AN ATTEMPT TO MODEL HUMAN TRUST WITH REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing works to compute trust as a numerical value mainly rely on ranking, rating or assessments of agents by other agents. However, the concept of trust is manifold, and should not be limited to reputation. Recent research in neuroscience converges with Berg's hypothesis in economics that trust is an encoded function in the human brain. Based on this new assumption, we propose an approach where a trust level is learned by an overlay of any model-free off-policy reinforcement learning algorithm. The main issues were i) to use recent findings on dopaminergic system and reward circuit to simulate trust, ii) to assess our model with reliable and unbiased real life models. In this work, we address these problems by extending Q-Learning to trust evaluation, and comparing our results to a social science case study. Our main contributions are threefold. (1) We model the trust-decision making process with a reinforcement learning algorithm. (2) We propose a dynamic reinforcement of the trust reward inspired by recent findings of neuroscience. (3) We propose a method to explore and exploit the trust space. Experiments show that it is possible to find a set of hyperparameters of our algorithm that reproduces the Dunning-Kruger effect, whereby beginners are overconfident while experts tend to assess their confidence level more accurately.

1 INTRODUCTION

Trust seems to be one of the most overused words in the Information Technology world. Yet the concept is struggling to find a universal definition: "the term is never defined. The usage is much clearer than the concept itself" (Hunyadi, 2020). At the end of the day, the question of trust in the digital world is rarely raised; scientists and industrialists prefer to focus on the effects of its erosion - most often - or its improvement - more rarely - on related issues: digital identity, privacy, personal datas, security, customer relationship, artificial intelligence, financial services digitalisation.

1.1 WHAT IS TRUST?

Trust functions as a way to reduce complexity in a social system, while allowing it to evolve towards richer interactions. It acts as a stabilising structure for societies (Gambetta et al., 2000). Trust makes it possible to act in contexts where information is lacking, or even unavailable (the relations between present and future events being indeterminable). Luhmann (2000) has provided semantic distinction between trust and confidence. In short:

- **Confidence** allows an individual to act in spite of an inherent danger in the social system because he has faith in institutionalised factors of the society, his culture, his knowledge.
- **Trust** is an assessment of risk by an individual who makes a rational choice. It is a risky bet on a future of which he does not know all the possibilities.

1.2 TRUST IS AN UNDERLYING AND FUNDAMENTAL STRUCTURE OF SOCIAL SYSTEMS

A society is built around social systems: economic, such as companies, political, such as states, legal, such as a code of laws. A social system is composed of people. But it is also an aggregation of groups of people. For example, the social system of any company is composed of its employees,

shareholders, customers and partners. A social system can be represented by a set of available actions to its members. A *choosen action* presupposes a rational choice. An *assured action* is conditioned by a rule or a situation implicitly or explicitly linked to the social system. A real action may have a part of chosen motivation and a part of assured motivation.

Trust allows the binding of groups of people whose number does not exceed 100 to 150 (Aral, 2020; Harari, 2016): "The neocortex ratio is strongly correlated with group size" (Dunbar, 1992). This form of trust, which allows us to decide on a chosen action, is based on a calculation of risk in an uncertain context . A software developer, for example, trusts his/her scrum master to prioritise issues and manage deadlines according to the custormers' wills. It is a matter of trust when relationships need to be cemented within the project; the agile method's ceremonies should be seen as tools for building trust.

Confidence enables the groups to be linked together. No state, no code of law, no multinational company could have been built without confidence. Condidence makes it possible to unify a social system in spite of the oppositions of the groups that form it; a state, for example, is often the result of the aggregation of antagonistic tribes or feudalities. For most likely evolutionary reasons, condidence is based on a *shared narrative* (or *fiction* or *totem* (Harari, 2016)).

According to Luhman, confidence is a precondition for the establishment of trust. When he/she engages in a new project with a scrum master he/she does not know, our software developer faces the risk of depending on the relationship between the latter and the customer (*trust*) because the two colleagues are in the same company and share a *shared narrative* which is a necessary condition for confidence: the corporate culture. If employees no longer have a shared narrative - whether it is based on economic value creation, human values or some other project - the danger is that the lack of confidence inhibits employees in their interpersonal relationships and they are only willing to collaborate with a small number of colleagues they already know. "The lack of confidence leads to retreat into a restricted universe of purely local dimensions" (Luhmann).

However, trust is a necessary condition for the very existence of a social system. At its birth or when it is in crisis, the generative capacity of the system relies on the willingness of its members to take risks in their relationships with each other, i.e. engage on trust. When a startup is pivoting towards a shiny new business or when a competitor offers a better product at a lower cost, it is the ability to trust each other between the scrum master, the manager and the software developer that makes it possible to challenge each other and be resilient. "A system, whether it is economic, legal or political, requires trust as an input condition. Without this trust, it cannot induce supportive activities in situations of uncertainty or risk" (Luhmann).

Yet the properties of the social system are bound to confidence. A very vertical organisation often relies on the embodiment of the shared narrative by the leader. Conversely, in a horizontal system, the shared narrative is continuously built between the stakeholders (employees, shareholders, customers, etc.) on the basis of the company's project.

1.3 Research Questions

With the digitalization of human relationships, shared narratives tend to become more *liquid* (Bauman, 2013). As a consequence, confidence is decreasing to the benefit of trust. This can be observed through many positive or negative externalities (e.g. fakenews, distrust of politicians). Thus, to better understand the phenomenon, our goal is to study trust through a novel approach, trying to compute a numerical representation of trust.

1. Can we build a digital model of trust?
2. Is it possible to compute trust?
3. Is it possible to understand the drivers of its evolution?

The main contribution of this paper is a *trust* learning algorithm based on reinforcement learning. We called this algorithm SelfQ-Trust because the trust level an agent has in itself is evaluated by a Q-value calculated by the algorithm. Additional contributions allowed us to make the SelfQ-Trust algorithm an overlay of any model-free reinforcement learning algorithm. We propose a dynamic reinforcement of the trust reward ; a kind of distributional reinforcement learning applied on a vector of trust levels ; a method for explore and exploit the trust space.

2 RELATED WORK

The economics of internet platforms heavily depend on the quality of information and the security of users (Barbosa et al., 2020). Trust models based on multi-agent systems (MAS) are used to help users choose their interactions by avoiding those who are allegedly incompetent or malicious (Ramchurn et al., 2004; Sabater & Sierra, 2005). Many algorithms assign each agent a score based on the result of its past interactions with others (Jøsang et al., 2007). Both implicit (e.g. frequency, duration, closeness of relationships) and explicit (e.g. comments and ratings from other agents, payment informations) measures are used and the score is considered as a measure of the trust the community can place in the agent (Ben-Naim et al., 2020). The applications cover almost all uses of the Internet. Examples include: e-commerce (Amazon, etc.) (Xiong & Liu, 2003), collaborative economy platforms (Airbnb (Alsheikh et al., 2019), etc.), messaging (WeChat (Lien & Cao, 2014), etc.)

In such a model, the concept of trust is often confused with that of reputation (Jøsang, 2007). To the best of our knowledge, the theoretical frameworks are based on a cognitive model (Cho et al., 2015) or on game theory (Wang et al., 2016). In cognitive models, an agent evaluates the trust it places in another based on the latter's underlying degree of belief. But as it does not have direct access to the mental states of the latter, the trust decision is based on the observation of the outcome of the interaction. The observation model takes into account the uncertainty that arises from the - at least initial - unpredictability of the other agent. In this context, Bayesian networks are a logical choice of knowledge representation (Esfandiari & Chandrasekharan, 2001). Game theory models are based on Gambetta's definition of trust: "Trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends" (Gambetta, 1988). Trust is the result of a game which involves a utility calculation based on knowledge of the history of relationships. A well-known trust game used in econmics is that of Berg et al. (Berg et al., 1995). The assessment of these models can be done theoretically by establishing a set of properties to be achieved by the system. The experimental approach consists in developing a test bed where different trust systems compete (Sabater & Sierra, 2005).

Previous work attempting to measure trust in MAS with Q-learning (see e.g. Vijaya Kumar & Jeyapal (2014); Aref & Tran (2018)) has relied on explicit measures of the relationship - which introduces a modelling bias. Meanwhile, the hypothesis advanced by Berg et al. (1995) in their work on the "trust game" is being confirmed by recent work in neuroscience. Meyniel et al. (2015) argue that in humans, trust is not derived from a heuristic process but rather is formed during their learning process. For those reasons, we have taken the problem to be that of modelling trust as a human brain learning process by exploiting the latest knowledge on the functioning of the reward circuit.

3 BACKGROUND

3.1 TERMINOLOGY

We consider a social system $\mathcal{S}$ defined by an environment and a finite set of predefined actions $\mathcal{A}$. A population $\mathcal{I}$ of individuals represented by a multi-agent system (MAS) whose trust evolution relative to $\mathcal{S}$ is to be modelled. As defined by Russell & Norvig (2020), "an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors". Any agent of $\mathcal{I}$ is defined mathematically as an application f which maps every possible percepts sequence $(o_1, ..., o_n) \in \mathcal{P}^*(\mathcal{O})$, where $\mathcal{O}$ is the set of possible observations of the environment by the agent, to an action $a \in \mathcal{A}$ that the agent can perform or to any other function (e.g., each sub-function of an algorithm intrinsic to the agent) that affects its possible actions: $f : \mathcal{P}^*(\mathcal{O}) \to \mathcal{A}$.

In his doctoral work, Marsh (1992) was the first to formalise trust as a computational concept. He introduced a measure of trust $T_x(y, \alpha) \in [-1, 1[$ as the probability that an agent y will act if $T_x(y, \alpha) \geq 0$ (respectively minus the probability it will not act, if $T_x(y, \alpha) < 0$) in the situation α to reach an outcome that another agent x presupposes to expect if it had a full trust in y, before y acts. If $T_x(y, \alpha) > threshold(y, \alpha)$ the agents cooperate.

Research in social sciences and information technology has extended the English terminology from property law [1] [2]. Thus, according to Hardin (2002), a "trustor" is an entity that trusts another entity, the "trustee". According to Cofta (2007), a *trustor* can be a social actor (such as a person or an institution) or a technical actor (such as a computer or a software), which acts on behalf of a social actor. In the following, we will note $f \in \mathcal{I}$ to refer to a trustor who is placing its trust in a trustee $g \in \mathcal{I} \setminus f$ when it wants to perform an action chosen from a part of the available actions $\mathcal{P}(\mathcal{A})$.

3.2 Q-LEARNING

A single-agent reinforcement learning problem is commonly modelled by a Markov decision process (MDP). A MDP is usually formalized by a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}\}$ where $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $R \in \mathcal{R}$ stands respectively for state, action and reward. For any $s \in \mathcal{S}$, agent's choice to perfom an action $a \in \mathcal{A}$ occurs whith the probablility $T(s, a, s') \in [0, 1]$ where s' is the next state following s. It will result in the environment entering the new state s' and give a reward $R(s, a)$. A policy π describes which action the agent takes in each state. Formally, it is therefore a function $\mathcal{S} \rightarrow \mathcal{A}$ in the case of a deterministic policy or $\mathcal{S} \times \mathcal{A} \rightarrow [0; 1]$ in the stochastic case. The goal for the agent is to learn a policy that maximizes the cumulative rewards received over its learning process.

A common way to achieve the goal is to solve the Bellman equation applied to the total sum of discounted rewards $Q_\pi(s, a)$ reaped by the agent if it starts in state s and first takes the action s, before then applying the policy π ad infinitum :

$$Q^*(s, a) = \mathbb{E}\left(R + \max_{a' \in \mathcal{A}} Q^*(s', a')\right) \tag{1}$$

Introduced by Watkins (1989), Q-learning is a numerical method of solving Equation 1. At each step of the algorithm, agent choose action a from state s using policy derived from Q (e.g. ϵ-greedy). Then it makes a step in environment, observe the new state s' and the reward R. Q-value is updated by applying a temporal-difference method (TD) (Sutton, 1988) :

$$Q(s, a) \leftarrow Q(s, a) + \alpha\left[R + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s, a)\right] \tag{2}$$

where $Q(s, a)$ is an estimate of the pair (state,action) (s, a), Q-value and $\alpha \in [0, 1]$ is a learning rate setting the updating speed of Q-values at each time step. Watkins & Dayan (1992) formally showed that the algorithm converges to the optimal policy $Q^*(s, a)$ if all actions in all states are infinitely sampled and the (state,action) values are represented discretely.

3.3 MODELLING THE DOPAMINERGIC SYSTEM

Our work is based on the Berg's hypothesis that trust is a function that has been encoded in the human brain through evolution: "Evolutionary models predict the emergence of trust because it maximizes genetic fitness" (Berg et al., 1995). Consequently, important elements of our contribution are directly inspired by recent results on the modelling of learning in the human brain.

Cognitive neuroscientists linked the functioning of the neurotransmitter dopamine to the theoretical computational framework of reinforcement learning in the early 2000s (Holroyd & Coles, 2002). It is now established that the release of dopamine in the reward circuit reveals a prediction error by the individual (Glimcher, 2011). This surprise signal favours the learning of reward predictions and shapes the individual's future behaviour.

The reward circuit describes a network between two groups of neurons, one located in the ventral tegmental area and the other in the nucleus accumbens. There is one in the left hemisphere and one in the right hemisphere of the brain. The reward circuit serves to reinforce some of our behaviours: for example, it is involved in eating or sexual behaviour, but it is also involved in screen addictions

[1] **trustor** (*Collins*): (in property law) a person who sets up a trust transferring property to another person

[2] **trustee** (*Collins*): someone with legal control of money or property that is kept or invested for another person, company, or organization

(Bohler, 2020). The main neurotransmitter emitted in the reward circuit is dopamine, but others are also present; this is why one can also read - even if it is not rigorously exact from the point of view of neuroscience - of the dopaminergic circuit when we study the mechanisms of attention.

Predictions of reward were previously represented as a single scalar quantity. However, recent artificial intelligence research on distributional reinforcement learning has inspired a new model in which the brain represents possible future rewards not as a single mean, but rather as a probability distribution, effectively representing multiple future outcomes simultaneously and in parallel (Bellemare et al., 2017). Recordings from the mouse ventral tegmental area provided strong evidence for the neural realisation of distributional reinforcement learning (Dabney et al., 2020). From a functional point of view, an interesting recent hypothesis suggests that dopaminergic neurons encode both differences between rewards and expectations in a goal-directed system. These prediction errors trigger reward learning (Bogacz, 2020).

4 METHODS

We propose the following definition by merging Luhmann's definition with Marsh's formalism:

Definition 1 (Trust) *Trust is both a decision-making process in a relationship between two intelligent agents and the measure by which the decision is made. Consider that a trustor f, in a state s, has to choose an alternative according to a bet in a relationship with a trustee g. The trust level $T_f(g, s) \in [-1, 1]$ is calculated from the subjective measure of the probability that the bet will come true. If the trust level is positive or above a so-called trust threshold, the trustor chooses to act in the situation where he trusts the trustor. If the trust level is negative or below a mistrust threshold, it chooses to act in the situation where it is mistrustful.*

4.1 PROBLEM STATEMENT

We consider a population $\mathcal{P}$ of agents whose behaviour in a given environment is modelled by a MDP $\{\mathcal{S}, \mathcal{A}, \mathcal{R}\}$. The agents learn an optimal policy by means of any underlying model-free off-policy reinforcement learning algorithm.

The goal is to estimate the trust level $T_f(g, s) \in [-1, 1]$ that a trustor $f \in \mathcal{P}$ can have in any agent $g \in \mathcal{P}$ which advises it to take the action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ following the introspection of its MDP model. We require that: (1) the trust measurement algorithm runs in parallel with the MDP learning algorithm and (2) it can be implemented with any model-free off-policy RL algorithm.

4.2 LEARNING PRINCIPLES

The general explanations concern the multi-agent case, in which each agent assumes both the role of *trustor* and that of *trustee*. For a better understanding of the basics of the method, simplifying assumptions are made: the trust environment perceived by the agent is deterministic and the *trust actions* defined hereafter are discretised.

The trust level $T_f(g, s)$ is learned by a trust model formally described by a MDP $\{\mathcal{S}, \hat{\mathcal{A}}, \hat{\mathcal{R}}_{f,g}\}$, where $\hat{\mathcal{A}} = \mathcal{A} \times [-1, 1]$ is the set of trust actions and $\hat{\mathcal{R}}_{f,g} : \mathcal{S} \times \mathcal{A} \times [-1, 1] \rightarrow \mathbb{R}$ is a reward function. By applying the Definition 1 with a trust threshold of 0, we define a trust action $\hat{a} = (a, t)$ as follows: f trusts g to take action a with trust level t means that the closer t is to 1, the greater the likelihood that f will take action a and the closer t is to -1, the greater the likelihood that f will not take action a.

SelfQ-Trust Algorithm (see Appendix A) focuses on the single-agent case, with agent f assuming both roles of trustor and trustee. From a technical point of view, the trust evaluation should be seen as an introspection function of the agent that evaluates in real time the relevance of its learning algorithm's prediction of an underlying MDP with respect to its current state.

Overall operation The trustor f asks any agent g for the action a it would have taken if it had been in the same state s as f. This advice allows f to choose its action. f then learns its own model of trust towards g thanks to a trust reward $\hat{r}$. g may be a trustee found in a queue of trusted agents or f itself. In the latter case, the algorithm calculates the trust that f has in itself.

To do so, the algorithm learns a Q-function $\hat{Q}_{f,g} : \mathcal{S} \times \hat{\mathcal{A}} \rightarrow \mathbb{R}$ that calculates the quality of a state–trust action combination. At all times, the trust that f places in g in state s can be derived from the Q-function:

$$\begin{aligned} & T_f(g,s) = \frac{\arg\max_k q_k - 1}{n} - 1 \\ \text{with} \quad & \hat{Q}_{f,g}(s) = (q_k)_{1 \leq k \leq 2n+1} \text{ the Q-value trust vector} \end{aligned} \tag{3}$$

However, our initial attempts to make a simple instantiation of Q-learning in the state-trust action space yielded poor results. We faced two main challenges. Firstly, a fixed reward did not work for learning a realistic trust model. We thought that the value of the reward should vary according to the agent's experience in learning the MDP. Secondly, when reinforcement is applied to only one trust level, the model quickly locks on the initial level. To explore trust, reinforcement must instead be extended to all or part of Q-value trust vector.

A dynamic reinforcement model is used to answer the first issue. We applied a kind of distributional TD learning (see e.g. Dabney et al. (2020)) on the trust vector of the Q-value to address the second one. Finally, we propose an ad hoc method of exploration and exploitation.

Dynamic reinforcement. We use a recent hypothesis suggesting that dopaminergic neurons encode differences between rewards and expectations in a goal-directed system (Bogacz, 2020). We consider that trust is such a system where someone aims at better meet its expectations if he/she trusts someone else. A way to vary the reward according to the experience of trust relationships were to introduce a kind of attention mechanism. The general idea is (1) to increase the reward at the end of an episode when a positive trust action leads to an improved score and conversely (2) to make the reward proportional to the trust level of the trust action. For this purpose, we assume that the agent is able to measure its performance and that of other agents in solving the MDP. The trustor uses this measure and the prior trust level it has towards its advisor to calculate the magnitude of the trust reward. The magnitude of the trust reward $\hat{m}_{f,g}$ is regularly updated according to the observed performance of the trustee g in finding the optimal policy of the MDP. Each time the trustor f chooses an action advised by a trustee g, it receives a trust reward $\hat{r}$ depending on the prior trust level t it places in g and the magnitude of the trust reward $\hat{m}_{f,g}$:

$$\hat{r} = \hat{m}_{f,g} t \tag{4}$$

In the mono-agent case, we update $\hat{m}_{f,f}$ at the end of each episode i w.r.t. the score v_i the agent achieved in the episode i:

$$\hat{m}_{f,f} = -\frac{v_i - \underset{1 \leq j < i}{max}(v_j)}{v_i} \tag{5}$$

Distributional reinforcement learning. Instead of applying learning *only* on the trust level t prior to the relationship, it is applied on all possible trust levels. Applied to Q-learning simple case, we therefore weight the updating part of the Q-value trust vector $\hat{Q}_{f,g}(s) \in \mathbb{R}^{2n+1}$ by the realisation of a Gaussian random value centred on the prior trust level using Equation 6:

$$\begin{aligned} & \hat{Q}_{f,g}(s) \leftarrow \hat{Q}_{f,g}(s) + \delta(X_{2n-\hat{a}}, ..., X_{4n-\hat{a}}) \\ \text{with} \quad & \delta = \hat{\alpha}\left[\hat{m}_{f,f} t + \hat{\gamma} \underset{\hat{a}' \in \hat{\mathcal{A}}}{max} \hat{Q}_{f,g}(s', \hat{a}') - \hat{Q}_{f,g}(s, \hat{a})\right] \\ \text{and} \quad & X \in \mathbb{R}^{4n+1} \sim \mathcal{N}(0,\, 1) \\ \text{with} \quad & \hat{\alpha}, \text{ the learning rate and } \hat{\gamma}, \text{ the discount factor of trust learning} \end{aligned} \tag{6}$$

Exploration and exploitation of trust model. We use an epsilon-greedy method where $\hat{\epsilon}$ refers to the probability of choosing to explore the trust model. The specifics of the instantiation of this method in the context of trust learning are: (1) Trust exploration amounts to the trustor choosing to have full trust in the trustee; we hypothetize that trusting someone - or oneself - in a first relational experience - should it be self-oriened - is choosing to systematically take into account the advice

he/she gives. (2) Rather than just using the model being learned, the exploitation involves an uncertainty on the level of trust that the trustor grants to the trustee. In fact, as it is stated in the survey of Cho et al. (2015), the uncertainty on the a priori trust that one grants in an interpersonal relationship tends to fade with time.

In the mono-agent case, agent f chooses an action $a \in \mathcal{A}$ w.r.t. to its MDP model. Then, exploration is chosen with the probability $\hat{\epsilon}$. Otherwise exploitation is done with an uncertainty on the prior trust level t computed using Equation 7:

$$t = \frac{X}{n} \text{ , with } X \sim \mathcal{U}\left(\max(-n, n(T_f(f,s) - \hat{\epsilon})), \min(n, n(T_f(f,s) + \hat{\epsilon}))\right) \tag{7}$$

Overlay of an RL algorithm. The trust learning function can use any model-free reinforcement algorithm (Q-learning, DQN, DDQN, DDPG, A2C, A3C, etc.). We chose Q-learning or its simplicity to do the proof of concept. The use of a DQN, then of an on-policy algorithm (e.g. A3C) are expected to be validated in our future work.

5 EXPERIMENTS

There are public quantitative surveys dealing with trust (e.g. interpersonal trust, confidence in public institutions (Fitzgerald et al., 2016; Ortiz-Ospina & Roser, 2016)). Nevertheless, the available datasets are likely to be extremely biased because "the term is never defined. The usage is much clearer than the concept itself" (Hunyadi, 2020). Moreover, assuming that trust depends on at least 31 separate factors (see "composite trust" by Cho et al. (2015)), the interpretation of general questions about trust can be very different from one individual to another.

However, there are studies in the social sciences and economics that give results on observable properties of known systems. These studies can be used as test cases to assess the model. Among them, we have choosen the so-called *Dunning-Kruger effect* (DKE) to assess the trust of an agent in itself. Specifically, in their seminal paper, Kruger & Dunning (1999) put forward as an explanation a metacognitive difficulty of the unqualified people which prevents them from accurately recognising their incompetence and evaluating their real abilities. Conversely, the most qualified people would tend to underestimate their level of competence and would wrongly think that tasks that are easy for them are also easy for others. Dunning (2011) suggests that the effect is universal, as it has been observed in very different areas of learning.

Experimental Setup. We implemented the SelfQ-Trust algorithm in Python; the code is available at `https://github.com/selfQtrust/code`. Environment is a randomly generated maze of dimension (n, n). The goal of the agent is to start from the upper left corner and find the exit at the lower right corner. The maze generation algorithm ensures that at least one path exists between the entrance and the exit.

The agent learns the trust it has in itself in each state of the environment. In the maze case, the state is the cell where the agent is located. Thus, during an episode, an agent learns a sequence of trust levels, each of which is linked to a cell he visited. Using this sequence, we calculate three metrics to aggregate the confidence measures during the episode: the mean trust on all states of the environment at the end of the episode ($M1$); the mean trust over all states the agent passed through during the episode ($M2$); the max trust the agent had during the episode in the states it passed through ($M3$).

Simulation of the Dunning-Kruger Effect (DKE). In our first experiments, we were looking at whether the algorithm is able to reproduce the DKE effect. We trained a population of 1000 agents over 3 mazes of dimension $(6, 6)$, $(10, 10)$, $(30, 30)$. Computations of the three metrics $M1$, $M2$, $M3$ are shown in Figure 1.

The results show that the SelfQ-Trust algorithm reproduces the DKE trend: the agent's trust in itself goes through a peak systematically present at the beginning of the training, then decreases abruptly to a more or less pronounced local minimum, to end up rising and then stabilising. Moreover, the effect strength seems to increase with the complexity of the environment. The following experiment raises the question of the dependence of DKE on hyperparameters.

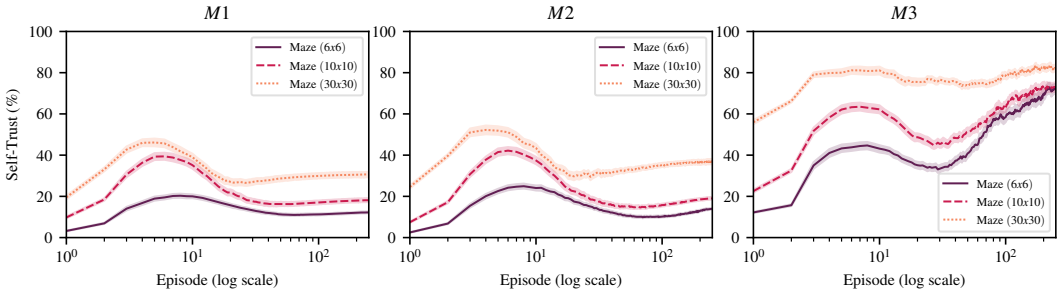

Figure 1: DKE simulation. 1000 agents are trained on 500 synchronised episodes of the underlying MDP and the trust model on each of the 3 mazes. Hyperparameters are set to: $(\alpha, \gamma, \hat{\alpha}, \hat{\gamma}) = (0.1, 0.99, 0.01, 0.99)$ for the 3 mazes. Estimate of the central tendency (mean estimator) and 95% confidence interval of self-trust computed by SelfQ-Trust vs. learning episode.

Sensitivity to learning hyperparameters. The learning model uses two pairs of learning hyperparameters: the underlying MDP that the agent uses to solve the maze problem uses an α learning rate and a γ discount factor; the trust model uses a rate of trust learning $\hat{\alpha}$ and a discount factor of trust learning $\hat{\gamma}$. The environment is the same maze of dimension $(5, 5)$ for the whole experiment. We trained a population of 50 agents on 60 episodes of MDP learning as well as on 60 episodes of trust learning - we choosed those settings accordingly to Sanchez & Dunning (2018) experiments on Overconfidence Effect. The second metric $M2$ is used to show the results in Figure 2.

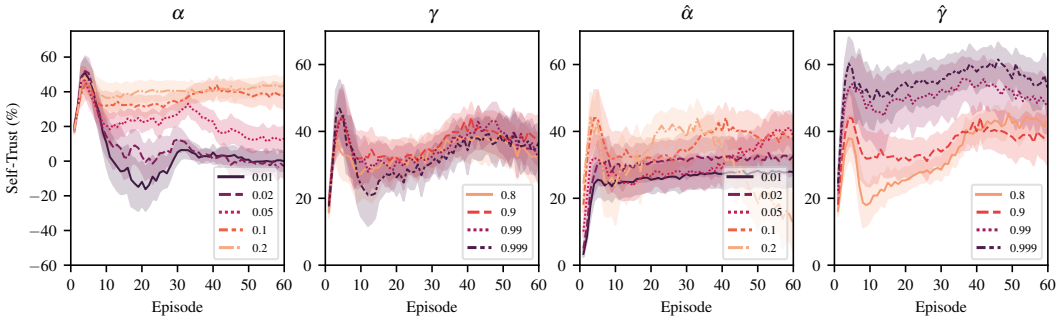

Figure 2: Sensitivity to learning hyperparameters α, γ, $\hat{\alpha}$ and $\hat{\gamma}$. We set $(\alpha, \gamma, \hat{\alpha}, \hat{\gamma}) = (0.1, 0.9, 0.1, 0.9)$ then for each of the figures we vary the hyperparameter studied in the set $\{0.01, 0.02, 0.05, 0.1, 0.2\}$ for α and $\hat{\alpha}$ and in the set $\{0.8, 0.9, 0.99, 0.999\}$ for γ and $\hat{\gamma}$. Estimate of the central tendency (mean estimator) and 95% confidence interval of self-trust computed by SelfQ-Trust vs. learning episode.

The experiment confirms the DKE trend, regardless of the hyperparameters choosen. Moreover, the magnitude of the trend appears to be correlated with α, $\hat{\alpha}$ and $\hat{\gamma}$ on the experience. γ does not seem to have much effect on the trust model (all curves are in the same statistical confidence interval). We wonder why the peak of confidence - which we will later call the peak of overconfidence - almost always occurs during the same short window of learning episodes?

Sensitivity to the complexity of the environment. The complexity of the environment should be narrowly assessed from the agent's point of view. We thus calculate it by training a population of agents with the underlying MDP learning algorithm - i.e. a simple Q-learning on a maze environment in the case of the experiment. We randomly generate 1000 mazes of dimension $(10, 10)$. We train an agent on each over 1000 episodes with $\alpha = 0.1$ and $\gamma = 0.99$. We define complexity as the smallest number of steps the agent has taken in the environment during the training episodes.

We then train 600 agents with SelfQ-Trust on a subset of mazes according to the range of the measured complexity values. The number of episodes of the underlying MDP training and that of the trust model are equal to 60. Hyperparameters are set to: $\alpha = 0.1$, $\gamma = 0.99$, $\hat{\alpha} = 0.1$, $\hat{\gamma} \in \{0.9, 0.99\}$. Results are in Figure 3.

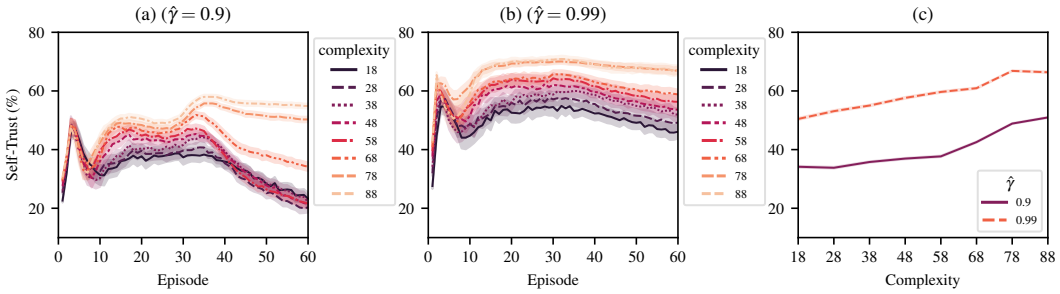

Figure 3: Sensitivity to environment complexity measured with the metric $M2$. 600 agents are trained on 8 mazes of dimension $(10, 10)$ and complexity $(18, 28, 38, 48, 58, 68, 78, 88)$. (a,b) Self-trust vs. episode with two different discount factors of trust. $\hat{\gamma} = 0.9$ in (a) and $\hat{\gamma} = 0.99$ in (b). (c) Self-trust tendency per agent vs. complexity in both cases. All curves show an estimate of the central tendency (mean estimator) and 95% confidence interval

Figure 3(c) show that the beginner's overall self-confidence tends to increase with the complexity of the task being learned. Moreover, the closer the trust discount factor is to 1, the greater the overconfidence effect. Shall the $\hat{\gamma}$ factor of SelfQ-Trust algorithm be able to measure the overconfidence effect studied by Sanchez & Dunning?

We also observe on Figure 3(a) and (b) that the overconfidence peak occurring at the outset of learning is not exclusive. Depending on the trust discount factor and the level of complexity, one or two more peaks appear afterwards. This repetition of overconfidence peaks at the beginning of learning has not been reported in any of the social science research we have studied, and may constitute a new hypothesis. On the other hand, we wonder whether the first very sharp peak - which always appears at about the same episode in our experiments - is not due to a bias of our model.

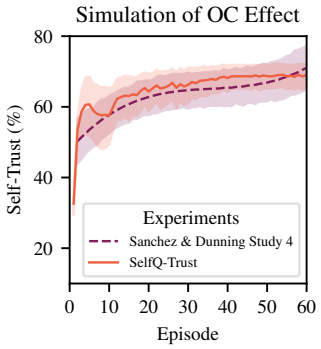

Figure 4: SelfQ-Trust reproduces the results of an experiment involving human volunteers.

Simulation of the Overconfidence Effect. One of most recent scientific result on the DKE is the measure of the Overconfidence (OC) effect by Sanchez & Dunning (Sanchez & Dunning, 2018). We extracted a dataset of the measure of the *confidence* (i.e. self-trust) from their paper and tuned the SelfQ-Trust hyperparameters to reproduce the results of their Study 4. Parameters are: Maze $(10, 10)$ of complexity $= 88$, 60 agents trained, 60 episodes of learning MDP as trust model, $\alpha = 0.1$, $\gamma = 0.99$, $\hat{\alpha} = 0.05$, $\hat{\gamma} = 0.99$. Results are shown in Figure 4.

6 Conclusion

In this paper we introduce SelfQ-Trust, a RL algorithm for measuring self-confidence of any human like entity that can be modelled by an intelligent agent trying to solve a MDP. By using a dynamic reinforcement, our method is able to adapt the magnitude of the trust reward according to the performance of the agent by itself. In doing so, we believe we have applied a recent hypothesis suggesting that dopaminergic neurons encode differences between rewards and expectations in a goal-directed system. We plan future experiments involving human experimenters and serious games to further investigate this design hypothesis. By using a distributional TD learning on the trust model, our method is able to explore self-confidence in real-time while the agent is trying to solve the MDP. For future development dealing with the multi-agent case, we plan to use the distributional Bellman operator proposed by Bellemare et al. (Bellemare et al., 2017). Our method shows results to be in line with research on the Dunning-Kruger Effect in social psychology. Moreover, we show that numerical simulations of our model allows to formulate new hypotheses; e.g. the intensity of the overconfidence effect depends on the complexity of the task to be learned.

REFERENCES

Sinan S Alsheikh, Khaled Shaalan, and Farid Meziane. Exploring the effects of consumers' trust: a predictive model for satisfying buyers' expectations based on sellers' behavior in the marketplace. *IEEE Access*, 7:73357–73372, 2019.

Sinan Aral. *The hype machine*. Currency, 2020.

Abdullah Aref and Thomas Tran. A hybrid trust model using reinforcement learning and fuzzy logic. *Computational Intelligence*, 34(2):515–541, 2018.

Natã M Barbosa, Emily Sun, Judd Antin, and Paolo Parigi. Designing for trust: A behavioral framework for sharing economy platforms. In *Proceedings of The Web Conference 2020*, pp. 2133–2143, 2020.

Zygmunt Bauman. *Liquid modernity*. John Wiley & Sons, 2013.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

Jonathan Ben-Naim, Dominique Longin, and Emiliano Lorini. Formalization of cognitive-agent systems, trust, and emotions, 2020.

Joyce Berg, John Dickhaut, and Kevin McCabe. Trust, reciprocity, and social history. *Games and economic behavior*, 10(1):122–142, 1995.

Rafal Bogacz. Dopamine role in learning and action inference. *Elife*, 9:e53262, 2020.

Sébastien Bohler. *Où est le sens? Les découvertes sur votre cerveau qui changent l'avenir de notre civilisation*. Robert Laffont, 2020.

Jin-Hee Cho, Kevin Chan, and Sibel Adali. A survey on trust modeling. *ACM Computing Surveys (CSUR)*, 48(2):1–40, 2015.

Piotr Cofta. *Trust, complexity and control: confidence in a convergent world*. John Wiley & Sons, 2007.

Will Dabney, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew Botvinick. A distributional code for value in dopamine-based reinforcement learning. *Nature*, 577(7792):671–675, 2020.

Robin IM Dunbar. Neocortex size as a constraint on group size in primates. *Journal of human evolution*, 22(6):469–493, 1992.

David Dunning. The dunning–kruger effect: On being ignorant of one's own ignorance. In *Advances in experimental social psychology*, volume 44, pp. 247–296. Elsevier, 2011.

Babak Esfandiari and Sanjay Chandrasekharan. On how agents make friends: Mechanisms for trust acquisition. In *Proceedings of the fourth workshop on deception, fraud and trust in agent societies, Montreal, Canada*, pp. 27–34, 2001.

Rory Fitzgerald et al. Ess round 8: European social survey round 8 data (2016). data file edition 2.1., 2016. https://www.kaggle.com/pascalbliem/european-social-survey-ess-8-ed21-201617.

Diego Gambetta. Trust: Making and breaking cooperative relations. 1988.

Diego Gambetta et al. Can we trust trust. *Trust: Making and breaking cooperative relations*, 13: 213–237, 2000.

Paul W Glimcher. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, 108(Supplement 3): 15647–15654, 2011.

Yuval Noah Harari. *Homo Deus: A brief history of tomorrow*. Random House, 2016.

Clay B Holroyd and Michael GH Coles. The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychological review*, 109(4):679, 2002.

Mark Hunyadi. Au début est la confiance. *Lectures, Publications reçues*, 2020.

Audun Jøsang. Trust and reputation systems. In *Foundations of security analysis and design IV*, pp. 209–245. Springer, 2007.

Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

Justin Kruger and David Dunning. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of personality and social psychology*, 77(6):1121, 1999.

Che Hui Lien and Yang Cao. Examining wechat users' motivations, trust, attitudes, and positive word-of-mouth: Evidence from china. *Computers in human behavior*, 41:104–111, 2014.

Niklas Luhmann. Familiarity, confidence, trust: Problems and alternatives. *Trust: Making and breaking cooperative relations*, 6:94–107, 2000.

Stephen Marsh. Trust in distributed artificial intelligence. In *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pp. 94–112. Springer, 1992.

Florent Meyniel, Daniel Schlunegger, and Stanislas Dehaene. The sense of confidence during probabilistic learning: A normative account. *PLoS computational biology*, 11(6):e1004305, 2015.

Esteban Ortiz-Ospina and Max Roser. Trust. *Our World in Data*, 2016. https://ourworldindata.org/trust.

Sarvapali D Ramchurn, Dong Huynh, and Nicholas R Jennings. Trust in multi-agent systems. *The knowledge engineering review*, 19(1):1–25, 2004.

Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2020.

Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artificial intelligence review*, 24(1):33–60, 2005.

Carmen Sanchez and David Dunning. Overconfidence among beginners: Is a little learning a dangerous thing? *Journal of Personality and Social Psychology*, 114(1):10, 2018.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Anitha Vijaya Kumar and Akilandeswari Jeyapal. Self-adaptive trust based abr protocol for manets using q-learning. *The Scientific World Journal*, 2014, 2014.

Yingjie Wang, Zhipeng Cai, Guisheng Yin, Yang Gao, Xiangrong Tong, and Qilong Han. A game theory-based trust measurement model for social networks. *Computational social networks*, 3(1): 1–16, 2016.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Li Xiong and Ling Liu. A reputation-based trust model for peer-to-peer e-commerce communities. In *EEE International Conference on E-Commerce, 2003. CEC 2003.*, pp. 275–284. IEEE, 2003.

A ALGORITHM

Algorithm 1: SelfQ-Trust estimating self-trust of an agent f learning to solve a MDP.

Data: trust learning rate $\hat{\alpha} \in]0, 1]$, $\hat{\epsilon} = 1$

Initialize $\hat{Q}_f(s) \sim \mathcal{N}(\mu, \sigma^2) - 1$ for all $s \in \mathcal{S}$ with e.g. $\mu = 0$ and $\sigma^2 = \frac{card(\hat{Q}_f(s))}{4} = \frac{2n+1}{4}$
Initialize intensity of trust reward $\hat{R}_f = 0$
Initialize empty list of scores for each past episode $(v_j)_{1 \leq j \leq i}$
foreach *episode* i **do**
  $s \leftarrow s_0$; $steps \leftarrow 0$
  foreach *step of episode - until* s *is* s_T **do**
    Get action a w.r.t. MDP model f and state s
    $n \leftarrow$ uniform random number $\in [0, 1]$
    if $n < \hat{\epsilon}$ **then** $\hat{a} \leftarrow 2n$
    else
      $T \leftarrow$ uniform random number
      $\in \{\frac{j}{n}\}_{-n \leq j \leq n} \cap [\max(-1, T_f(f, s) - \hat{\epsilon}), \min(1, T_f(f, s) + \hat{\epsilon})]$
      Get trust action $\hat{a}$ from T: $\hat{a} \leftarrow n(1 + T)$
    Take action a in env; learn MDP
    Learn trust: *LearnTrust*($\hat{a}, s, s'$)
    $s \leftarrow s'$
    $steps \leftarrow steps + 1$
  if $i \notin \{0, 1\}$ **then** $\hat{R}_f \leftarrow -\frac{v_i - \max_{1 \leq j < i}(v_j)}{v_i}$
  Append $-steps$ to (v_i)
  Decay $\hat{\epsilon}$
Function *LearnTrust*($\hat{a}, s, s'$)
  $\delta \leftarrow \hat{\alpha}\left[(\frac{\hat{a}}{n} - 1)\hat{R}_f + \hat{\gamma} \max_{\hat{a}' \in \hat{\mathcal{A}}} \hat{Q}_f(s', \hat{a}') - \hat{Q}_f(s, \hat{a})\right]$
  Get $X \in \mathbb{R}^{4n+1} \sim \mathcal{N}(0, 1)$; $Y \leftarrow (X_{2n-\hat{a}}, ..., X_{4n-\hat{a}})$
  $\hat{Q}_f(s) \leftarrow \hat{Q}_f(s) + \delta Y$