The Necessity of Learning for Agency

Tim Räz

March 9, 2016

Abstract

The present paper examines the notion of agency using a model from artificial intelligence (AI). The main thesis of the paper is that learning is a necessary condition for agency: Agency presupposes control, and control is acquired in a learning process. This thesis is explored using the so-called PS model. After substantiation the thesis, the paper explores the relation between agency and different kinds of learning using the PS model.

Contents

1 Introduction

The concept of agency is a tangled web at the core of several philosophical debates such as action theory, free will, and moral responsibility; see Schlosser

(2015). In these debates, the notion of agency is examined from different perspectives, and in view of different applications. In the present paper, I will examine what is involved in an agent's having control. The main thesis of the paper is that an agent has to acquire control in a process of learning. The goal of the paper is to defend and explore this thesis. I thus presuppose that an agent needs a certain kind of control over her actions, that is, control is necessary for agency. What control amounts to exactly is a contentious matter. It can be taken to mean that the action is "up to" the agent, that the reason (or mechanism) that leads to the action "belongs to" the agent.[1] It is easier to judge in particular cases whether an agent has control in this sense or not, and the judgement is easiest when control is absent because someone or something else is in control. The focus on the concrete, and on the contrast between clear (negative) and not-so-clear (potentially positive) cases will make the task of judging whether control is present or absent easier.

Traditionally, debates concerned with agency have focused on distinctively human agency; sometimes it is presupposed that only a person has the ability to act, or that only beings with consciousness, or beings with reasons, are candidates for agency. In the present paper, I will focus on a thinner notion of agency, which makes less heavy requirements. Agency is assumed to include behavior that is not explicitly intentional or guided by reasons, and it is not restricted to human agency.[2] I will therefore not require that the agent owns the reasons for her actions, but use the more liberal requirement that the agent owns the mechanism leading to the action. The discussion of agency in the present paper is based on an artificial agent that qualifies for this thin notion of agency. I will use the so-called "Projective Simulation" (PS) model from reinforcement learning.[3] This model processes its input using a network with probabilistic transitions between nodes; and, as a result of the reasoning process, it outputs a certain behavior, which is a (potential) action. The model is able to update the transition probabilities, which can be interpreted as a simple form of learning, and it can be provided with other, more sophisticated kinds of learning, as we will see. It is important to note that there is no representation of reasons or intentions in the model.

The paper proceeds as follows. I formulate and discuss the main thesis of the paper in the next section. The use of a probabilistic, artificial model in a discussion of agency can be seen as problematic for several reasons – I will discuss these worries in section 3. The PS model is introduced in section 4 (note that the more formal aspects of the model are relegated to the appendix). In section 5, I discuss the question why we should consider the PS model to be an agent. In section 6 to 9, some of the consequences of the thesis that learning is closely related to agency are explored, and we reap the fruits of employing

[1] See, e.g., Fischer and Ravizza (1998), Dennett (2003).

[2] As Elisabeth Anscombe (1957, p. 5) has pointed out, the movements of a cat can be interpreted as intentional, or even *be* intentional, without there being explicit intentions or reasons.

[3] See Russell and Norvig (2003) for an introduction to AI, Sutton and Barto (1998) for an introduction to reinforcement learning, and section 4 for references concerning the PS model.

a formal model by investigating the relation between agency and learning in more complex settings, different kinds of learning, and quantitative aspects of learning.

2 The Main Thesis

The main thesis of the present paper concerns control. Control presupposes that the mechanism leading to an action belongs to the agent, or that the agent "owns" the mechanism. But how does an agent acquire ownership of the mechanism in the first place? I propose that ownership is acquired in a learning process. Take yourself as an example. You are an agent. However, this was not always the case – you were not an agent when you were born. Somewhen between your birth and your reading this sentence, you acquired the ability to act, by pushing the world, and by noting how the world pushes back. If you have reasons to act, these reasons are your reasons because of what you have learned in interaction with the world, by learning what behavior is successful and what is not. The main thesis, then, is the following:

No Agency Without Learning: Learning is a necessary condition for agency: agency requires control, and control has to be acquired in a learning process.

The main goal of the present paper is to substantiate and explore this thesis. In a first step, I should clarify the concept of learning. What is a *learning* agent? One important feature of learning is that it is temporally extended: we learn if we take past experience into account. Thus, a necessary requirement for learning is history-dependence. If an agent's present behavior is in no way influenced by past experience, it is not a learning agent. However, history-dependence is not sufficient for being a learning agent – in particular, history-dependence is not sufficient for control, as we will see below.[4]

A second important ingredient of learning is that once we have learned, we become more successful in our behavior. It is therefore tempting to require that the result of the learning process be manifested in successful behavior. While I believe that there is in fact a connection between learning and some measures of successful behavior, we have to be careful here. For one, what has been learned need not be manifested in behavior; an agent may simply disregard what she has learned. Also, successful behavior need not be based on learning. However, under certain circumstances, learning is an indispensable part of an agent's success. We will return to this point below.

One of the problems we face when we want to explore the relation between agency and learning is that the learning process is temporally extended and mostly hidden, such that it is hard to gain insight into learning in biological

[4]The point that control is a historical notion has been made by Fischer and Ravizza (1998, Ch. 7), who argue that the ownership of control is essentially historical. They also note the importance of learning in the process of acquiring responsibility by children. See Dunjko et al. (201x, p. 5) for a formal statement of the history-dependence of learning in the context of AI.

agents. How can the thesis that learning is necessary for agency be substantiated in view of this problem? The solution that will be adopted here is to use an artificial agent, more specifically, a model from machine learning. In such models, the process of learning is simple, but more transparent than in actual, biological agents. I will examine how the model can be provided with a (rudimentary) degree of control, and, consequently, a (rudimentary) degree of agency.

3 Possible Worries

The use of models from AI in the context of agency can raise several worries. One worry concerns the use of a *probabilistic* model. This has a flavor of indeterminism, which, in turn, is thought to be incompatible with control by some compatibilists. I hope to keep the arguments in the present paper relatively independent of the question of the relation between agency and (in-)determinism. This is justified because the model on which I will focus is stochastic, but it can be emulated both in a deterministic and in an indeterministic world. The thesis that an agent has to be able to learn to acquire control should be explored independently of the fundamental (in-)deterministic nature of the world in which the agent is situated.

The method of using artificial agents has precedents in the literature on agency and free will; importantly, the method has been employed by both libertarians and compatibilitsts. Briegel and Müller (2015) use an artificial agent to argue for the compatibility of indeterminism and agency. They propose that the PS model should be considered to be an agent because its memory, which generates the model's output, has its roots in the model's own learning history. On the other side of the aisle, Dennett (2003, p. 46) notes that artificial agents have the ability to acquire their own reasons for acting in a deterministic setting, that this is a gradual learning process, during which control is handed over from the designers of a model to the artificial agent. The fact that both determinists and indeterminists note the role of acquiring reasons, or a mechanism, for agency suggests that we should explore the relation between learning and agency in its own right.

One could, more generally, doubt that the use of an *artificial* model is instructive in the context of agency. It could be thought that artificial agents are categorically different, or just too far removed, from humans and, maybe, higher animals, so that examining artificial agents is useless if one wants to get a grip on real agency. I grant that the danger of taking the model too seriously is real. We should not carelessly generalize our findings from simple artificial models to human agents. Some features of the model might be genuine phenomena of agency, others might be pure artifacts of the simplicity of the model. Ideally, the hypotheses generated on the basis of the model will be confronted with empirical results from cognitive science and psychology, where the same set of questions is studied using real, human (or animal) agents.

However, the use of simple, artificial agents has several advantages, which

are not to be had if the usual philosophical methods are employed. First, we can observe the process of how the artificial agent learns, acquires control, and acts, and we can do this in a transparent manner. We can directly inspect the different ways in which the model learns, and we can see how learning is affected by different scenarios and changing environments, because the artificial agent is a formal model. I believe that the relation between learning and agency has not received sufficient attention because the process of learning is too complex to be examined the usual philosophical methodology with its focus on single-action thought experiments.

Second, constructing an agent that has control complements the usual approach, taken in Frankfurt-type thought experiments; see, e.g., Frankfurt (1969). In these thought experiments, control and agency is examined by (artificially) restricting an agent's control. The present paper pursues a modeling strategy instead of a conceptual, subtractive strategy. The formal and quantitative nature of the model suggests interesting conceptual relations that would be harder to discover and explore in the usual philosophical methodology.

Third, the modeling approach to learning and agency provides us with an additional handle on control. We can examine how the artificial agent acquires control by gradually handing it over to the agent: The artificial agent acquires control, while the designers of the artificial agent give it up. One of the important contrasts is between properties of the model that are up to us, the designers of the model, and properties that are not up to us, and thus (potentially) up to the model.[5]

4 Introducing the PS Model

The PS model is a directed graph with probabilistic transitions between nodes; see figure 1 for a schematic representation.[6] The model receives input on the initial nodes, usually dubbed "percepts"; the final nodes are the model's output, usually dubbed "actions". Here I will usually call the final nodes "output", in order to make it clear that the output need not be an action in a substantive sense. The transition probabilities p_{ij} between the nodes are updated according to a learning rule, which rewards successful behavior.[7] The learning rule is a core component of the model, and we will encounter different versions of the rule in the course of the discussion. A second core feature of the model is its graph structure, which is also subject to modification: Nodes can be added and subtracted in some incarnations of the model.

Consider, first, the basic PS model in a simple scenario. The scenario is called invasion game; see figure 2.[8] The attacker (A) wants to invade the world; it can do so at discrete points on a line. The attacker starts at some fixed

[5] This point is also made and explored in Dennett (2003).

[6] A brief, formal introduction of the PS model can be found in the appendix; see H. J. Briegel (2012); Mautner et al. (2015) for a comprehensive introduction of the model. Figure 1 is taken from H. J. Briegel (2012, p. 3).

[7] See equation (3) in the appendix for the standard learning rule of the PS model.

[8] The scenario was first introduced in H. J. Briegel (2012); figure 2 is taken from there.

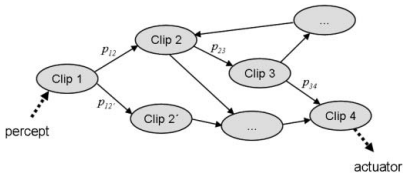

Figure 1: The PS model

position. It can move one step to the left or one step to the right. The attacker announces its moves by showing either the right or the left arrow, before moving one position to the right or to the left. The defender (D), the basic PS model, can block these attacks by moving to the grid point of the attack, i.e., by imitating the attacker's behavior. The point of the game is that, in the beginning, the PS model does not "know the meaning" of the arrows; it has to learn how the attacker's signals and moves hang together.

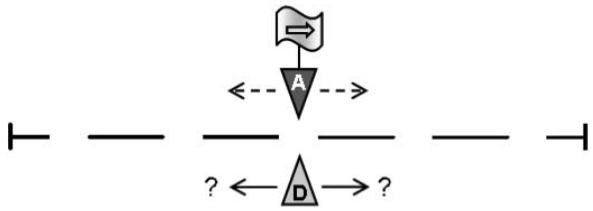

Figure 2: Invasion Game

The basic PS model's structure[9] is as follows: It has two possible percepts, $\{\rightarrow, \leftarrow\}$ shown by the attacker, and it has two possible outputs, $\{+, -\}$, moving to the right (+) or to the left (-). Both percepts are connected to both outputs. Initially, the PS model is a blank slate: the probabilities of choosing (+) or (-) when it receives $\rightarrow$ as an input are both 0.5; the same is true when the input is $\leftarrow$. This means that, initially, the model's behavior is random. The second core part of the model is the learning rule, which determines how the probabilities of the connections between percepts and outputs changes according to the success, or failure, of the model's behavior. Informally speaking, the learning rule captures the fact that successful behavior (blocking the attacker) is rewarded by increasing the weight of the corresponding edge, while non-successful behavior is not rewarded.[10]

It is possible to simulate the above scenario; usually, the typical changes in success probabilities are determined by training a large number of identical agents in the same scenario, which yields an average learning curve. After several rounds of training, we observe that the transition probabilities change

[9] See figure 3; figure 3 is taken from H. J. Briegel (2012, p. 5).

[10] The formal learning rule used in H. J. Briegel (2012) is equation (3) below.

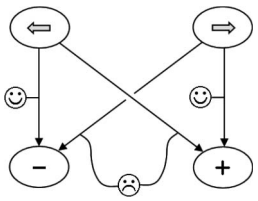

Figure 3: The structure of the PS model (the labels can be interpreted as rewarded and unrewarded outputs respectively)

such that the PS model has a high probability of blocking the attacker, while initially, the blocking probability was random. Informally speaking, this means that the model gradually "learns" to choose + when perceiving →, and - when it perceives ←, because this behavior is rewarded.

5 Is the PS Model an Agent?

The basic PS model allows us to make our investigation into the relation between agency, control, and learning more concrete. The first question is whether the basic PS model in the above scenario is a candidate agent. Consider, first, the situation of the PS model in the first round of training in the invasion game. The structure of the model has been chosen by us, the designers of the model. We have fixed the possible percepts and outputs of the model. What is more, we have also initialized the probabilities to be random. Finally, a random generator determines the first actual output of the model according to the probability given by us. This means that all aspects that contribute to the model's output are under our control, except for the random source. Thus, the resulting output cannot be interpreted as an action. It is fully determined by the structure and the random choice. We are not warranted in calling the behavior of the PS model an action in the first rounds of training.

Compare this to the situation after the PS model has undergone several rounds of training. As we have observed above, the transition probabilities can change over time, depending on the interaction between model and environment. This change in the transition probabilities can be interpreted as a simple form of learning. Typically, the PS model will enhance its blocking probability by associating the right move with the right perceptual input. The increase in blocking probability has the shape of a typical learning curve; see figure 4 below for an example of such a curve. The important point is that the transition probabilities have not been preset by us, the designers of the model. Rather, they arise as a result of the model's interaction with the environment. The model has learned which behavioral patterns lead to success; the transition probabilities can be interpreted as simple "reasons" – the *model's* reasons – for behaving in a certain way. The transition probabilities are not up to us, and they are not random; rather, they are up to the model because they have been

acquired by the model in a learning process. In this sense, we can interpret the outputs of the model as – admittedly very rudimentary – actions.

The PS model's behavior in this scenario does certainly not constitute full-blown agency. The model's structure, the "reasoning process" leading to the model's output, and the environment, are much too simple for such an attribution. However, if simplicity is our reason for refusing to call the model's behavior actions, then we may have made a step in the right direction. If the point of contention is the model's simplicity, then it is sensible to continue the investigation by making the model more complex – in the right way. The qualification is key: If the model's simplicity is the main problem, then not any kind of added complexity will do. We do not want to decrease the obvious inadequacy of the simple model by making the model's reasoning process incomprehensible.

There are still various reasons for not granting a model the status of an agent. In order to see why the basic PS model might be a step in the right direction, it is helpful to contrast it with two other models. Take, first, the following alternative model, in the setting of the invasion game. Instead of using probabilistic transitions between clips, this model is provided with hard-wired edges, such that the model always "chooses" the same, correct move in response to the attacker's announcements, i.e., we use the deterministic connections $(\rightarrow, +)$ and $(\leftarrow, -)$. This model behaves perfectly from the start in the above environment. However, its behavior is completely determined by us, the designers, and it does not have a chance to acquire anything akin to reasons. We should not attribute even a rudimentary form of agency to this model. Second, consider a model with probabilistic transition rules, but without learning rule. In this model, the transitions between percepts and output are completely random, and stay random. This model behaves successfully with a probability of 0.5. However, just as in the deterministic case, this fully indeterministic model is not a candidate agent, but merely a random generator. Note that, in contrast to these two models, the PS model steers clear both of being fully deterministic and of being fully indeterministic, or random.[11]

There is a clear contrast between the basic PS model on the one hand, and the hard-wired and the fully random model on the other. Our reluctance to accept the basic PS model's behavior as agency is different from our denying that the latter two models are candidate agents. In the latter cases, the mechanism yielding the output is not the model's. Both the hard-wired and the fully random model have not acquired the mechanism on their own. What is more, there is no possibility that these models will ever acquire such a mechanism on their own – they cannot learn. In the case of the PS model, the reason is not equally clear-cut. Initially, all properties of the model are preset by us; this is why we should not consider the model's behavior to be actions in the first few rounds of training. However, over time, the model changes its behavior through interaction with its environment, and increases its success rate. Our reservations about the status

[11] Here I follow Briegel and Müller's (2015) argumentative strategy of avoiding the "agency dilemma", van Inwagen's (1983) "consequence argument", designed to show the incompatibility of agency and determinism on the one hand, and van Inwagen's (2000) "replay argument", designed to show the incompatibility of agency and indeterminism on the other.

of the later outputs of the basic PS model, whatever they may be, are not of the same kind as in the case of the models that are unable to learn.

Again, this does not establish that the basic PS model exhibits real agency. There is still the reasonable worry that we should not call the behavior of such a simple model an action. In particular, the model has very little "wiggle room": The learning process is elementary, its outcome predictable, and the connections between percepts and outputs are fixed except for their weights. We can make this worry more concrete: The structure of the model's memory (the edges of the graph), is not up to the model, but to us, the designers. The same is true for the learning rule, which is constant, and thus also up to the designers. These difficulties are real, but they are not insurmountable. There are ways in which the model can be made more independent of the designer's choices along both of these dimensions. The model can be made more flexible with respect to its own structure, and with respect to its learning rule. In the next few section, we will see how the PS model can be provided with more learning abilities, and, consequently, more control as the scenarios get more complex.

6 Learning in Dynamic Environments

In the version of the invasion game we considered above, the environment is constant, i.e., the association between the arrows shown by the attacker, and the reward for a correct response, does not change. Let us now consider a simple example of a changing, dynamic environment, in which the "meaning" of the signs shown by the attacker changes after a certain period of time. I will then discuss how this affects the PS model and the other models we considered above, and what this tells us about agency.

The dynamic scenario is divided into two time periods. In the first period, the attacker announces its moves with arrows pointing in the correct direction, and the connections $(+, \rightarrow)$ and $(-, \leftarrow)$ are rewarded. The PS model learns to associate the announcements and the moves appropriately, and increases its success rate. Then, after 250 rounds of learning, the second period begins. In the second period, the meaning of the signs is suddenly inverted: the attacker announces its moves by pointing in the opposite direct, and the connections $(+, \leftarrow)$ and $(-, \rightarrow)$ are rewarded. This can be interpreted as a radical change in the environment.

In figure 4[12], we can see how a dynamic environment affects different versions of the PS models with different "degrees of forgetfulness", formalized as settings of the damping parameter γ; see equation (3) in the appendix. The effect of more damping is that the model gradually "forgets" what it has learned over time. We can see that the models' maximal success rate depends on the damping parameter: if there is more damping, the maximal success rate is lower. On the left hand side of figure 4, models with high damping have a low asymptotic success rate. If the environment is dynamic, a high damping parameter turns into an advantage: The models with higher damping recover faster from a change

[12] Figure 4 is taken from H. J. Briegel (2012, p. 6).

in the environment; see the right hand side of figure 4. Of course, we have to be careful in the interpretation of these results; interpreting parameter settings in terms of forgetfulness is metaphorical.

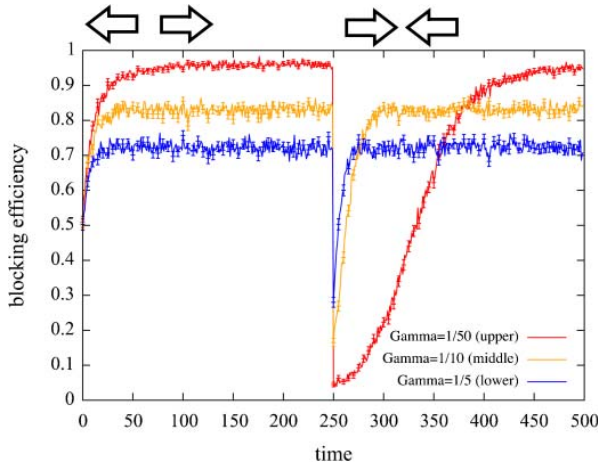

Figure 4: Dynamic Invasion Game: Learning curve

Generally speaking, the effect of the sudden change of the environment on the PS model is drastic. Consider the red learning curve in figure 4. The success rate of this model drops from a very high to a very low level. At the end of the first learning period, this PS model has a success rate of over 90%; immediately after the change in the environment, its success rate drops to under 10%, much worse than random. However, we can also see that the PS model is able to overcome the radical change in the second learning period: it can learn the meaning of the inverted signs, and after more trials, it reaches the same success rate as before the inversion; see the right hand side of figure 4. This shows that the basic PS model is not only able to learn in a static environment, but that it can also cope in a dynamic environment.

To fully appreciate the PS model's performance, it is useful to compare it with the hard-wired and the fully random model. The fully random model does equally bad in the dynamic as in the static environment, because it is not able to learn. The hard-wired model, equipped with the edges $(+, \rightarrow)$ and $(-, \leftarrow)$, is slightly more interesting. In the first period, it does better than the PS model in that its success rate is perfect. In the second period, however, the hard-wired model's success rate drops to zero. It is hard-wired in the wrong way – it does exactly the opposite of what leads to success. Thus, the PS model has an edge over these two alternative models in a dynamic environment.

Of course, the PS model is not the only model that can cope with dynamic environments. We can improve on the hard-wired model such that it does at least as good as the PS model in the dynamic environment just considered. Take the following, improved hard-wired model: Given one of the percepts, it chooses

one of the two outputs in the first round of learning. If this output is rewarded, it continues to use this connection. If it is not rewarded, it chooses the second output the next time the percept comes up. This improved hard-wired model will do fine in the dynamic environment. What is more, it is also a learning model in the sense that its output does not only depend on the last percept, but also on previous ones.

However, we would not want to attribute even a rudimentary form of agency to the improved hard-wired model, just as we did not want to attribute agency to the simple hard-wired model. The reason is the same as above: The improved hard-wired model lacks control. It has been told by us, the designers, how to react in every possible situation. It does not help that this model's output is history-dependent, i.e., that its output is not only a function of the last percept, but of other, past percepts. The point is that the output function is fixed and entirely determined by the designers; therefore, the model is not a candidate agent. This shows that learning cannot be mere history-dependence. An agent has to be able to learn "in the right kind of way". In order for this to be the case, the output function itself needs to be able to be changed depending on the environment.

Why is the PS model's mode of learning superior to the improved hard-wired model's mode of learning? The most important difference is that the improved hard-wired model has been provided with all relevant information about the environment. We, the designers, have anticipated what will happen, and programmed the reactions into the structure of the model. The improved hard-wired model is nothing more than a database, in which an output is stored for every possible history. The same is not true for the PS model, which is more autonomous, and has to learn what the appropriate behavior is as the environment changes. Thus, we should add the following as a requirement for real learning agents: The mechanism producing the action should not merely be history-dependent, but it should be able to change in interaction with the environment.

7 Learning in Complex Environments

In the previous section, I have argued that learning is more than mere history-dependence: hard-wired models, even if they are history-dependent, do not learn in the right kind of way. In the present section, I will argue that the right kind of learning is not only a theoretical requirement for agency, but also a practical necessity if we want to build models that are able to cope in more complex environments. The scenario we will examine is the so-called grid-world scenario.[13] In the grid-world scenario, models are situated in a world of finitely many grid points; see figure 5.[14]

[13]The grid-world scenario is a standard benchmark scenario to test models from reinforcement learning; see Melnikov et al. (2014) for a discussion of grid world and the PS model's performance in this scenario.

[14]Figure 5 is taken from Melnikov et al. (2014, p. 3).

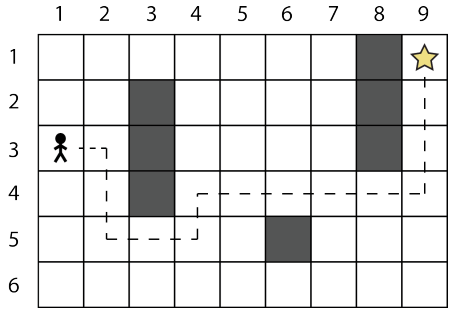

Figure 5: The Grid World Scenario

In this scenario, the models have to learn to get from the starting point to one particular square of the grid, marked with a star. The models are only rewarded if they find the target, but not for the other moves. During the game, a model has the four basic options of moving up, down, right or left. A model receives coordinates as percepts, e.g., $(x, y) = (2, 3)$. Every move is counted as a time step. If a model hits the boundary or a wall, it stays where it is, but the time step is counted. Once a model finds the target, it is rewarded and returned to its initial position, and the next trial begins.

Let us examine how different models cope in this scenario. Consider, first, a hard-wired model. Such a model could be programmed to just systematically go through all grid points by moving out from the starting point in a spiral motion. Once it has found the target, and been rewarded, it uses its database, where the shortest paths to all possible grid points are stored, and, from trial two on, always uses the shortest possible path. This hard-wired model does not learn anything; it just does what it is told. It is not a candidate agent. What may be even more important is that the hard-wired model has a practical drawback. On a naive implementation, the memory that is necessary to store all possible paths will be quite large. The model needs to store the optimal path for every possible target position; if we choose, for the sake of simplicity, a quadratic grid of size n^2, it will, on a brute force approach, need a memory for $n^2 - 1$ paths. The situation gets even worse as the environment gets more complex. Not only can we make the grid larger, but we can also add and subtract walls at all grid points. The number of possible scenarios, and consequently, of paths to be stored, now grows exponentially, and the memory of the enhanced model will have to grow accordingly.[15] The memory necessary to build the hard-wired model explodes and becomes a practical impossibility, even for grid worlds of modest size.

Thus, in this scenario, a clever, learning model becomes a practical necessity. One of the difficulties to be overcome in scenarios such as grid world is that the

[15]There are e^{n^2} possible grid worlds of size n^2 if walls can be added to all grid points. Note that not all of these scenarios will be viable.

reward is delayed. The reward is only handed out after many basic moves, and irrespective of how exactly the complex search was carried out. The model does not get any guidance on how to carry out the search, it is only rewarded for completing it. How is a model supposed to learn how to carry out the search more efficiently? One way of solving this problem, proposed in Melnikov et al. (2014), is to supplement the PS model's learning rule with the so-called "glow mechanism". The idea is that not all basic moves should be rewarded equally, but higher rewards should be given to basic moves that were carried out more recently. The glow mechanism works as follows. The model memorizes the sequence of basic moves in that, if a certain basic move is taken, the corresponding edge is set to glow. The glow decays over time at a constant proportion. If the model finds the target, the reward is distributed to the basic move leading to success according to their glow: more glow means more reward. The rationale behind this reward scheme is that the more recent moves contributed more to the successful completion of the path than those lying further in the past.[16]

One of the main findings of Melnikov et al. (2014) is that this mechanism works. On the basis of the glow mechanism, the PS model is able to learn to find the target after a reasonable number of training runs. The average number of basic steps needed to find the goal decrease from several hundred in the beginning of training to around 15 steps. The performance of the PS model is comparable to other models of reinforcement learning. The exact quantitative performance depends on the exact form of the learning rule. What is remarkable about the PS model in the grid world scenario is that the path is not "given" to the model in any obvious sense, but the model genuinely "finds" the path on its own; the path crystallizes in the course of the learning process. The complex "action" of choosing a short path towards the goal itself is not controlled in an obvious sense by the designers of the model.

In sum, we can learn three main lessons form the PS model in the grid world scenario. First, in contrast to a naive hard-wired model, the PS model is not provided with the information of how to behave in all possible situations, but it genuinely finds the solution to the problem on its own; a good solution path crystallizes in a learning process in interaction with the environment. Second, as scenarios get more complex, learning is no longer a luxury, but a practical necessity; from a certain point on, building a model without a certain degree of control becomes infeasible because resources such as memory are limited. Third, the PS model achieves this by filtering out the relevant information: Not all the basic moves contribute equally to its success. This is what the glow mechanism achieves.

8 Learning to Generalize

I have noted in section 5 that the basic PS model is limited in several respects. One limitation is that the model's structure, its graph, is fixed; see figure 3. From the perspective of agency, the graph structure is not "up to" the model;

[16] See the appendix A, especially equation (4), for the formal details of the glow mechanism.

consequently, the output of the model should not be considered to be an action, insofar as the output is due to the graph structure. The same observation applies to the learning rule of the basic PS model: the learning rule does not change, and, therefore, limits the model's ability to learn.

The model's inflexibility also puts practical limitations on the kind of problem it can solve. In the present section, we will see environments in which the basic PS model is inefficient, or even entirely inadequate; there are cases in which the basic PS model is never able to produce an output on the basis of an updated probability, i.e., it is not able to act. This motivates a modification of the model. The modified model has the ability to adapt its structure to the environment: it has the ability to add edges to its own structure under certain conditions. The model is provided with a transition rule not only for probabilities, but also for these modification. In this way, the design of the model's structure is "handed over" to the model.

More specifically, the enhanced PS model here has the ability to "generalize".[17] Intuitively, a model with generalization can learn from percepts that it has not encountered before. This is possible if the model can recognize that a new percept is similar to percepts which it has encountered before. Consider the following environment, which is a modification of the invasion game. The attacker announces its moves using arrows $(\rightarrow, \leftarrow)$. Additionally, the arrows are now colored either red or green. Thus, there are now four percepts. The outputs are, again, moving right or left. In this environment, generalization is a relevant ability: We can create different scenarios, in which some properties of the percepts are relevant, while other can be neglected. In one scenario, only the shape of the arrows matter. The model has to learn to move to the right when it is shown a (green or red) right arrow, and to the left when it is shown a (green or red) left arrow. In a different scenario, the model should learn to pay attention to colors only, such that it moves to the right when it sees green, and to the left when it sees red, irrespective of the direction of arrows.

The basic PS model is able to cope in this environment. Consider the model depicted in figure 6.[18] Assume we are in a scenario where only the direction of the arrows matters, while the colors of the arrows are irrelevant. Over time, the basic PS model will build up the appropriate probabilities for all percepts. However, the model is not able to "recognize" that the distinction between red and green is irrelevant. The shapes are in no way correlated by the model.

In contrast, a PS model with generalization is able to "recognize" that certain distinctions are relevant, while others are not. Such a model can modify its structure along the following lines. An additional layer of classification clips can be added between percept and output clips. In the present scenario, there are two kinds of categories, color and shape. For each property, red, green, left, and right, a clip can be added: If the model is given a percept, say, a green left arrow, it adds the corresponding categories, which provides it with the option of classifying this percept as green or left; see figure 7.[19]

[17] The following discussion is based on Melnikov et al. (2015).

[18] Figure 6 is taken from Melnikov et al. (2015, p. 3).

[19] Additionally, a "fully general" clip, #, which comprises all of the above categories, can be

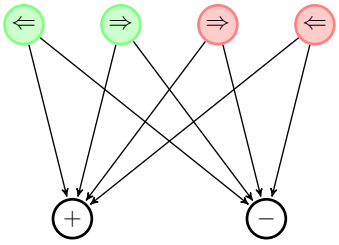

Figure 6: Generalization: basic model

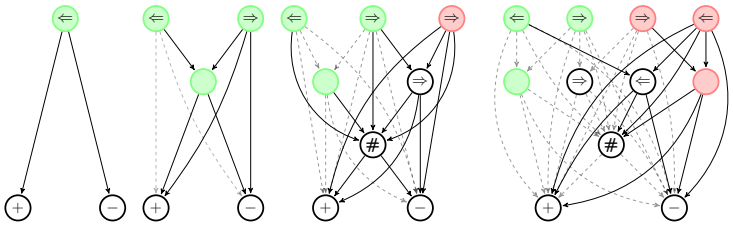

Figure 7: Model with Generalization

The model with generalization has the ability to learn which categories are relevant in a particular scenario, e.g., that only the color of arrows matters in a particular situation. All we have to presuppose is that the model knows the different categories, e.g., color and direction, and has the ability to recognize percepts as falling into the different categories. The model learns on its own which classifications are relevant and should be associated with particular actions. It can be shown that, in this scenario, it is an advantage if the model has the ability to generalize: The model with generalization learns faster than the basic model.

An even more striking result about generalization is that in some scenarios, models without generalization are unable to learn at all, while models with generalization do fine. The scenario is such that there are, again, two kinds of categories, direction and color. Each percept has one of four directions: left, right, up and down. However, the percepts come in infinitely many different colors, and in each round, the model in question gets a percept with a new color, such that no color is repeated twice. Thus, while the model is exposed to percepts with the same direction many times, it never sees two percepts with the same color. This scenario is dubbed "never-ending color scenario".[20]

We can now compare the basic model and the model with generalization with respect to this scenario. The structure of the basic model is such that

added in a third layer. Figure 7 is taken from Melnikov et al. (2015, p. 4).

[20]See Melnikov et al. (2015, Sec. IV) for a full description of the scenario.

each percept is directly related to the output clips of moving in one of the four directions. However, each percept is entirely new, it has never been perceived before. Therefore, the basic model's choice between moving in one of the four directions is random.[21] Compare this to the model with generalization. Over time, this model will learn to connect the direction categories with the correct action with high probability. The ability to categorize does not merely give the model with generalization an edge over the basic model. In the never-ending color scenario, the ability to generalize makes the difference between candidate agency and non-agency: The basic model is never able to produce an output that is not random. The model with generalization, on the other hand, is able to draw on its experience, and exhibits a (still rudimentary) form of agency. In the never-ending color scenario, generalization makes the difference between agency and non-agency.[22]

At first sight, it might seem that the never-ending scenario is extreme in that we never allow the same color to return twice; on this basis, we could question the real-life relevance of the scenario. However, in reality, we never encounter the exact same situation twice, we only encounter situations that are very similar some relevant respect. Our perceptions are always classified in one respect or another – encountering the exact same perceptual input twice is only possible in an abstract, formal setting. We thus should not dismiss this scenario as implausible or contrived prematurely.

In sum, we have seen that it is possible to provide a model with a certain degree of control over its own structure, such that it is able to deal with classification tasks. Additionally, we have also seen that such a modification can be a necessity in certain scenarios in that a model without this ability is not able to cope at all. Of course the model's control is still limited. In particular, the model is not able to form its own categories. Also, the model has to be provided with the ability to recognize percepts as falling under one or the other of the categories. Melnikov et al. (2015, Sec. V) explicitly point out these inflexibilities and discuss ways in which these limitations might be overcome; for example, classification could be made independent of fixed semantic information. In this way, the task of forming a classificatory system would be turned over to the model.

[21] Note that there is a 0.25 chance that the model makes the correct move, and the probability of the corresponding choice is raised according to the updating rule. However, the model is never able to apply this "knowledge", because it only perceives every percept once. Its performance never rises above the level of random behavior.

[22] This does not mean that the basic model is worthless. It is important to note that if the scenario were slightly different, in that colors come up more than once, the basic model would be able to act as well. In this sense, it is appropriate to call the basic model a "potential agent": There is a slight modification of the scenario that enables the model to perform at least one action.

9 Learning Speed Matters

In the two previous sections, we have seen that different kinds of environment necessitate different kinds of learning – forgetfulness, and being able to recognize similarities, can be a blessing. In the present section, we will examine a further quantitative aspect of learning with a direct impact on agency: We will see that the speed at which a model learns can make the difference between agency and non-agency.

In the scenarios we have considered so far, the environments were so-called round-based environments. In round-based environments, the environment provides an input and then lets the model process the input, waiting until the model provides an output. Everything is on hold during the deliberation process. The performance of models in round-based environments is measured on the basis of rounds, and does not take into account how long it takes a model to reason from input to output. This is no problem as long as the models' deliberation process is simple, as it is the case with the models we considered above: deliberation only takes a few steps on a small graph, and negligible time. However, the deliberation process can get more complex and take up a non-negligible amount of time as the models themselves get more complex. Deliberation speed may then influence learning speed and, consequently, agency.[23]

Take the following scenario. Recall that, in the dynamic invasion game from section 6, it took the basic PS model approximately 100 learning rounds to achieve its maximal success probability. The deliberation speed of the basic PS model is very high – it only takes one step in the network – and can, therefore, be neglected. What would happen if the deliberation speed of the model was, say, 300 times lower? Such a slow model could not produce an output in the environment from which it has received the input. In this sense, it would never be able to act: The environment has changed before the model was able to learn anything about it.

More generally, here is why deliberation speed matters. Take two models X, Y that have the same input-output profile and are thus indistinguishable in a round-based environment. Assume that the environment is dynamic and non-recurrent, i.e., once the environment has changed, it does not change back to a previous state. Assume, further, that the internal deliberation speed of X is small in comparison to changes in the environment, but the internal deliberation speed of Y is large in comparison to changes in the environment. How do models X and Y perform in this environment? X is able to learn and cope in this environment, while Y is not able to learn at all. Consequently, X is, at least potentially, an agent, while Y is not. This shows that if we take internal

[23] Cases where internal deliberation speed matters have been explored in the PS framework; see Paparo et al. (2014). Paparo et al. distinguish "passive scenarios", where only the agent's input-output profile, its round-based behavior, is taken into account, while the internal deliberation speed is ignored, and "active scenarios", where deliberation speed is also taken into account. Paparo et al. show a quadratic speed-up in learning can be achieved by a PS model that has quantum properties in comparison to a model that does not employ quantum properties. Here I will not go further into the details of this particular result, although it would be interesting in itself.

deliberation speed into account, and the deliberation speed of a model is slower than changes in the environment, then this model is unable to learn, and, therefore, unable to act. A quantitative difference between models, their learning speed, can lead to the qualitative gap between agency and non-agency.[24]

In what kind situation might a difference in reasoning speed actually occur? One way of making the abstract argument more concrete is if the changes in the environment are brought about by *other agents*. Take a scenario where several models compete for the same reward. The model that wins the competition might harvest the reward, which prevents slower models from learning, because if there is no reward, there is no learning. In this scenario, the fastest model is the pace-maker for changes in the environment. This scenario has several connections to situations that might be of actual scientific relevance. Take, for example, a classroom situation, in which a teacher asks questions and rewards only those students that come up with the answer first. If only the same, fast student that is rewarded, while the other students remain empty handed, the slower students might not learn at all and might lose interest, while the fastest student gets faster and faster.

In sum, comparing different, concrete models in different, dynamic scenarios brings interesting, quantitative aspects of the relation between learning and agency to the fore. In particular, we saw that the deliberation speed of a model can make all the difference between that model's being able to learn, and not being able to learn at all.

10 Conclusion

The main thesis I proposed and defended in the present paper is that there is a close connection between agency and learning; more specifically, I claimed that learning is necessary for agency, mediated by control. In order for an agent to own the mechanism producing her actions, she has to acquire this mechanism in a learning process. I explored this thesis with the help of the PS model. I contrasted the PS model with models that are not able to learn and, consequently, unable to act; this contrast suggested that, while the PS model exhibits a very rudimentary form of agency, the reservations we may have about this model are no longer of a principled nature, but a matter of degree.

I then explored further the relation between agency and learning with the help of the PS model. The examination revealed connections between agency and learning that would be hard to obtain on the basis of the usual philosophical methodology of single-action thought experiments. We saw that agency becomes salient in environments that are dynamic and complex, such that an needs to be able to learn the best course of action. Furthermore, while it is clear that the basic PS model is not the final word on real agency, its concrete structure suggests how the model can be provided with more and more control, e.g., by providing it with the possibility of modifying its own structure. This reveals a confluence between the philosophical question of how agents can acquire control,

[24] This argument was first proposed in a slightly different form in Paparo et al. (2014, p. 3).

and the engineering task of building models with more and more autonomy. Finally, we saw that there is a close connection between the internal reasoning speed and a model's ability to actualize its agency.

In the present paper, I have drawn on the PS model in order to substantiate, and explore, the thesis that learning is a necessary condition for agency. The use of models from AI is fruitful because these models make it possible to articulate and test subtle relationships between aspects of learning, memory, and agency in a perspicuous manner. The methodology of directly testing philosophical theses using the PS model will be pursued further; the prospect of examining learning in multi-agent scenarios is particularly exciting. Of course, it would be desirable to test the ideas articulated in the present paper using other models form AI as well. However, these ideas can also be confronted with empirical findings about animal and human agency, by drawing on cognitive science, psychology, and biology.

A The Formal PS Model

The core structure of the PS model[25] is a directed graph, together with an assignment of probabilities to the edges. The graph is defined on a set $\{c_1, c_2, ...\}$ of vertices called clips. The set of vertices can be differentiated into input clips $s_1, s_2, ...$, output clips $a_1, a_2, ...$, and internal clips; all of these can be provided with further structure if needed. Edges are written as (c_i, c_j), which should be read as $c_i \to c_j$. In order to define the transition probabilities assigned to the edges, we first define the function $h^{(t)}(c_i, c_j)$, the so-called h-value, which is a time-dependent edge weight. Usually, the PS model is initialized as a "blank slate", i.e., we set $h^{(0)} = 1$ for all edges. The h-value then yields the conditional probability of transitioning from clip c_i to c_j:

$$p^{(t)}(c_j|c_i) = \frac{h^{(t)}(c_i, c_j)}{\sum_k h^{(t)}(c_i, c_k)} \tag{1}$$

This means that the probability of going to c_j, given that we are at c_i, is the h-value of the edge (c_i, c_j) relative to the sum of all h-values of the outgoing edges of c_i. Put differently, we normalize the h-value to get the conditional probability. Note that, initially, all transitions are equiprobable, i.e., the transitions are random.

The PS model is formulated within the paradigm of reinforcement learning. One way in which the model can be taken to learn is by updating the transition probabilities according to rewards; the rewards, in turn, are assigned depending on the model's outputs, which yield a more or less successful interaction with the environment. The simplest learning rule that implements this idea modifies the h-values as follows:

$$h^{(t+1)}(c_i, c_j) = h^{(t)}(c_i, c_j) + \lambda \tag{2}$$

[25] See H. J. Briegel (2012); Mautner et al. (2015) for an introduction of the model.

The parameter λ can be interpreted as the reward, which is provided by the environment. It is non-negative, where $\lambda = 0$ means that a certain output is not rewarded. We only modify those h-values that were used in the random walk resulting in a particular output. If $\lambda > 0$, the h-value increases in the time step in question, and the probability increases accordingly. The standard PS model uses a learning rule adds an additional damping or "forgetfulness" parameter:

$$h^{(t+1)}(c_i, c_j) = h^{(t)}(c_i, c_j) - \gamma(h^{(t)}(c_i, c_j) - 1) + \lambda \tag{3}$$

In this equation, a damping term with parameter γ is added. The damping parameter $0 \leq \gamma \leq 1$ decreases the h-values of all edges in every round, such that the model "forgets" what it has learned in previous rounds. Obviously, (2) results from (3) if we let $\gamma = 0$. The damping parameter has the advantage that a PS model with damping is able to adapt faster to changing environments; however, it has the drawback of limiting the optimal success probability below 1. This is due to the fact that the model continually forgets positive rewards even if the environment is constant.

The PS model in the grid world scenario uses the glow mechanism.[26] The glow mechanism assigns a glow to edges that are used in the course of the "reasoning process". If an edge is visited, glow is set to 1, and it decreases at a constant rate over time. In order to implement the glow mechanism, the learning rule is modified by adding a glow function g:[27]

$$h^{(t+1)}(c_i, c_j) = h^{(t)} + g^{(t)}(c_i, c_j)\lambda \tag{4}$$

The glow function g is updated according to the following rule, using the glow parameter η:

$$g^{(t+1)}(c_i, c_j) = g^{(t)}(c_i, c_j)(1 - \eta) \tag{5}$$

The efficiency of the glow mechanism depends, in particular, on the setting of the glow parameter, η. η takes values between 0 and 1, where 0 means that glow does not decrease at all, while 1 means that glow disappears after one time step. Extreme settings do not lead to successful behavior. More concretely, in the grid world scenario shown in figure 5, if we set η to 0, the model learns at a very slow rate – quantitatively, the model needs more than 800 basic moves on average to find the target after 100 training runs. Thus, it is very hard, or even impossible, to learn a path if all, or none, of the basic moves are remembered. For the above form of the learning rule, $\eta = 0.07$ can be shown to be optimal in this scenario.

References

Anscombe, G. E. M. 1957. *Intention.* Oxford: Basil Blackwell.

[26] The following discussion is adapted from Melnikov et al. (2014); see this paper for a detailed exposition and discussion of the glow mechanism.

[27] I have left out the "forgetfulness parameter" from equation (3) for simplicity's sake.

Briegel, H. J. and T. Müller. 2015. A chance for attributable agency. *Minds and Machines* 25(3): 261–79.

Dennett, D. C. 2003. *Freedom Evolves*. Penguin.

Dunjko, V., J. M. Taylor, and H. J. Briegel. 201x. Framework for learning agents in quantum environments. ArXiv xxxx.

Fischer, J. M. and M. Ravizza. 1998. *Responsibility and Control*. Cambridge University Press.

Frankfurt, H. 1969. Alternate Possibilities and Moral Responsibility. *Journal of Philosophy* 66: 258–78.

H. J. Briegel, G. D. l. C. 2012. Projective simulation for artificial intelligence. *Scientific Reports* 2(400).

Mautner, J., A. Makmal, D. Manzano, M. Tiersch, and H. J. Briegel. 2015. Projective simulation for classical learning agents: a comprehensive investigation. *New Generation Computing* 33(1): 69–114.

Melnikov, A. A., A. Makmal, and H. J. Briegel. 2014. Projective simulation applied to the grid-world and the mountain-car problem. *Artificial Intelligence Research* 3(24).

Melnikov, A. A., A. Makmal, V. Dunjko, and H. J. Briegel. 2015. Projective simulation with generalization. ArXiv:1504.02247v1.

Paparo, G., V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel. 2014. Quantum Speedup for Active Learning Agents. *Physical Review X* 4(031002).

Russell, S. J. and P. Norvig. 2003. *Artificial Intelligence - A Modern Approach*. New Jersey: Prentice-Hall.

Schlosser, M. 2015. Agency. The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.), http://plato.stanford.edu/entries/agency/.

Sutton, R. and A. Barto. 1998. *Reinforcement Learning*. MIT Press, 1st ed.

van Inwagen, P. 1983. *An Essay on Free Will*. Oxford University Press.

———. 2000. Free will remains a mystery: The eighth Philosophical Perspectives lecture. *Philosophical Perspectives* 14: 1–19.