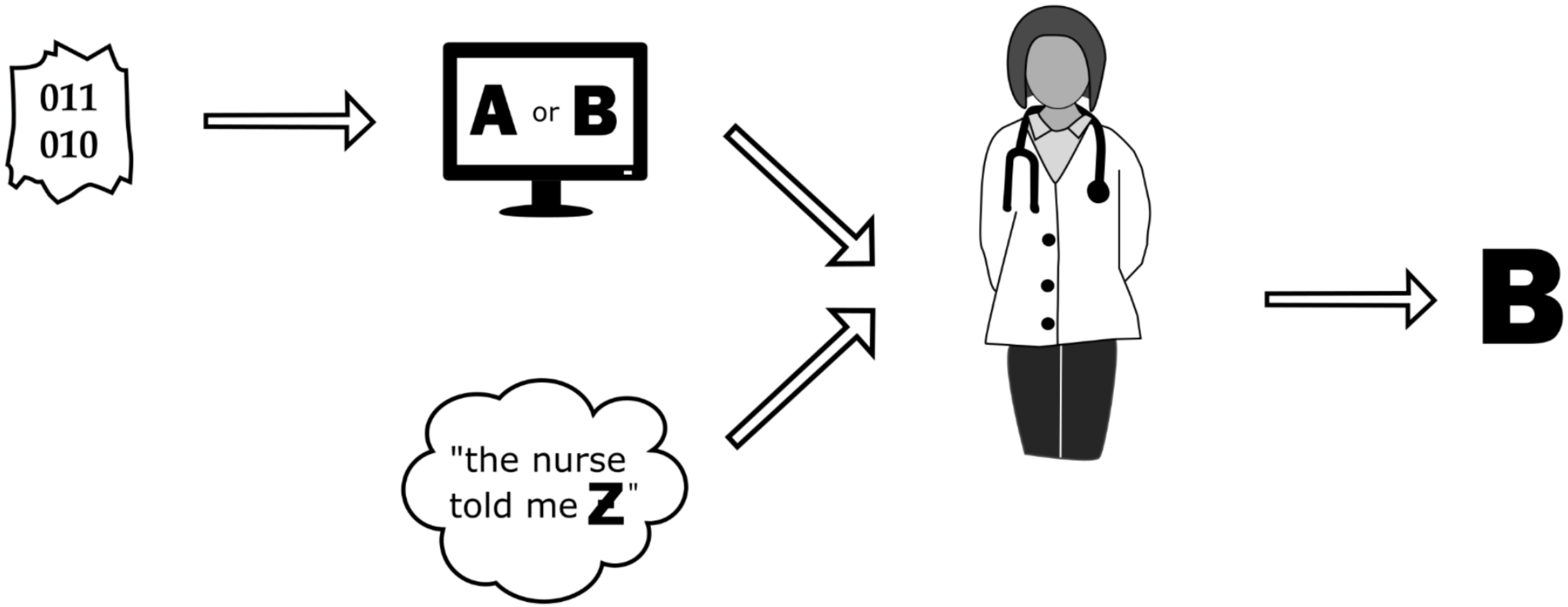
A Bandit Model for Human-Machine Decision Making with Private Information and Opacity

Sebastian Bordt, Ulrike von Luxburg

University of Tübingen, Max Planck Institute for Intelligent Systems, Tübingen, Germany

The Model



The computer informs the human, who then decides

Why this Model?

In machine learning, we have many different learning models:

- Supervised Learning
- Reeinforcement Learning (Learning in MDP's)
- Bandits (Multi-Armed Bandits, Contextual Bandits, ...)
- Active Learning
- Clustering
- ...

What about human-machine learning?

What are the learning problems that we are interested in?

Some examples:

- COMPAS
- Cardiac Arrest and Diabetic Retinopathy detection
- Medical Imaging
- Program Admission

Whenever a computer and a human try to jointly arrive at decisions!

Wait, why don't we just formulate these problems as supervised learning problems?

- When our goal is come up with computer programs that support human decision makers in novel ways, we might simply don't have the datasets
- How do we know what's the right way to set up the supervised learning problem?
- Nevertheless, we will almost always use supervised learning as an intermediate step.
- However, we are ultimatly not interested in performance on proxy prediction problems, but in the ultimate performance of the joint human-machine decision making venture.
- As an example, consider the problem that COMPAS tries to solve.

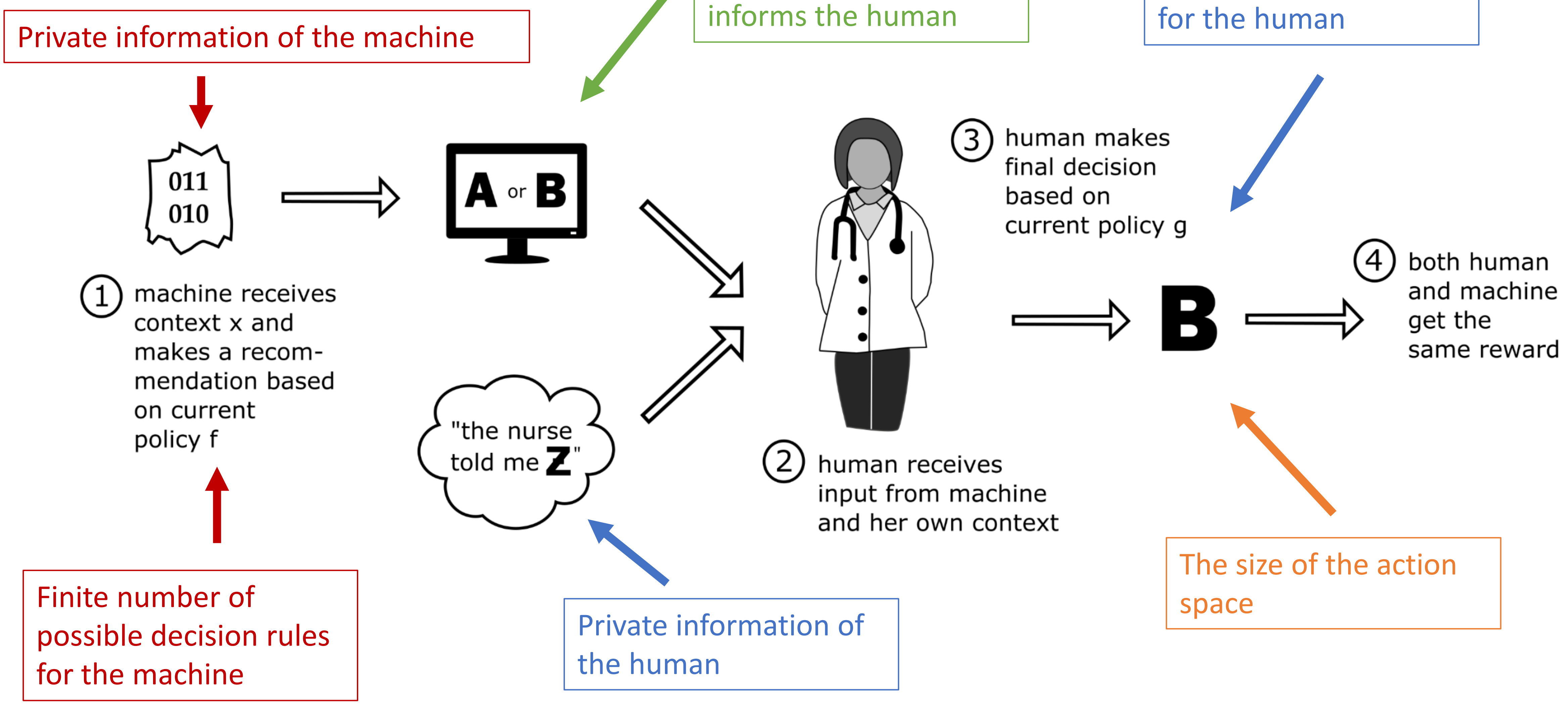
Assumptions on the Structure of the Problem: What do we want to cover in the model?

1. The computer learns how to advise the human
2. The human learns how to work with the computer
3. The human has to make the final decision
4. Decision making is hampered by the fact that both decision makers understand each other only imperfectly. This might be due to
   a) the presence of private information.
   b) opacity about the other players decision process.

What Questions do we want to Answer?

- What are the consequences of private information and opacity?
- What are optimal solution strategies for human-machine decision making problems?
- Which quantities influence the hardness of human-machine decision making problem?
- How can we design human-machine interaction such that efficient learning is possible for both parties?

The Model



Private information of the machine
011
010
A or B
(1) machine receives context x and makes a recom-mendation based on current policy f
"the nurse told me Z"
Finite number of possible decision rules for the machine
The size of the space in which the machine informs the human
Private information of the human
(2) human receives input from machine and her own context
(3) human makes final decision based on current policy g
Finite number of possible decision rules for the human
B
(4) both human and machine get the same reward
The size of the action space

The Model

In round $t = 1, ..., T$

1. Context $x_t \in \mathcal{X}$ is revealed to Player 1
2. Player 1 decides on a recommendation $r_t \in \mathcal{R}$
3. Context $z_t \in \mathcal{Z}$ and recommendation r_t are revealed to Player 2
4. Player 2 decides on an action $a_t \in A$
5. Reward $y_t \in [0, 1]$ and action a_t are revealed to both players

Figure 2: Interaction in our contextual bandit model.

Minimax Regret

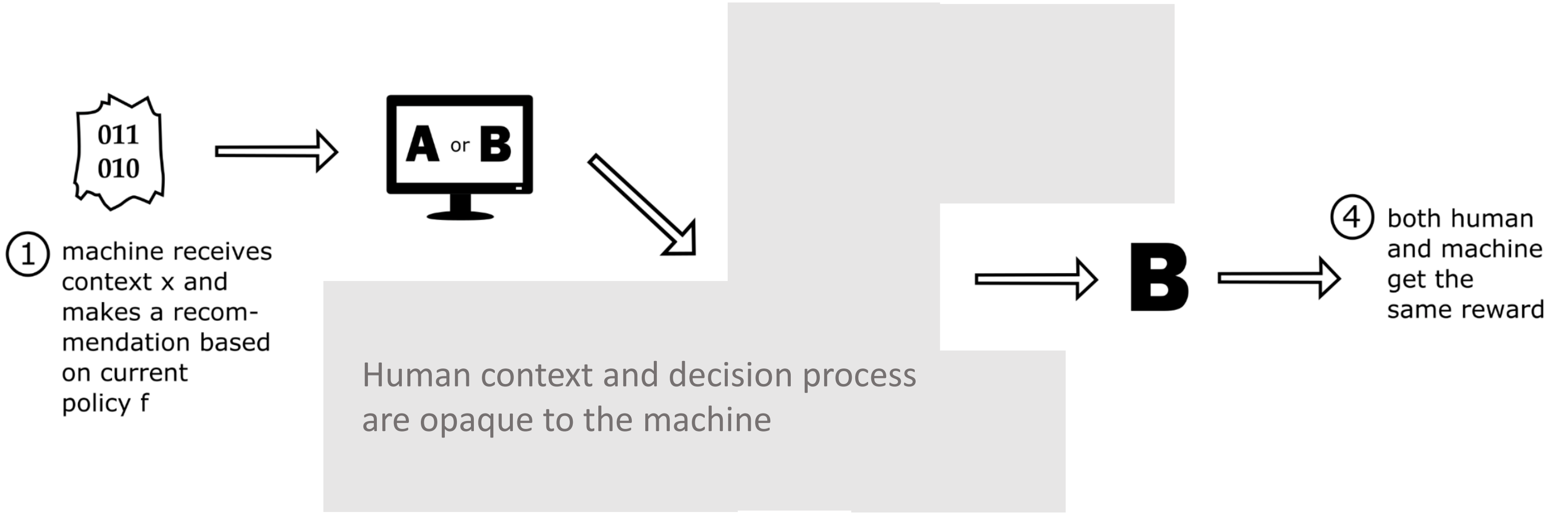
$$R_T = \inf_A \sup_{\mathcal{D}} \sup_{|\Pi_1|=N_1} \sup_{|\Pi_2|=N_2} \mathrm{Reg}_T$$

Worst-case lower bound for optimal algorithmic advice

Theorem 3. *(Lower bound in the number of policies of the first player) Assume that Player 2 only plays actions that are suggested by policies in Π_2. Let $N_2 = 1$ and $K = 2$. There exists a universal constant $c > 0$ such that*

$$R_T \geq c\sqrt{TN_1}.$$

The human does not have to learn



Have to try all decision rules separatly: Advising an opaque human can be much harder than choosing actions directly

Worst-case lower bound: Proof strategy

- Take a worst-case bernoulli bandit with N_1 arms
- Let the hidden context vector of the human be the payoffs of this bernoulli bandit (a N_1 - dimensional vector of zeroes and ones)
- Let the space of recommendations be of size N_1
- Let the action payoffs be fixed at 0 and 1
- Let the policy of the human be that she assigns recommendation i the payoff of the i-th arm of the bernoulli bandit

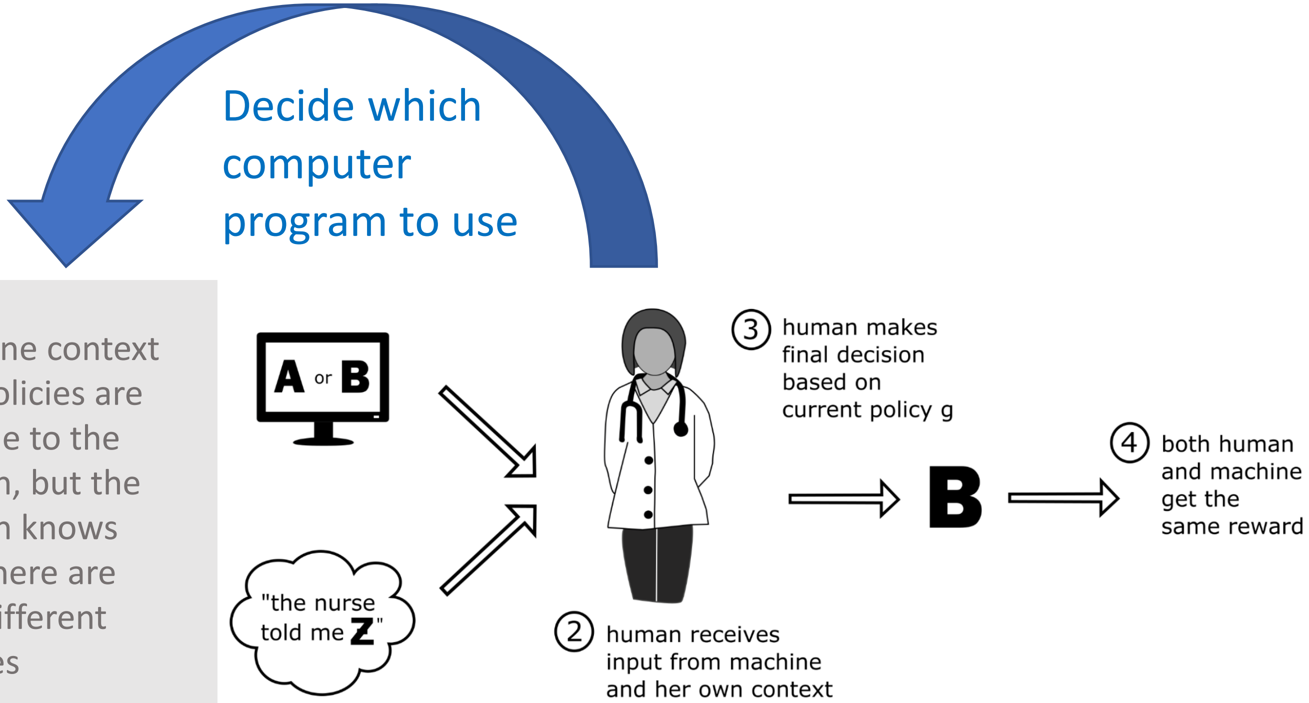
Solution strategy for the human

Theorem 4. *(Logarithmic regret in the number of policies of the second player) The P2-EXP4 algorithm, with* $\eta = \sqrt{2\log(N_1 N_2)/(TKN_1)}$ *and* $\gamma = 0$*, satisfies*

$$R_T \leq \sqrt{2TKN_1 \ln(N_1 N_2)}.$$

Additional term 2 K ln(N_1 N_2). Subject to the hardness constraint faced by the machine, the human can learn efficiently.

Machine context and policies are opaque to the human, but the human knows that there are N_1 different policies



Algorithm P2-EXP4
Parameters: $\eta > 0$, $\gamma > 0$
Initialization: $Q_1 \in [0,1]^{N_1 \times N_2}$ with $Q_{1,ij} = \frac{1}{N_1 N_2}$
For each $t = 1, \ldots, T$

1. Player 2 tells Player 1 to play policy i_t according to $q_{ti} = \sum_{j=1}^{N_2} Q_{t,ij}$
2. Player 1 recommends $r_t = f_{i_t}(x_t)$
3. Player 2 chooses action a_t according to (1)
4. Players receive reward y_t and Player 2 estimates $\hat{y}_{tk} = 1 - \frac{1_{\{a_t = k\}}}{q_{t,i_t} p_{tk} + \gamma}(1 - y_t)$
5. Player 2 propagates rewards to policies $\hat{Y}_{t,ij} = 1_{\{i_t \neq i\}} + 1_{\{i_t = i\}} \hat{y}_{t,g_j(r_t,z_t)}$
6. Player 2 updates Q_t using exponential weighting

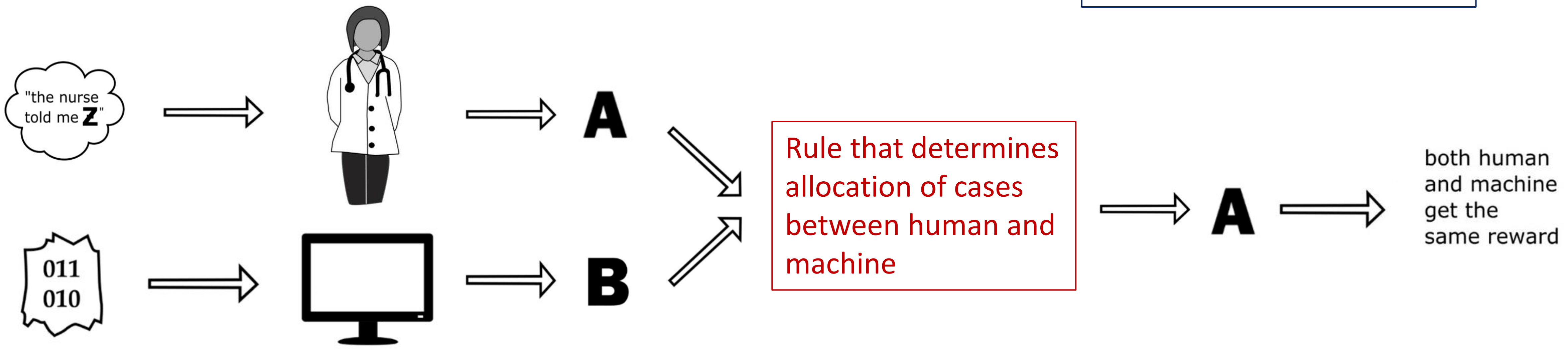
$$Q_{t+1,ij} = \frac{\exp(\eta \hat{Y}_{t,ij}) Q_{t,ij}}{\sum_{l,m} \exp(\eta(\hat{Y}_{t,lm}) Q_{t,lm}}$$

Different variants of the problem, some allow for efficient learning

Definition 5 (Policy space independence). *We say that the two policy spaces* Π_1 *and* Π_2 *are independent with respect to* $\mathcal{D}$ *if, for all* $f_1, f_2 \in \Pi_1$ *and* $g_1, g_2 \in \Pi_2$,

$$\begin{aligned} & Y\big(g_1(f_1(x), z)\big) - Y\big(g_1(f_2(x), z)\big) \\ = \; & Y\big(g_2(f_1(x), z))\big) - Y\big(g_2(f_2(x), z)\big). \end{aligned}$$

More variants, discussed in Section 7 of the paper



More variants of the problem …