A Model-Based Reinforcement Learning Approach for a Rare Disease Diagnostic Task.

Rémi Besson[1], Erwan Le Pennec[1], Stéphanie Allassonnière[2]
CMAP Ecole Polytechnique[1], Université Paris-Descartes[2]
remi.besson@polytechnique.edu

1. Introduction

There are three compulsory ultrasound test during pregnancy in France. Some classical measures are done for every women, e.g the research of trisomy 21 is well known and mastered. On the contrary there is no strict protocol defined for the research of rare diseases. We want to help obstetricians to improve ultrasonic diagnostic.

2. Available Data

We have a list of 80 rare diseases. Each disease has a list of associated symptoms (typical symptoms). Our database references 220 different symptoms. We write:

$$B_i = \begin{cases} 1 & \text{if the fetus has the symptom of type i} \\ 0 & \text{otherwise .} \end{cases}$$

We denote the diseases: $D \in \{d_1, ..., d_k\}$. We know $\mathbb{P}[D = d_j]$ as well as $\mathbb{P}[B_i = 1 \mid D = d_j]$ for all the symptoms S_i typical of D. Note that joint distribution of symptoms given the disease is not available, but only the marginals.

3. What we aim to optimize

Markov Decision Process (MDP) framework.

- **State**: What we know at the current time about the symptoms of the patient.
  $s = (s^{(i)})_{i=1}^{220} \in \mathbb{S}$, $s^{(i)} = 1$ if symptoms i is present, 0 if absent, 2 if non observed yet.
- **Action**: The next symptom we suggest at the obstetrician to look at. $a \in \mathbb{A}$
- **Policy**: $\pi : \mathbb{S} \to \mathbb{A}$

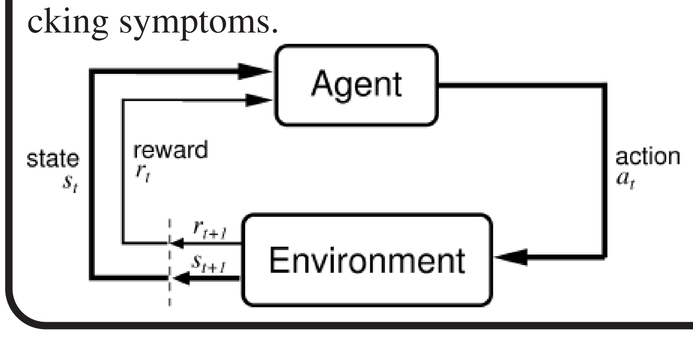
We aim to learn

$$\pi^* = \arg\min_{\pi} \ \mathbb{E}_{\mathcal{P}}\big[I \big| s_0, \pi\big], \tag{1}$$

where $s_0 = (2, ..., 2)$ is the initial state, $\mathcal{P}$ the law of the environment currently used and I is the random number of inquiries before reaching a terminal states, i.e:

$$I = \inf\{t \mid H(D \mid S_t) \leq \epsilon\}$$

where $H(D \mid S_t) = \sum_{s_t} \mathbb{P}[S_t = s_t] H(D \mid S_t = s_t)$ is the entropy of the random variable disease D given what we know at time t: S_t. We should think s_t as a realization of the r.v S_t.
We are ensured that $H(D \mid S_{t+1}) \leq H(D \mid S_t)$, see [4], "information can't hurt". In summary, when we consider that entropy is sufficiently low and that we can stop and propose a diagnosis, we know that on average, the uncertainty about the patient's disease would not have increased if we had continued checking symptoms.



4. Planning algorithms

- **A policy-based approach**: $\pi_\theta(s, a) = e^{\theta^T \phi(s,a)} \big/ \sum_b e^{\theta^T \phi(s,b)}$ where $\pi_\theta(s, a)$ is the probability to take action a in state s, $\phi(s, a)$ is a feature vector: a set of measures linked with the interest of taking action a when we are in state s.
- **A value-based approach**: the objective is to learn the Q-values, the expected reward when taking action a in s:

$$Q_\pi(s, a) = \mathbb{E}\left[\sum_{t'=t}^{I} r_{t'} \mid s_t = s, a_t = a, \pi\right]$$

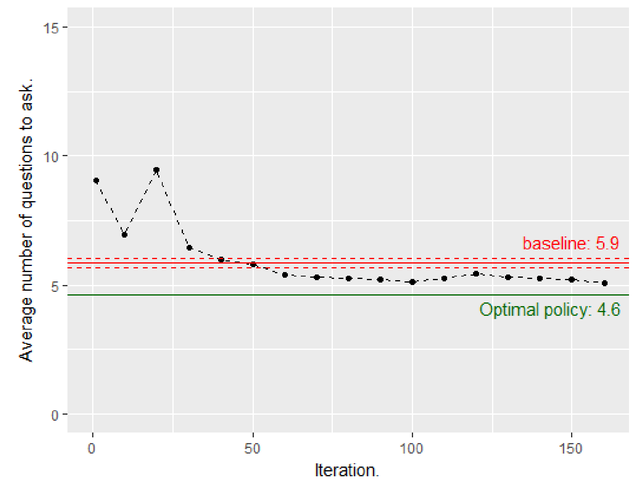
which are parameterized, for example by a neural network, $Q_\pi(s, a) \approx Q_w(s, a)$. These parameters w are learned through a Deep Q-Network (DQN) algorithm [5] where the classic temporal difference update is switched by a Monte-Carlo (MC) update which exhibits better performance on our problem.

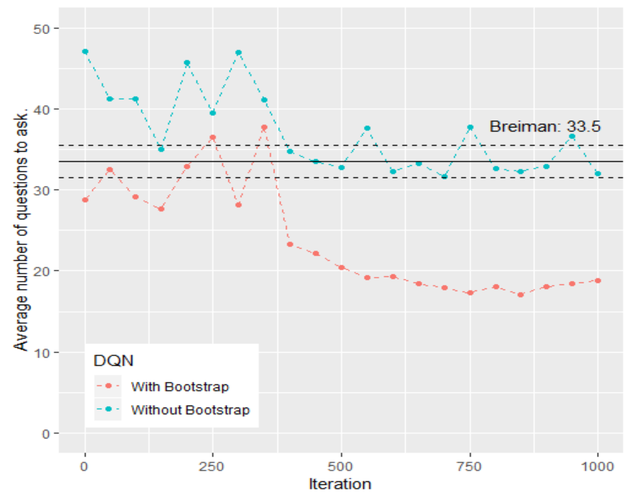
5. High-dimensional issues

A DQN algorithm is not tractable for the main task (1).
We then created 220 tasks $\mathcal{T}_i$ to solve, $\forall i$, $s_{(i)} = (2, ..., 2, 1, 2, ..., 2)$:

$$\pi^*_{(i)} = \arg\max_{\pi} \mathbb{E}_{\mathcal{P}}\left[I \mid s_{(i)}, \pi\right]. \tag{$\mathcal{T}_i$}$$

Some of the tasks $\mathcal{T}_i$ are sufficiently low-dimensional to be solved using a classic look-up table Q-learning algorithm (see [3]). For the others tasks we use a DQN algorithm. Playing games from the start to a terminal state (MC update) may be time-consuming and returns will suffer from a high variance. The key idea is to note that the different tasks have intersections, we then play games from the start state $s_{(i)}$ to a state where we already learned how to play and bootstrap.





6. Learning a model of the environment

How to mix experts and data ? We define our estimator as the distribution closest to the initial a priori compatible with the data:

$$\hat{p}^{\mathcal{L}}_{\epsilon_n} = \arg\min_{p \in \mathcal{C} / \mathcal{L}(p_n^{\text{emp}}, p) \leq \epsilon_n} \mathcal{L}(p^{\text{maxent}}, p) \tag{2}$$

where $\epsilon_n := \epsilon_n^\delta = \arg\min_l \mathbb{P}[\mathcal{L}(p_n^{\text{emp}}, p^\star) \leq l] \geq 1 - \delta$.
Notations: $\mathcal{L}$ a given dissimilarity measure. p_n^{emp} the empirical distribution. p^{maxent} the initial a priori given by experts. $p^\star$ the true distribution we aim to estimate.

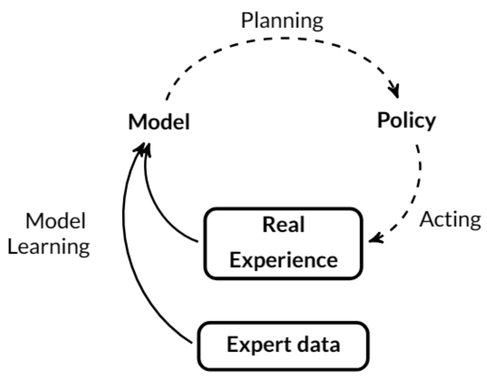
Theorem 1. *Let $\hat{p}_n^{1,1}$ (resp $\hat{p}_n^L$) be $\hat{p}^{\mathcal{L}}_{\epsilon_n}$ in the case where $\mathcal{L}$ is the L^1 norm (resp. the $\mathbb{KL}$ divergence) then we have with probability at least $1 - \delta$:*

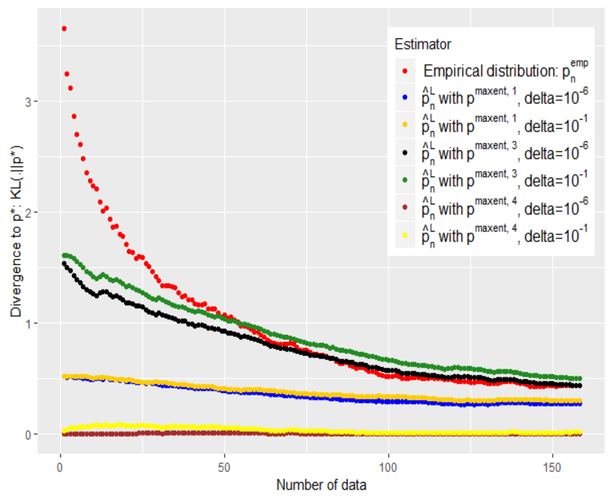
$$\|p^\star - \hat{p}_n^{1,1}\|_1 \leq 2 \min\{\epsilon_n, \|p^\star - p^{maxent}\|_1\} \tag{3}$$

and

$$\mathbb{KL}(\hat{p}_n^L || p^\star) \leq \min\left\{\mathbb{KL}(p^{maxent} || p^\star), \epsilon_n \left(L_n + 1\right)\right\} \tag{4}$$

where $L_n = \frac{\mathbb{KL}(p^{maxent} || p^\star) - \mathbb{KL}(p_n^{emp} || p^\star)}{\mathbb{KL}(p_n^{emp} || p^{maxent})}$.





8. References

[1] Jaynes E.T., Information Theory and Statistical Mechanics, *The Physical Review*, 1963.
[2] Sutton R.S, Barto A.G., Introduction to reinforcement learning, *Bradford Book*, 1998.
[3] Cover T. and Thomas J., Elements of Information Theory, *Wiley Series*, 2006.
[4] Mnih V. and al., Playing Atari with Deep Reinforcement Learning., 2013