.*
2. *Compute the associated softmax policies for every policy of the AI thanks to a value-iteration algorithm.*
3. *Compute the loss l.*
4. *Compute the gradient ∇.*
5. *Update the transition matrix: $P = P - \eta(i)\nabla$.*
6. *Project the matrix over the set of transition matrices $P = proj(P)$.*
7. *$i = i + 1$*
8. *If $i < n$, go back to step 2. Otherwise return the best transition matrix encountered.*

The parameters of the algorithm are the total number of iterations n and the size of the steps η that may depend on the iteration number.

Moreover we have to define the function *proj*. Indeed after the update of the matrix P, it does not belong anymore to the set of transition matrices. We chose the following projection algorithm.

Algorithm 2. *Projection over the set of transition matrices.*

1. *Take P as an input and the parameter $\epsilon > 0$.*
2. *For all $(s, a, b) \in S \times A^{AI} \times A^{H}$, do:*
   (a) *For all $s' \in S$, $P_{s,a,b}(s') = \max(P_{s,a,b}(s'), \epsilon)$.*
   (b) *$\mathcal{S} = \sum_{s' \in S} P_{s,a,b}(s')$*
   (c) *For all $s' \in S$, $P_{s,a,b}(s') = \frac{P_{s,a,b}(s')}{\mathcal{S}}$.*
3. *Return P.*

To optimize the algorithm, we use several improvements.

1. a random noise is added to the gradient to accelerate the exploration. Its amplitude is a fraction of the gradient's norm, controlled by a parameter T called temperature.

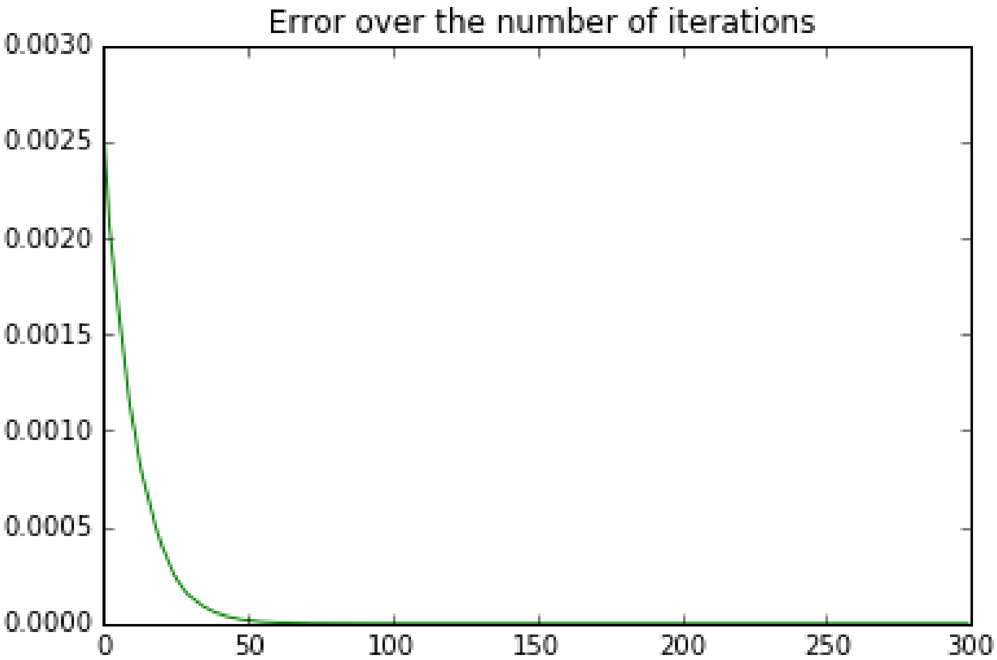


Figure 7: Gradient descent: negative log-likelihood over the number of observations. Data generated for $\#S = \#A^{AI} = \#A^H = \#\mathcal{P} = 2$ and $\#\mathcal{O} = 5$.

2. if the precedent iteration has lead to a significant progress in term of error, the direction of this gradient is preferred. Precisely, there is an inertia parameter $\alpha > 0$ and at iteration i: $\nabla_i = \nabla + \alpha(l_{i-2} - l_{i-1})\nabla_{i-1}$, where ∇_i and l_i are the gradient and the loss at iteration i and ∇ is the gradient at the current point.

When we optimize the parameter for one particular instance of the problem, we generally achieve a good result. For the experiment of figure 7, $T = 0.05$,$\alpha = 1$ and $\epsilon(i) = \frac{300-i}{3000}$.

Nevertheless, we were not able to find parameters that would give a satisfying result for every instance of the problem. This is disappointing since we aim at creating an online algorithm.

5.5 L2 ERROR

Since we did not find an universal parametrization for the previous gradient descent, we looked for another error function that would represent the same reality: the distance from our model to the set of admissible transition matrices according to observations.

Theorem 17. *For one observation (π_{AI}, π_H), the set of admissible matrices is given by:*

$$\mathcal{P} = \{P \in \mathcal{B} | \forall(s, a, a') \in S \times A^H \times A^H, Q_s^{P^{\pi_{AI}}}(a) - Q_s^{P^{\pi_{AI}}}(a') = \frac{1}{\beta}\log(\frac{\pi_{H,s}(a)}{\pi_{H,s}(a')})\}$$

Proof. We have seen that, if P is admissible then:

$$\forall(s, a) \in S \times A^H, Q_s^{P^{\pi_{AI}}}(a) = \frac{1}{\beta}(\log \pi_{H,s}(a) - \log Z_s)$$

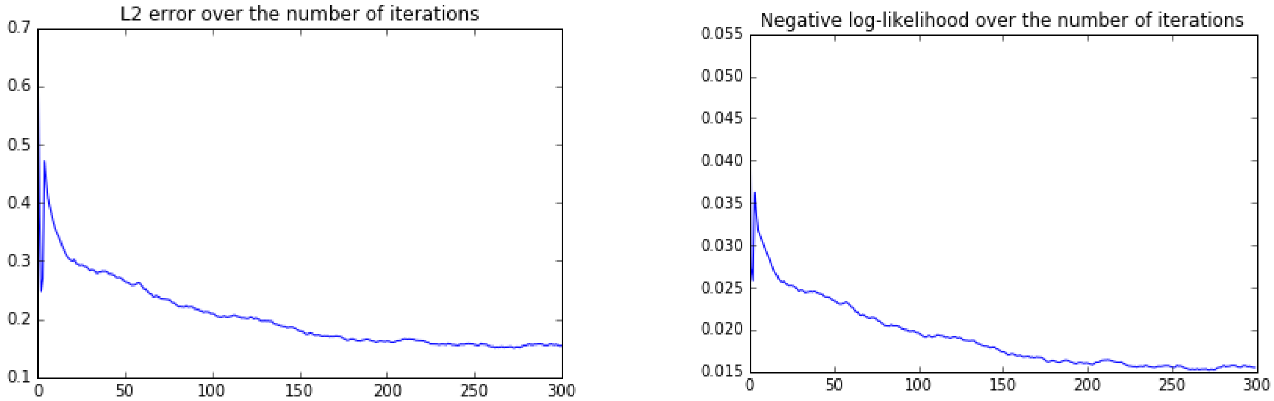


Figure 8: Gradient descent for L2 error. Data generated for $\#S = \#A^{AI} = \#A^H = \#\mathcal{P} = 2$ and $\#\mathcal{O} = 5$.

Taking the difference we get the condition of the theorem.

Conversely, assume that:

$$\forall (s,a,a') \in S \times A^H \times A^H, Q_s^{P^{\pi AI}}(a) - Q_s^{P^{\pi AI}}(a') = \frac{1}{\beta}\log(\frac{\pi_{H,s}(a)}{\pi_{H,s}(a')})$$

Then:

$$\forall s \in S, \exists Z_s \in \mathbb{R}_+^*, \forall a \in A^H, Q_s^{P^{\pi AI}}(a) = \frac{1}{\beta}(\log \pi_{H,s}(a) - \log Z_s)$$

and:

$$\forall (s,a) \in S \times A^H, \pi_{H,s}(a) = \frac{e^{\beta Q_{s,a}^{P^{\pi AI}}}}{\sum_{a' \in A^H} e^{\beta Q_s^{P^{\pi AI}}(a')}}$$

i.e. P is admissible. □

This theorem motivates the following definition.

Definition 29. *For the observation set $\mathcal{O}$, the L2 error of the transition matrix P is given by: item*

Since $\forall P \in \mathcal{B}, l_2(P) \geq 0$, $\mathcal{P} = \operatorname{argmin} l_2(P)$. We use a new gradient descent algorithm to minimize this error. Nevertheless, we plot the negative log-likelihood of the transition matrix.

Property 4.

$$\forall i \in \{1, \dots n\}, \forall (s,a) \in S \times A^H, \frac{\partial l_2}{\partial Q_s^{P^{\pi^i_{AI}}}(a)} = \sum_{a' \in A^H} Q_s^{P^{\pi^i_{AI}}}(a) - Q_s^{P^{\pi^i_{AI}}}(a') - \frac{1}{\beta}\log\frac{\pi^i_{H,s}(a)}{\pi^i_{H,s}(a')}$$

All other useful derivative functions have been given in the previous part.

The experiments (figure 8) show that a decrease of L2 norm is generally associated with a decrease in term of negative log-likelihood. Concretely it gives us another method to look for an admissible transition matrix when the first gradient descent gets stuck in a local minimum.

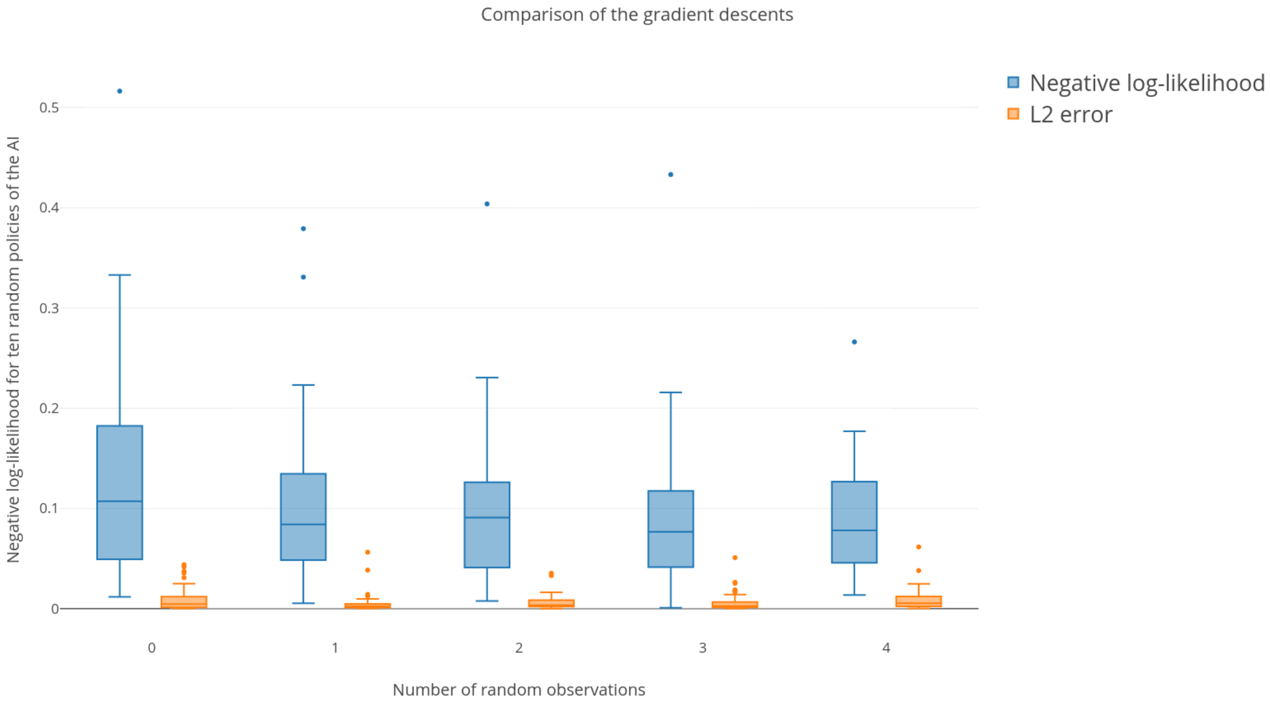


Figure 9: Data generated for 250 MDPs with $\#S = \#A^{AI} = \#A^{H} = \#\mathcal{P} = 2$. The observations are constituted by the four deterministic policies of the AI and a fixed number of random policies (X axis).

5.6 Comparison of the methods

5.6.1 • Gradient descents

Since we want to use the algorithm in an online application, we have to set definitely the parameters of gradient descents. Here (9) we compare the estimated transition matrices in term of their generalization power. We measure the sum of the negative log-likelihood of the estimated transition matrices for new random observations.

It appears clearly that paradoxically, the L2 gradient descent seems more efficient at producing an effective estimation of the transition matrix.

5.6.2 • Gradient descent and probabilistic method

Each step of the gradient descent algorithm is constituted by a value iteration algorithm. As a consequence, for the same computing power, the probabilistic method that chooses the best model among a randomly generated set must allowed to test n models where n is the number of iterations of the gradient descent.

To compare both methods, we use the same metrics as previously.

The experiment (figure 10) shows that the L2 gradient descent is at least as efficient as the probabilistic method. Nevertheless, the efficiency of the gradient descent can be improved in a practical case when the parameters of the algorithm can be optimized over the data.

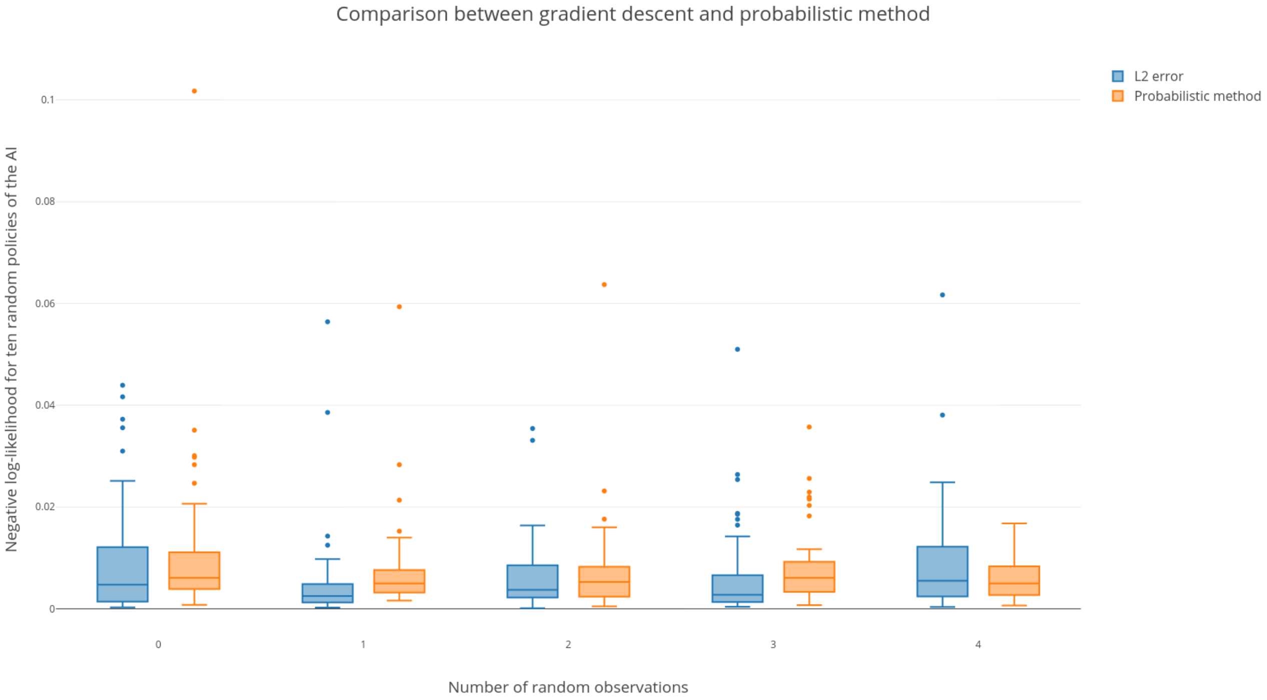


Figure 10: Data generated for 250 MDPs with $\#S = \#A^{AI} = \#A^{H} = \#\mathcal{P} = 2$. The observations are constituted by the four deterministic policies of the AI and a fixed number of random policies (X axis).

5.7 Conclusion

For a reduced number of observations, the parametrization of the set of admissible transition matrices allows to find a matrix in this set. Nevertheless, when the family of policies of the AI in the observations is not linearly independent, the consistency conditions are not explicit.

As a consequence, we had to design an heuristic algorithm that tries to find an admissible transition matrix. We showed that a standardized version of this algorithm is as efficient as a probabilistic method that only tests a large number of models.

6
EQUIVALENCE CLASSES AND INDISTINGUISHABILITY

Our gradient descents are efficient but generally fail. They don't produce a model that is perfectly consistent with all observations. In order to explain why gradient descents get stuck in local minima, we analyze the evolution of the set of admissible models when we increase the number of observations. We empirically show that we can construct indistinguishable curves with models that would produce the same behavior for any policy of the AI. As a consequence, the real interesting object is not the model of the human but its indistinguishability class. That's why we propose to use a classification algorithm to solve the prediction problem.

6.1 Formalization

Definition 30. *Given a softmax parameter β, a discount factor γ, a rewards vector R and a policy of the AI π_{AI}, two transition matrices P and P' are indistinguishable for π_{AI} if they produce the same exact policy of the human.*

We use the notation: $C_I(P, \pi_{AI}) = \{P' | P \text{ and } P' \text{ are indistinguishable for } \pi_{AI}\}$. $C_I(P, \pi_{AI})$ is an equivalence class, called the indistinguishability class of P for π_{AI}.

In our previous work, we provided a parametric description of indistinguishability classes.

Definition 31. *Given a softmax parameter β, a discount factor γ and a rewards vector R, two transition matrices P and P' are equivalent if they are indistinguishable for every policy of the AI.*

We use the notation: $C_E(P) = \cap_{\pi_{AI}} C_I(P, \pi_{AI})$. This set is called the equivalence class of P.

In the Helper-AI framework, two questions are of particular interest:

1. What is the shape of $C_E(P)$? If it is reduced to the matrix P then we can hope to estimate the real belief of the human. Otherwise we have to accept any member of the class.

2. How fast does the intersection of several indistinguishability classes converge to the equivalence class? In particular, can we have $\bigcap_{\pi_{AI} \in \mathcal{F}} C_I(P, \pi_{AI}) = C_E(P)$ for $\mathbb{F}$ a finite family of policies of the AI?

6.2 Construction of an equivalence curve

We answer to this questions locally. Starting from an arbitrary transition matrix P, we construct an approximation of $C_E(P)$ in a neighborhood of P. More precisely, we show experimentally that there exists a neighborhood of P such that its intersection with $C_E(P)$ is contained in an affine line.

Let P' be a transition matrix in a neighborhood of P in $\mathcal{B}$. We will write:

$$P' = P + dP$$

The condition $P' \in \mathcal{B}$ imposes linear constraints over dP.

Moreover, let e be the vector of $\mathcal{R}^{A^H}$ such that $\forall a \in A^H, e_a = 1$. The following property results from theorem 17.

Property 5. *Let π_{AI} be a policy of the AI and $P \in \mathcal{B}$ a transition matrix.*

$$P' \in C_I(P, \pi_{AI}) \Leftrightarrow \forall s \in S, Q_s^{P\pi_{AI}} - Q_s^{P'\pi_{AI}} \in vect(e)$$

where for $s \in S$, $Q_s^{P\pi_{AI}}$ is the vector of $\mathbb{R}^{A^H}$ of the Q-values of each action in state s for the MDP $(P^{\pi_{AI}}, R^{\pi_{AI}}, \gamma)$.

Property 6. *If for all $s \in S$, $|\operatorname{argmax}_{a\in A^H} Q_s^{P\pi_{AI}}(a)| = 1$, then for all $s \in S$, the application $P \to Q_s^{P\pi_{AI}}$ is infinitely differentiable at the point P.*

Proof. If the condition $\forall s \in S, |\operatorname{argmax}_{a\in A^H} Q_s^{P\pi_{AI}}(a)| = 1$, then there exists only one optimal policy for the human for the transition matrix P. This policy π^* is deterministic and is constant over a neighborhood of P in $\mathcal{B}$.

Since $V = (I - \gamma P^{\pi_{AI},\pi^*})^{-1} R^{\pi_{AI},\pi^*}$, $P \to V$ is infinitely differentiable in P.

Let $(s, a) \in S \times A^H$, $Q_s^{P\pi_{AI}}(a) = R_{s,a}^{\pi_{AI}} + \gamma P_{s,a}^{\pi_{AI}} V$, then $P \to Q_s^{P\pi_{AI}}(a)$ is infinitely differentiable in P. □

The condition of the previous theorem are verified in $\mathcal{B}$, except in a set of measure 0. We use the notation $\Delta_{P,\pi_{AI},s}$ for the differentiate of $P \to Q_s^{P\pi_{AI}}$ at point P.

Property 7. *There exists a neighborhood N of P such that:*

$$P + dP \in C_I(P, \pi_{AI}) \cap N \Rightarrow \forall s \in S, \Delta_{P,\pi_{AI},s,a}(dP) \in vect(e)$$

This gives new linear constraints over dP. dP is an element of the vector space $\mathcal{M}_{S\times A^{AI}\times A^H,S}(\mathbb{R})$ of finite dimension. By increasing the number of observations, we want to forbid every direction for dP. We would show that P is an isolated point of $C_E(P)$ in $\mathcal{M}_{S\times A^{AI}\times A^H,S}(\mathbb{R})$.

The simulations (figures 11, 12, 13, 14) show that the first order conditions always let a few directions free for dP. Does it mean that $C_E(P)$ is not reduced to P in a neighborhood of P?

In order to check that hypothesis, we try to build an approximation of a curve of elements of $C_E(P)$ thanks to the free directions exhibited by our experiments.

Algorithm 3. *Construction of an equivalence curve.*

1. *Initialize the curve to a transition matrix $P \in \mathbb{B}$.*
2. *Take as an input the parameter n. This parameter is the number of needed observation before we consider that we have explored all forbidden directions. n depends on the dimension of the MDP.*
3. *Take as input the step size η and the number of steps N.*
4. *Repeat N times the following process.*
    (a) *Generate n different policies for the AI.*
    (b) *Observe the answer of the human to this policies.*
    (c) *Compute the list of forbidden directions for this observation.*
    (d) *There always exist an authorized direction (orthogonal to the vector space of forbidden directions), choose dP in the vector space of authorized directions.*
    (e) *$P = P + \eta dP$*

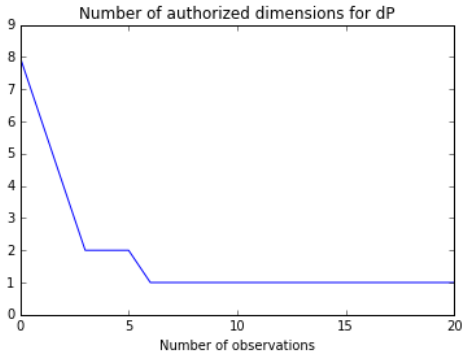


Figure 11: Search for forbidden directions for $C_E(P)$ in a neighborhood of P. $\#S = \#A^{AI} = \#A^H = 2$

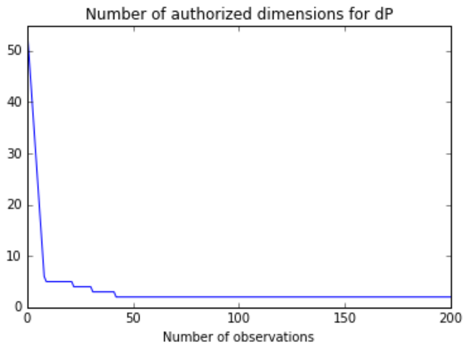


Figure 12: Search for forbidden directions for $C_E(P)$ in a neighborhood of $P.\#S = \#A^{AI} = \#A^H = 3$

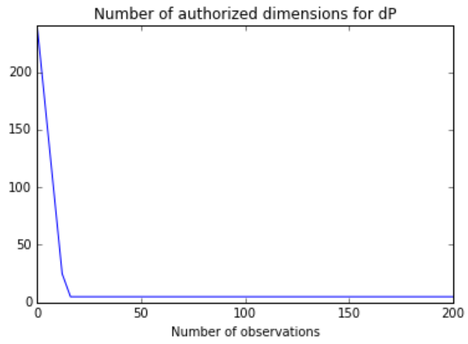


Figure 13: Search for forbidden directions for $C_E(P)$ in a neighborhood of P. $\#S = 6, \#A^{AI} = 2, \#A^H = 4$

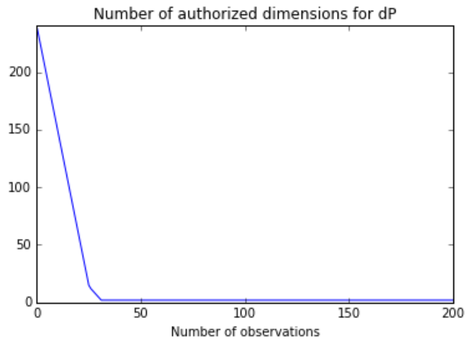


Figure 14: Search for forbidden directions for $C_E(P)$ in a neighborhood of $P.\#S = 3, \#A^{AI} = 10, \#A^H = 4$

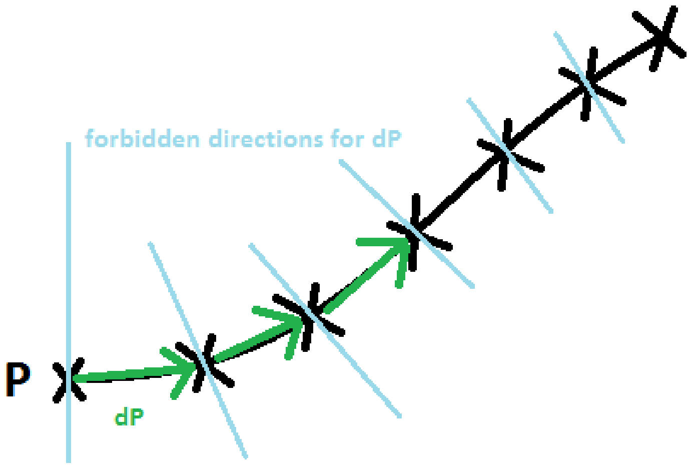


Figure 15: Idea of the construction of an equivalence curve

(f) Add P to the curve.

We show that this method allows to construct transition matrices at a macroscopic distance from each other but that are close to the same equivalence class. We measure the distance to the equivalence class as the sum of the negative log-likelihood for 100 randomly generated policies of the AI. This sum remains below 10^{-5}, for two matrices distant from a distance of one (measured in term of L2 norm).

We partially answer the initials questions. For a given transition matrix P, $C_E(P)$ is not reduced to P. We can construct a continuous curve in $C_E(P)$. Locally, the equivalence class can be approximated by a finite number of indistinguishability classes.

6.3 Regression point of view

This conclusion shows that we can not expect to solve the estimating problem. The equivalence class contains an infinite number of transition matrices. Even if the transition matrix is the model that is used by the human, the aim of the estimation should be the equivalence class (or the intersection of indistinguishability classes for a finite number of observations).

In this part we fix the number n of observations. n is assumed to be large enough to give a good approximation of equivalence classes. We fix the n policies of the AI.

For a given transition matrix P, if we observe the answer policies of the human for the n policies of the AI, we can estimate the equivalence class of P thanks to a decision tree. Indeed, we can generate an arbitrary large dataset of transition matrices and compute the answer policies for every element of this dataset.

Since the aim of our work is the prediction of the policy of the human for a given policy of the AI, we use a regression tree. The features are the n answer policies of the human and the policy of the AI for which we want to predict the answer of the human.

To achieve better results, we finally use a random forest of regression trees. This method has two advantages. First the training of the regressor is made before the prediction. That authorizes an online algorithm. Moreover, the softmax parameter β is not necessary since the information it carries is already contained in the features (in the n answer policies of the human). For the same computing power, the results are a lot worse than the gradient descent methods but we assume the computation time is less important for the regression method since it is not an online algorithm.

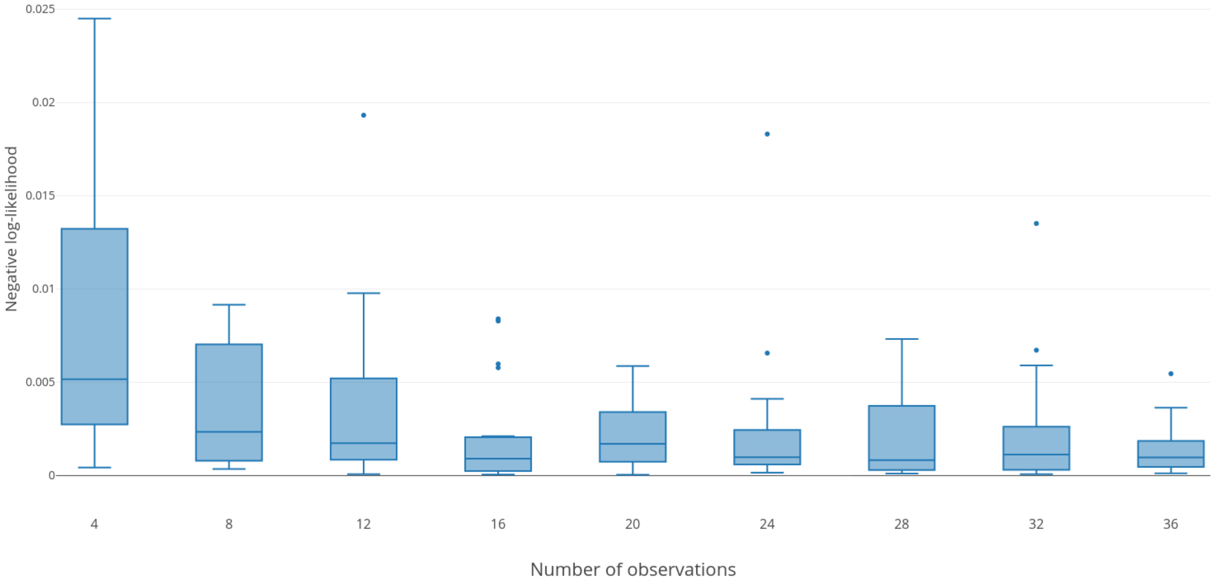


Figure 16: Error of the random forest over the number of observations in the features. Data generated for 270 MDPs of dimension $|S| = |A^{AI}| = |A^{H}| = 2$. The training set contains 10 000 instances.

CONCLUSION

When talking about artificial intelligence, one quickly comes to the idea of building an intelligent system that would help human beings to carry out a given task. That is the Helper-AI problem. We formalize it as a two-player decision game.

Nevertheless when the human being and the artificial system do not agree on their model of the world, they achieve a subobtimal cumulative reward. Christos Dimitrakakis and his team in the EconCS group at Harvard University have measured the loss that is created by a divergence between the two models.

But if the artificial intelligence is able to guess the model used by the human, it can predicts his future behavior and then achieve a better cumulative reward. We have been working during four months on the estimation of this transition matrix.

We show that for a reduced number of observations, we are able to give a parametrization of the set of models that are consistent with the observations. Nonetheless, for a higher number of observations, we had to design some heuristic algorithms, the most efficient one is called L2 gradient descent. Of course, we are always able to determine the best model in a finite set.

Finally, we show that the transition matrix is not the good purpose of the estimating procedure. This aim should be an equivalence class that is not reduced to one model. We used a simple classifier to produce an approximation of this equivalence class.

Even if the classification among equivalence classes can be made thanks to an huge dataset, the question of the geometry of equivalence classes is not solved. Moreover the process of choosing the optimal policy for the AI once we know the equivalence class of the model of the human remains to be studied.

REFERENCES

[NR00] Andrew Y. Ng and Stuart Russel. Algorithms for inverse reinforcement learning. *ICML*, 2000.

[PDRT] David Parkes, Christos Dimitrakakis, Goran Radanovic, and Paul Tylkin. Multi-view decision process. (unpublished).

[Put05] Martin L. Puterman. *Markov Decision Processes, Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics, 2005.