

Abduction, Deduction & Causal-Relational Models

Kristinn R. Thórisson^{1,2} and Arthur Talbot^{1,3}

¹ Center for Analysis and Design of Intelligent Agents, Reykjavik University, Iceland

² Icelandic Institute for Intelligent Machines, Iceland

³ Ecole normale supérieure Paris-Saclay, France

thorisson@ru.is, arthur.talbot@ens-paris-saclay.fr

Abstract

We consider the role of reasoning in a resource-limited controller that explicitly and continuously models its environment, and uses these models as a basis for its prediction and action. Several important features of such *cumulative modeling* are identified, with an emphasis on how abduction and deduction can be used to continuously prune and refine the model set towards representing true causal relations between observed and manipulated variables.

1 Introduction

All cognitive (and computational) processes are restricted by time and energy, and while in some cases we may ignore these constraints when looking at details of a cognitive system’s operation, one must avoid oversimplifying to the point of ignoring so fundamental limitations [12].

We see an embodied cognitive system as a *goal-driven learning controller* of a body and its task-environment. By “goal-driven” we mean that the system actively seeks state spaces that meet certain conditions (i.e. goals¹); by “learning” we mean that experience can be used to direct subsequent behavior in favor of such goal seeking. To target artificial *general* intelligence we must assume these are complex goals composed of a set of sub-goals, each involving a relatively large set of variables spanning potentially long periods of time, whose successful achievement requires diligent tracking of time, typically at multiple orders of magnitude.

The environments we are interested in are sufficiently complex and dynamic as to regularly generate novel states, that, due to the enormous size of all potential combinatorics between the environment and the goals and sub-goals such a controller might be required to handle over its lifetime, makes it practically impossible to code the controller and its permissible states all by hand. To achieve its goals, such a controller must *learn* to identify the relevant variables it should observe and control online, *on the job*.

¹We use the term “goal” in a broad sense, rather than narrow, i.e. a goal can be anything from a collection of loosely-related variables with an broad range of permissible values to a single variable with a specific value and low error tolerance.

As shown by Conant and Ashby’s *Good Regulator* theorem, every good controller of a system must contain a model of that system [1]. By “good” the authors mean a controller that achieves its goals, as measured in light of its specification (i.e. the difference between the target state of the world, as specified by a goal, and the actual state of the world, measured with the appropriate methods). One challenge such a controller faces is finding the best balance between identifying and retaining (in memory) only variables that matter to the task at hand and other related variables that (e.g. in the future) might interfere or help with tasks/goals. Spending a lot of time on identifying and modeling seemingly relevant variables may end up being wasted time, should those models never be needed. Nevertheless, even in an environment with a large range of variables spanning broad variability, sufficient experience with the relevant variables on separate instances of tasks undertaken should enable a good modeler to make models that allow it to predict and achieve goals with increasing accuracy and proficiency. As the quality of models increases, the frequently of surprises are reduced, and probability compromised controller performance is lowered.

Here we describe the use of explicit bi-directional models in capturing regularities in a task-environment, and how abduction and deduction can work in tandem to build up a model set that can inform a cumulative modeling process. After a brief review of prior work, in Section 3 we show that from the creation of models that can be used for deduction, prediction will also be enabled, and by using such models in reverse via backward chaining, planning via abduction becomes possible. In Section 4 we present arguments for how knowledge which is useful for prediction may be insufficient for achieving goals, requiring additional explicit causal relations to be represented. We show how using a single model for two purposes, in a bi-directional role—namely, for both deduction and abduction—and retaining only those that work in both roles, produces a model set that captures true *causal relations* between observable variables in the environment. The knowledge thus built up is best described as causal-relational.

2 Related Work

Trying to create models that target causal relationship is not new. For instance, Nguyen-Tuong et al. [3] review aspects of model learning in the field of robot control. They consider models that are created via classical regression tech-

niques and error measurement, and are only designed as an help to determine missing data. While their goal overlaps to some extent with ours, a major difference is in the scope of the models; whereas their target scope is limited to human-robot interaction, our approach is entirely model-based and model-driven.

Vaandrager [11] also addresses the task of model learning. He presents state machines that are created via membership queries. With enough queries, it is possible to create a machine that will be able to fully determine a system behavior. Similar to Nguyen-Tuong et al. [3], Vaandrager’s goal overlaps to some extent with ours in that models are used as a core representation, predicting future states of the world from current state. However, his modeling process is radically different, being based on state diagrams, which results in overly complex models. Further, his approach is limited to deterministic and completely known worlds.

As mentioned, we are interested in goal-oriented agents in diverse task-environments such as the physical world. The nature of our target environments must in the limit be considered non-deterministic, as we assume an agent never observes or knows fully all variables and their relations. Our approach to task-environments has been described in prior papers [10; 9].

Highly relevant to the present discussion is Pearl’s work on theoretical and practical aspects of causation [7]. Pearl argues that human-level intelligence requires modeling cause and effect, and because conventional machine learning does not, certain problem solving is forever out-of-reach using only those methods [8]. One of the key ideas he develops is the do-calculus [5], which provides rules for determining causal relations between facts (observations). This allows direct and indirect effects to be identified. In [6] (p. 36) Pearl states: “... *causality deals with how probability functions change in response to influences (e.g., new conditions or interventions) that originate from outside the probability space, while probability theory, even when given a fully specified joint density function on all (temporally-indexed) variables in the space, cannot tell us how that function would change under such external influences. Thus, ‘doing’ is not reducible to ‘seeing’, and there is no point trying to fuse the two together.*” This work represents a probabilistic theory that directly supports our work here.

3 Cumulative Modeling

A learning controller placed in a complex world, where plans are necessary for achieving goals, may proceed by constructing explicit models of the environment. Models are created based on experience, i.e. observed variables and their relations, and the process of creating models involves testing them to estimate their *usefulness*: The more accurately they help the controller achieve its goals the more useful they are. We refer to system that creates models continuously and incrementally this way as a CUMULATIVE MODELER. A cumulative modeler is thus a controller that models its environment in a targeted fashion so as to support its activities intended to achieve its goals in the environment.

That the modeling is *cumulative* means that new models

are *integrated* with prior models, so as to create a greater whole as the models accumulate, ideally creating a complete model set that models the controller’s environment to a sufficient level to allow it to achieve its top-level goal(s). Incomplete model sets are inevitable for any complex phenomenon or environment like the physical world, since at any point in time unobservable variables are a given.

An example of an implemented prototype cumulative modeler is found in Nivel et al. [4]. The main manner in which this modeler handles variability and dynamics in a task-environment is via *prediction*, where values and value ranges of variables are modeled explicitly to enable generating potential future value ranges from any state. Models created via our cumulative modeling process can be used to predict the behavior of the environment (including the controller’s own body), and becomes the main basis on which actions are chosen/generated. The most common initial state when using deductive prediction, for any embodied controller in a complex environment, is the *now*. For this state, to know “what’s coming”, predictions are done continuously and consistently. The closer the models match actual relations of variables in the environment the more useful they are for predicting it, and the more efficiently and effectively the controller may achieve its goals by using these to predict the outcomes of its own actions on the world.

3.1 Deduction for Prediction

For any complex environment, prediction from initial conditions or state S_{init} is a key method for determining what will happen next. The initial state could be any point in time t_1 , e.g. $t_1 = now$, or a hypothetical state of the world. For prediction, *relevant models* are initialized with the values of the *relevant variables* (e.g. at time t_1), and then traced from antecedent to subsequent states, producing future states, in a “forward-chaining” fashion. Since models capture relational transformations in the environment they contain a sequence of states where one (α) is antecedent and the other (β) is subsequent. Models may also represent forces (e.g. gravity or energy) with relevant calculations. The result of forward-chaining—should they go on for a sufficient duration without interruption—will tell the controller what is likely to be the state at some point in time $t_1 + \Delta$. What-if scenarios can be run by varying the initial conditions, e.g. an action that the controller can take, and observing the change in the predicted future state.

A model may be created when the controller observes an event α and a following event β . The model can be seen as an hypothesis that the observed event α caused the observed event β , so that when observing again an event α in the future, this model will predict that β will be observed. (For example, one can observe that after eating an apple, one is not hungry anymore.) Models that do this prediction the best (work better than others) are kept and used, others are deleted. When a better model comes along it will be preferred over the old one(s).

The forward-chaining of models is thus a form of deduction, because as far as the models are concerned – and at any point in time these are essentially the best knowledge about the observed environment that the controller has at its dis-

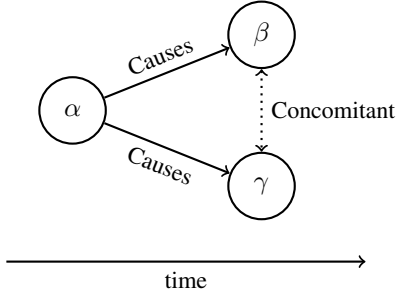


Figure 1: Relations between three variables α , β and γ .

positional – they are used to compute “inevitable” conclusions from any state. A complicating factor is of course that typically, for any controller that has been modeling its environment for some time, there exist multiple models that may be relevant for any S_{init} , and thus the controller needs a method for managing this set at any point in time: which ones to use, which ones to trust, etc.

3.2 Abduction for Planning

The models thus created are always created to respond to certain goals or goal: If an agent has no top-level goal it will have nothing to do and nothing to learn, as it has no reasons to do so. When there is a goal to be attained, the controller will try to find a way achieve it using existing models, and it creates new ones if the present ones don’t suffice. To achieve goals it will initiate the opposite mechanism of prediction: if A causes B , and there is a model that predicts B from A , the system will try to make situation A happen – the model’s cause will become its sub-goal. In the case of the apple, the system will know that if being satiated is a desired state, eating an apple will achieve the goal. A controller that has been modeling for a while in an environment will often have many ways to achieve its goals.

4 Producing Causal-Relational Models

The bi-directionality of the models moves them towards capturing true causal relations² between variables of the environment. To see why, consider the situation where a cause α has two effects, β and γ (figure 1). We assume that to the modeler α appears before β and γ , but β and γ appear together. Four models could be used to describe what is seen every time we observe these variables:

1. Model M_1 : $\beta \Rightarrow \gamma$
2. Model M_2 : $\gamma \Rightarrow \beta$
3. Model M_3 : $\alpha \Rightarrow \beta$
4. Model M_4 : $\alpha \Rightarrow \gamma$

²For x to be a deterministic necessary and sufficient cause of y in conditions z , y must disappear in the absence of x and appear in the presence of x , given no changes in z .

Any of these models will predict observed events correctly: If you see β you will see γ , and vice versa; if you see α you will see β and γ . They can be combined to cover the full experience with all variables: M_3 and M_1 ; M_4 and M_2 ; M_3 and M_4 . However, not all of them represent the actual causal relations, and not all of them can be used to achieve goals in the α - β - γ domain: If you want to stop seeing γ it does not help to remove β , or vice versa – to remove either β or γ the only variable that will help achieve the right sub-goal is α , as specified by M_3 and M_4 . Thus, when each of these models is used for *both* prediction and goal achievement, models M_1 and M_2 will be deleted due to their incorrect predictions. What remains from such a process are only models that capture causal relations in the domain, to the extent that this can be represented as relationships between *observable* variables.

While it would be possible to represent forward relations and inverse relations separately, rather than combining them in a single model as proposed here, bi-directional models makes the representation more compact. Furthermore—and perhaps more importantly—it is also more likely to make the models more *useful*. To illustrate why, consider a circuit with two buttons A and B , and a light C . When you press A , the light turns on. When you press A again, the light turns off. Button B does nothing. Agent X then witnesses someone pressing simultaneously both buttons many times. (For this example let’s assume the agent already knows that “buttons can turn on lights” and that “lights don’t make people push buttons”.) From its observations X can create at least one model, namely $A + B \rightarrow C$. Alternative models could also be generated, representing the hypotheses that only a subset of the events are related, i.e. $A \rightarrow C$ and $B \rightarrow C$. Given these three models and a goal of turning the light on ($C = true$) when it’s off, a bi-directional model generated from observation can be read backwards to infer the correct subgoal, namely to push (one or two) buttons.

Of course, no amount of a-priori reasoning will help determine which of these is *most* useful: Falsifying one of the alternative models by pressing only A or B would be a good strategy. Another would be verify that the two-buttons-pressed model works. In any case, we see here how bi-directional causal-relational models help the controller achieve its goals from observation alone. If the agent just wants to turn the light on and be done with it, the two-buttons-pressed model will suffice: Pressing both buttons is the best strategy since, out of the three models, this is the model that is most consistent with observable evidence and thus most likely to achieve the goal (there might of course be hidden mechanisms that prevents X from replicating the goal the other agent (seemed to) achieve, e.g. a hidden fingerprint reader in the buttons).

Upon repeated usage, both in observation mode and in action mode, it can be seen that each modification of the model set \mathcal{M} , using the methods above, makes it more reliable within a given sub-domain $\mathcal{M}_{\mathcal{D}}$. Repeated usage and testing of the models increases the overall reliability of the set as a whole in small steps, as they capture the target phenomena. The system is continuously trying to improve each of its models, hypothetically reaching the maximum precision allowed by the environment and the allotted time and resources.

When this point is reached, every phenomenon is modeled as well as possible.

4.1 The Asymmetry of Abduction & Deduction

These models work like the logical implication (\Rightarrow), i.e. a *perfect* model that says $\alpha \Rightarrow \beta$ guarantees that if α is witnessed, β will be witnessed next. But they are non-axiomatic; in no case are you assured that if we witness β , α is the cause. This can be seen also by the fact that if we want to witness e.g. an event β (which is our goal), trying to achieve α (make α happen/observed) may or may not be the best solution, but will always be a solution [2].

To illustrate, consider the case where we have perfect models of a phenomenon. Given an initial state S_{init} the variables (and their values) of that state will tell us with 100% certainty which models are relevant, and these models will produce subsequent states with complete perfection. In this case we will be certain of the future state, given the initial one. Even if models are not 100% correct, we will have a set of possible futures, from which there will be only one that will be the most correct one. In the inverse direction – abduction – this is not the case: Since in the physical world a subsequent state S_{seq} can come about in many ways, given such a subsequent state and asked to identify the preceding will always require a *choice* as to what is found to be the most likely cause. Saying why the front door is “now open instead of closed” is impossible without more information about particulars of that door and the activities of agents capable of opening doors.

5 Conclusions

Two important conclusions can be drawn from the preceding considerations. First, using abduction and deduction together, on models representing relations between variables in an environment, can help reduce the model set to contain models that approach the actual causal relations between the variables. Thus, a controller with access to such models can use prediction to verify a-priori the likely future unfolding of events, given present state, and predict the effects of its own action and inaction – in other words, to plan. Second, even in partially non-deterministic worlds, having models that approximate causal relations between variables is better than having only statistical information because it explicitly identifies the relevant variables affected by any action (to the extent possible), and thus provides better support for goal achievement.

Acknowledgments

This work was supported by Reykjavik University and the Icelandic Institute for Intelligent Machines.

References

- [1] Conant, R.C., Ashby, W.R.: Every good regulator of a system must be a model of that system. *International Journal of Systems Science* **1**(2), 89–97 (1970)
- [2] Martha Cialdea Mayer, F.P.: Abduction is not deduction-in-reverse. *Logic Journal of the IGPL* **4**, 95–108 (1996)
- [3] Nguyen-Tuong, D., Peters, J.: Model learning for robot control: a survey. *Cognitive Processing* **12**(4), 319–340 (Apr 2011)
- [4] Nivel, E., Thórisson, K.R., Steunebrink, B.R., Dindo, H., Pezzulo, G., Rodríguez, M., Hernández, C., Ogibene, D., Schmidhuber, J., Sanz, R., others: Bounded Seed-AGI. In: *Artificial General Intelligence*, pp. 85–96. Springer (2014)
- [5] Pearl, J.: *do*-calculus revisited. In: de Freitas, N., Murphy, K. (eds.) *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*. pp. 4–11. AUAI Press, Corvallis, OR (2012)
- [6] Pearl, J.: Bayesianism and causality, or, why I am only a half-Bayesian. vol. 12, pp. 19–36 (2001)
- [7] Pearl, J.: *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edn. (2009)
- [8] Pearl, J.: Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution. arXiv:1801.04016 [cs, stat] (Jan 2018), arXiv: 1801.04016
- [9] Thórisson, K.R., Bieger, J., Schiffel, S., Garrett, D.: Towards Flexible Task Environments for Comprehensive Evaluation of Artificial Intelligent Systems & Automatic Learners. In: *Proceedings of AGI-15*. pp. 187–196. Springer-Verlag, Berlin (Jul 2015)
- [10] Thórisson, K.R., Bieger, J., Thorarensen, T., Sigurðardóttir, J.S., Steunebrink, B.R.: Why Artificial Intelligence Needs a Task Theory — And What it Might Look Like. In: *Proceedings of AGI-16*. Springer-Verlag, New York, NY, USA (2016)
- [11] Vaandrager, F.: Model learning. *Commun. ACM* **60**(2), 86–95 (Jan 2017). <https://doi.org/10.1145/2967606>, <http://doi.acm.org/10.1145/2967606>
- [12] Wang, P.: Insufficient Knowledge and Resources-A Biological Constraint and Its Functional Implications. In: *AAAI Fall Symposium: Biologically Inspired Cognitive Architectures*. Citeseer (2009)