Causal Learnability

Loizos Michael
Open University of Cyprus
loizos@ouc.ac.cy

Abstract

The ability to predict, or at least recognize, the state of the world that an action brings about, is a central feature of autonomous agents. We propose, herein, a formal framework within which we investigate whether this ability can be autonomously *learned*.

The framework makes explicit certain premises that we contend are central in such a learning task: *(i)* slow sensors may prevent the sensing of an action's *direct* effects during learning; *(ii)* predictions need to be made reliably in *future* and novel situations.

We initiate in this work a thorough investigation of the conditions under which learning is or is not feasible. Despite the very strong negative learnability results that we obtain, we also identify interesting special cases where learning is feasible and useful.

1 Introduction

An important part of the knowledge humans employ is causal in nature: it captures the manner and conditions under which a world state changes following the execution of some action. Understandably, then, such type of knowledge has been the focus of several formalisms / calculi developed for commonsense reasoning over the years (see, e.g., [McCarthy and Hayes, 1969; Thielscher, 1998; Miller and Shanahan, 2002]). For the potential of these formalisms to be realized, the causal knowledge that they assume is given should be somehow acquired. We investigate, herein, the extent to which this acquisition task can be carried out autonomously through learning.

The learning setting we consider is this: An initial state of affairs is observed, a given action is executed, and *some* state following the execution of the action is observed. Given several pairs of such initial / resulting states, the agent's goal is to approximate how the particular action maps initial states to resulting ones. Our work is based on the following premises:

(P1) The resulting state need not necessarily be the one immediately following the action's execution. Hence, the resulting state may differ from the initial state not only on those properties of the environment that are directly affected by the action, but also on those that are indirectly so by the ramifications of the action's effects.

(P2) The agent's goal is not simply to identify a pattern in the available data. Instead, the goal is to build a hypothesis that the agent can use to make predictions of how the action would affect a state that *it has not observed before*.

We prove a rich suite of results bordering and straddling the fine line that distinguishes learnable from unlearnable cases. Of note is the fact that the proofs of our positive results are constructive, and either provide novel or adapt existing algorithms that can be readily implemented into reasoning agents.

2 A Framework for Causal Change

Our framework for causal change follows typical frameworks of reasoning about actions and change in the literature, both in terms of syntax and semantics (see, e.g., [Kakas *et al.*, 2011]).

We fix a non-empty finite list $\mathcal{F}$ of ***fluents***, representing the properties of the environment that an agent may sense, and which may change over time as a result of action executions.

A ***state over*** $\mathcal{F}$ is a vector $\texttt{st} \in \{\texttt{0},\texttt{1}\}^{|\mathcal{F}|}$, determining a truth-value for each fluent in $\mathcal{F}$. We shall write $\texttt{st}[i]$ to refer to the value of the i-th fluent in $\mathcal{F}$ according to the state $\texttt{st}$.

A ***causal law*** (***of order*** k) ***over*** $\mathcal{F}$ is a statement of the form "S $\texttt{causes}$ L", where S is a set of literals (with $|S| = k$), and L a literal. A causal law is ***monotone*** if it uses no negated fluents. A ***domain*** c is a finite collection of causal laws.

A literal L is ***triggered in*** a domain c ***by*** a state $\texttt{st}$ if there is a causal law "S $\texttt{causes}$ L" $\in c$ such that $\texttt{st}$ satisfies S.

A ***model*** $\mathcal{M}$ ***of*** a domain c is a mapping of each time-point $T \in \mathbb{N}$ to a state $\mathcal{M}(T)$, such that: $\mathcal{M}(T+1)$ satisfies a given literal L if and only if either *(i)* L is triggered in c by $\mathcal{M}(T)$, or *(ii)* $\mathcal{M}(T)$ satisfies L, and $\overline{L}$ is not triggered in c by $\mathcal{M}(T)$.

This semantics captures the minimal set of principles necessary for causal domains, namely that the effects are brought about (Condition *(i)*), and that default inertia applies to fluents whose persistence is not caused to stop (Condition *(ii)*).

Example 1 (Faulty Wiring) *Consider a scenario described in terms of whether current flows (C), the wiring is okay (W), and the fuse is working (F). A domain c includes the causal laws "$\{C, \overline{W}\}$ $\texttt{causes}$ $\overline{F}$" and "$\{\overline{F}\}$ $\texttt{causes}$ $\overline{C}$". An initial state $\mathcal{M}(0)$ that satisfies $\{C, \overline{W}, F\}$ triggers $\overline{F}$, resulting in a new state $\mathcal{M}(1)$ that satisfies $\{C, \overline{W}, \overline{F}\}$. The latter state triggers $\overline{C}$ (and $\overline{F}$), resulting in a new state $\mathcal{M}(2)$ that satisfies $\{\overline{C}, \overline{W}, \overline{F}\}$. By persistence, all subsequent states $\mathcal{M}(T)$, $T > 2$ are equal to $\mathcal{M}(2)$. The mapping $\mathcal{M}$ is a model of c.*

Regarding the manner in which the developed formalization will come into play in the sequel, we point upfront the following. We shall not consider the scenario usually considered by the RAC community of computing models, or some aspect thereof, given access to a specific domain c. Instead, we shall formalize and investigate what could be considered the inverse scenario: given access to models, or some aspects thereof, of an *unknown* target domain, identify the domain.

3 Passive and Active Sensing of States

An agent's limited resources and sensor capabilities prevent it from having access to *all* models of the target domain, or *complete* access to any one of the target domain's models. We will consider, instead, a setting where the agent observes an initial state, then executes some action, and finally observes a resulting state. The cycle is, then, repeated a number of times.

Since the agent is attempting to learn the effects of its actions, we can assume that it cannot choose the initial state, but can only passively sense whatever it might be. We model this by letting the agent's sensors randomly — according to some unknown fixed, but arbitrary, distribution — choose an initial state corresponding to some model of the target domain.

We shall let $\texttt{st} \leftarrow \texttt{sense}(c)$ denote that state $\texttt{st}$ is sensed by the agent in the manner described above, when the target domain is, although unbeknown to the agent, c.

Although the agent is learning the effects of its actions, we may consider the case where it has done so for some of them, and can, thus, use them to partially choose the initial state; at which point, it will execute one of the actions whose effects it is still trying to learn. We model this by assuming that the agent may map the fluents in some subset of $\mathcal{F}$ to any truth-values it wishes. Given this constraint, the truth-values of the rest of the fluents are determined as in the passive case above.

We shall let $\texttt{st} \leftarrow \texttt{sense}(c \mid \texttt{fix})$ denote that state $\texttt{st}$ is sensed by the agent in the manner described above, *given* that the truth-value $\texttt{st}[i]$ of fluent F_i in $\texttt{st}$ matches that in tuple $\texttt{fix} \in \{\texttt{0}, \texttt{1}, *\}^{|\mathcal{F}|}$ whenever, of course, $\texttt{fix}[i] \in \{\texttt{0}, \texttt{1}\}$.

The action executed by the agent shall be assumed to be the same each time, in line with the agent's goal to gather enough information to deduce the action's effects. Our framework of causal change was, in fact, designed with this assumption in mind, and does not mention actions explicitly. This design choice is without loss of generality, since learning the effects of multiple actions can be reduced to learning a distinct target domain for each action (with the causal laws in each target domain being those applicable to the corresponding action).

The state sensed after the action execution need not immediately follow the action execution. We will assume that the agent cannot, in fact, choose how soon to observe this state after the action execution, since the rippling effect caused by the ramifications of the action might be too fast for the agent's sensors. We model this by letting a parameter determine the number of time-steps that need to elapse between the agent sensing the initial state and sensing this second state (which comes, of course, from the same model as the initial state). This last point makes precise and explicit Premise *(P1)*.

We update again the notation we have introduced earlier to capture the requirements mentioned above. We shall, then, let $\langle \texttt{st}_1, \texttt{st}_2 \rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$ denote that some model $\mathcal{M}$ of c is chosen according to some fixed arbitrary distribution, so that: $\texttt{st}_1 = \mathcal{M}(0)$ is the initial state of this model; $\texttt{st}_2 = \mathcal{M}(t)$ is the second state, sensed t time-steps after the action execution. Beyond $\texttt{st}_1, \texttt{st}_2$, no other information is revealed to the agent during any single sensing session.

The aforementioned distribution over models depends, in general, on the choice of $\texttt{fix}$. When $\texttt{fix} = *^{|\mathcal{F}|}$, the agent passively senses its environment, according to some distribution over models. When it attempts to actively fix the truth-values of some fluents, the remaining ones may be indirectly affected; hence, the distribution over models may be affected depending on $\texttt{fix}$. In all cases, the distribution remains the same for all sensings that involve the same choice of $\texttt{fix}$.

4 Learning with Predictive Guarantees

Given access to observed pairs $\langle \texttt{st}_1, \texttt{st}_2 \rangle$ of states obtained through $\texttt{sense}(c, t \mid \texttt{fix})$, the agent's goal is to approximate the target domain c, whatever it might be. This approximation we expect to be accompanied by certain guarantees. We discuss next what these guarantees might reasonably be.

The approximation need not be a collection of causal laws as defined in Section 2. Indeed, our causal framework aims to give semantics to the dynamics of an agent's environment, and not to prescribe how the agent will represent them. We shall, thus, denote by $\mathcal{H}$ the set of all representations that the agent may choose to use. He shall call each representation h in $\mathcal{H}$ a ***hypothesis***, and $\mathcal{H}$ itself the ***hypothesis class***.

Conceivably, the agent may have a prior bias on the nature of the target domain c. For instance, even though the agent does not know c directly, it may believe that (or wish to investigate the case where) c comprises only of monotone causal laws, or of causal laws of order at most 127. To capture this bias, we shall assume that the target domain c belongs in some ***domain class*** $\mathcal{C}$, which itself is known to the agent. In case no bias exists, we can simply let $\mathcal{C}$ contain all possible domains.

The setting so far is that assuming the target domain c belongs in $\mathcal{C}$, and given access to $\langle \texttt{st}_1, \texttt{st}_2 \rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$, the agent is expected to return a hypothesis $h \in \mathcal{H}$. We designate the period during which these happen the ***training*** phase, and we expect that it takes time polynomial in the relevant parameters. What other guarantees should we impose?

One type of guarantees, often found in the literature (which we shall discuss later on), is that h should be consistent with all pairs $\langle \texttt{st}_1, \texttt{st}_2 \rangle$ that the agent has sensed *during training*. That is, for each sensed pair $\langle \texttt{st}_1, \texttt{st}_2 \rangle$, there should be a model $\mathcal{M}$ of h such that $\texttt{st}_1 = \mathcal{M}(0)$ and $\texttt{st}_2 = \mathcal{M}(t)$.

This requirement by itself is, however, not very useful. The agent attempts to learn the effects of its actions so that it can utilize this knowledge in *future* situations. We consider, thus, a scenario where following the training phase, an agent enters a ***testing*** phase, where it uses the hypothesis h. Depending on the precise use of h, we get different types of guarantees. The testing phase makes precise and explicit Premise *(P2)*.

The formalizations that follow next build on the *Probably Approximately Correct* semantics [Valiant, 1984], by adopting the requirements that learning during the training phase is carried out efficiently, and that the learned hypothesis is accompanied by predictive guarantees during the testing phase.

Departing from the original PAC model, however, our formalizations do not assume that learning has access to individual states of the environment, nor that the values of the fluents within each single state are somehow correlated. Instead, they assume that learning has access to pairs of states, and that the values of the fluents in the second state are correlated with the values of the fluents in the first state. Such a type of correlation captures the causal structure of the environment that we seek to identify, as opposed to the static structure of the environment that the original PAC model seeks to identify.

4.1 Recognizing the Effects of Actions

Perhaps the simplest testing requirement is: Given an initial state $\texttt{st}_1$, *recognize* whether a second given state $\texttt{st}_2$ could have been caused by a given action. More precisely, the agent is given a pair of states $\langle \texttt{st}_1, \texttt{st}_2 \rangle$ such that $\texttt{st}_1 = \mathcal{M}(0)$, where $\mathcal{M}$ is a model of the target domain c chosen according to an arbitrary distribution that remains, however, fixed during the training and testing phases. Furthermore, one of the following two scenarios occurs, each with probability $1/2$: *either* $\texttt{st}_2 = \mathcal{M}(t)$ *or* $\texttt{st}_2$ is chosen arbitrarily among all states in $\{\texttt{0}, \texttt{1}\}^{|\mathcal{F}|}$. The agent should decide which is the case. We call the above testing requirement the ***recognition*** task.

We do not expect the agent will *always* succeed in deciding correctly. On the other hand, we do expect that its success rate will noticeably exceed the trivially obtainable rate of $1/2$.

In ***weak recognition***, we expect the agent's success rate to be at least $1/2 + 1/p(|\mathcal{F}|, |c|)$, for *some* arbitrary fixed polynomial $p(\cdot, \cdot)$. Thus, the success rate need only be slightly better than random guessing. We shall later see that in certain settings, achieving even this weak guarantee is infeasible.

In ***strong recognition***, we expect the agent's success rate to be at least $1 - \varepsilon$, for *any given* $\varepsilon \in (0, 1]$. Thus, the success rate need be arbitrarily close to perfect. The agent may employ resources that grow polynomially in $1/\varepsilon$ to compensate for this more arduous requirement. Thus, the better predictions we expect, the more time we allow the agent to train.

Definition 1 (Learning to Recognize the Effects of Actions) *A domain class $\mathcal{C}$ is* ***learnable by recognition by*** *a hypothesis class $\mathcal{H}$ if there exists an algorithm $\mathcal{L}$ for which it holds that:*
for every sensor $\texttt{sense}(c, t \mid \cdot)$ *such that $c \in \mathcal{C}$ and $t \in \mathbb{N}$, (and for every $\varepsilon \in (0, 1]$ in the case of strong recognition),*
during the training phase
$\mathcal{L}$ is given access to $\mathcal{C}$, $\mathcal{H}$, (and ε in the case of strong recognition), and pairs of states $\langle \texttt{st}_1, \texttt{st}_2 \rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$, *where* $\texttt{fix} \in \{\texttt{0}, \texttt{1}, *\}^{|\mathcal{F}|}$ *may be chosen by $\mathcal{L}$ at each call, and has the property that*
$\mathcal{L}$ runs in time polynomial in $|\mathcal{F}|$, $|c|$, (and $1/\varepsilon$ in the case of strong recognition), and with probability[1] *$2/3$ returns a hypothesis $h \in \mathcal{H}$ that succeeds in the recognition task.*

We shall later consider restrictions of this model, by imposing, for instance, constraints on $\mathcal{C}$, t, and $\texttt{fix}$, so that we can investigate causal learnability in more structured settings.

4.2 Generating the Effects of Actions

A second testing requirement is: Given an initial state $\texttt{st}_1$, *generate* a state $\texttt{st}_2$ that could have been caused by a given action. More precisely, the agent is given a state $\texttt{st}_1$ such that $\texttt{st}_1 = \mathcal{M}(0)$, where $\mathcal{M}$ is a model of the target domain c chosen according to an arbitrary distribution that remains, however, fixed during the training and testing phases. The agent should generate a state $\texttt{st}_2$ such that $\texttt{st}_2 = \mathcal{M}(t)$, with the value of t being that used during the training phase. We call the above testing requirement the ***generation*** task.

As in the recognition task, we do not expect the agent will *always* succeed in this task. Unlike the recognition task, there are now exponentially-many possible outcomes, and the trivially obtainable rate of $(1/2)^{|\mathcal{F}|}$ is, effectively, zero. Any noticeable improvement above zero would, thus, be welcome.

In ***weak generation***, we expect the agent's success rate to be at least $1/p(|\mathcal{F}|, |c|)$, for *some* arbitrary fixed polynomial $p(\cdot, \cdot)$. Thus, as in the weak recognition setting, the success rate need only be slightly better than random guessing.

In ***strong generation***, we expect the agent's success rate to be at least $1 - \varepsilon$, for *any given* $\varepsilon \in (0, 1]$. This setting follows the strong recognition setting on how ε comes into play during learning. We shall later see that in certain settings, achieving this rather strong guarantee is, in fact, feasible.

Definition 2 (Learning to Generate the Effects of Actions) *A domain class $\mathcal{C}$ is* ***learnable by generation by*** *a hypothesis class $\mathcal{H}$ if the provisions of Definition 1 hold, after substituting "generation" for each occurrence of "recognition".*

We shall later consider restrictions of this model, by imposing, for instance, constraints on $\mathcal{C}$, t, and $\texttt{fix}$, so that we can investigate causal learnability in more structured settings.

5 Learnability when Sensors are Fast

We consider first the case where sensors are fast enough to sense the state *immediately* following the execution of a given action, which incorporates the action's direct effects — even though the state may subsequently change further due to the action's indirect effects. Fast sensing *does not* imply any restrictions on the richness or complexity of the target domain.

Even Weak Learning of General Domains is Hard

We start this section with two strong negative results, assuming that DNF formulas (of size polynomial in the number of their variables) are hard to learn under the PAC semantics.

More precisely, the "DNF Learning Hardness" assumption we invoke is this: given access to randomly chosen inputs to a circuit implementing a hidden DNF formula, and to the corresponding outputs of the circuit, it is intractable to construct a good approximation of the DNF formula in the PAC sense. This constitutes one of the major open questions in Computational Learning Theory [Kearns and Vazirani, 1994], even if at most one of the following holds: *(i)* the inputs are drawn from the uniform distribution, or *(ii)* the inputs can be chosen (in addition to being drawn from some arbitrary distribution).

Theorem 1 (Non-Learnability of Direct Action Effects 1) *Given that the "DNF Learning Hardness" assumption holds, there exists a domain class $\mathcal{C}$ not weakly learnable by recognition by any hypothesis class $\mathcal{H}$, even if* $\texttt{sense}(c, t \mid \texttt{fix})$ *is such that the target domain $c \in \mathcal{C}$ contains only monotone causal laws, $t = 1$, the models of c are sensed uniformly at random, and* $\texttt{fix}$ *may be chosen from* $\{\texttt{0}, \texttt{1}, *\}^{|\mathcal{F}|-1} \times \{*\}$.

Sketch of Proof: For each DNF formula φ include in $\mathcal{C}$ the domain c_φ s.t. for each term $v_{i_1} \cdots v_{i_k} \overline{v_{i_{k+1}}} \cdots \overline{v_{i_m}}$ in φ, c_φ includes "$\{F^+_{i_1}, \ldots, F^+_{i_k}, F^-_{i_{k+1}}, \ldots, F^-_{i_m}\}$ $\texttt{causes}$ F_0".

[1] This probability corresponds to the confidence parameter δ in the PAC semantics. For simplicity, and without loss of generality, we have fixed it to a constant value, and shall not discuss it further. It is included so that the PAC requirements are properly captured.

If $\mathcal{C}$ is learnable, then so are DNF formulas, by calling the learning algorithm for $\mathcal{C}$ and providing it with pairs of states as follows: When $\texttt{fix} = *^{|\mathcal{F}|}$, construct states $\texttt{st}_1, \texttt{st}_2$ that match the inputs and output of the DNF formula. Otherwise, choose a state $\texttt{st}_1$ that agrees with $\texttt{fix}$ on those fluents that have their values fixed, and complete the rest $\texttt{st}_1$ uniformly at random, except set F_0 to $\texttt{1}$. Set $\texttt{st}_2 = \texttt{st}_1$. □

This constitutes a very strong negative result, as it: *(i)* applies to the weakest type of learning (weak recognition by any hypothesis class), *(ii)* uses the strongest type of sensing (fast sensing with essentially unrestricted fixing — the single fluent that cannot be fixed is the one affected by the action whose effects are being learned), and *(iii)* concerns a fairly restricted class of domains (with only monotone causal laws). Furthermore, *(iv)* only uniform sensing needs to be dealt with.

Despite the availability of fixing, the result goes through by exploiting the single value in $\texttt{fix}$ that cannot be set, to essentially render the fixing of the remaining values not useful. It is natural, hence, to ask whether allowing $\texttt{fix}$ to be chosen in a completely unrestricted manner would restore learnability.

Theorem 2 (Non-Learnability of Direct Action Effects 2) *Given that the "DNF Learning Hardness" assumption holds, there exists a domain class $\mathcal{C}$ not weakly learnable by recognition by any hypothesis class $\mathcal{H}$, even if* $\texttt{sense}(c, t \mid \texttt{fix})$ *is such that $t = 1$, and* $\texttt{fix}$ *may be chosen from $\{\texttt{0}, \texttt{1}, *\}^{|\mathcal{F}|}$.*

Sketch of Proof: For each DNF formula φ include in $\mathcal{C}$ the domain c_φ s.t. for each term $v_{i_1} \cdots v_{i_k} \overline{v_{i_{k+1}}} \cdots \overline{v_{i_m}}$ in φ, c_φ includes "$\{F_{i_1}, \ldots, F_{i_k}, \overline{F_{i_{k+1}}}, \ldots, \overline{F_{i_m}}\}$ $\texttt{causes}$ F_0".

If $\mathcal{C}$ is learnable, then so are DNF formulas, by calling the learning algorithm for $\mathcal{C}$ and providing it with pairs of states as follows: When $\texttt{fix} = *^{|\mathcal{F}|}$, construct states $\texttt{st}_1, \texttt{st}_2$ that match the inputs and output of the DNF formula. Otherwise, choose a state $\texttt{st}_1$ that agrees with $\texttt{fix}$ on those fluents that have their values fixed, and complete the rest $\texttt{st}_1$ uniformly at random. Choose a DNF input consistent with $\texttt{st}_1$, and using a membership query obtain the output of the target DNF formula. Set $\texttt{st}_2 = \texttt{st}_1$, under the provision that if F_0 is $\texttt{0}$ in $\texttt{st}_1$, then F_0 is set to the obtained output in $\texttt{st}_2$. □

The second negative result is also strong: we have given up causal law monotonicity, and uniform sensing; but we have allowed the unconditional fixing of attributes when sensing.

Special Domain Classes that are Strongly Learnable

From a theoretical perspective, the two negative results seem to indicate that there is little hope in deriving positive results of any significance. From a somewhat philosophical perspective, they seem to clash with the apparent ability of humans to identify causal relations in our surrounding environment.

Fortunately, this pessimism and the apparent paradox can be dismissed once we observe that the domain classes whose unlearnability Theorems 1 and 2 establish, comprise domains of *few* causal laws of *large* order. This *imbalance* in the domains is an artifact of the reduction from DNF formulas. We postulate that it does not occur in real-world domains; at least we are not aware of any such cases. Our aim, then, is to exclude consideration of such presumably unnatural domains.

A domain with causal laws of maximum order k is ***balanced*** if the number of its causal laws is a non-negligible[2] in $|\mathcal{F}|$ fraction of the number of all causal laws of order at most k. Domains containing *any* number of causal laws of order independent of $|\mathcal{F}|$, are easily shown to be balanced. In general, the definition insists that the size of the domain grows, not too slowly, with the order of the causal laws it contains.

Under the assumption of fast sensors, we are able to establish the learnability of balanced domains in a strong sense, as it: *(i)* applies to the strongest type of learning (strong generation with the hypothesis class being equal to the domain class), *(ii)* uses the weakest type of sensing among those that are fast (no fluent can be fixed), and *(iii)* concerns a general class of domains (with causal laws of arbitrary monotonicity).

Theorem 3 (Passive Learning of Balanced Domains) *Consider a domain class $\mathcal{C}$ comprising only domains that are balanced. Assume the hypothesis class $\mathcal{H}$ is equal to $\mathcal{C}$, and that* $\texttt{sense}(c, t \mid \texttt{fix})$ *is called always with $t = 1$, and* $\texttt{fix} = *^{|\mathcal{F}|}$*. Then, $\mathcal{C}$ is learnable by strong generation by $\mathcal{H}$.*

Sketch of Proof: Draw a number of pairs of states to populate a set $\mathcal{T}$ of training instances. Set $\ell = 0$, and repeat the following: Initialize a hypothesis h with all possible causal laws of order at most ℓ. For every pair of states $\langle \texttt{st}_1, \texttt{st}_2 \rangle \in \mathcal{T}$, remove from h all causal laws "S $\texttt{causes}$ L" such that $\texttt{st}_1$ satisfies S, and $\texttt{st}_2$ satisfies $\overline{L}$; such causal laws provably cannot be part of the target domain c. Now, test h on $\mathcal{T}$. If h correctly predicts all pairs of states in $\mathcal{T}$, then return h, and terminate. Else, increase ℓ by one, and repeat the process.

The algorithm terminates before ℓ exceeds the maximum order k of causal laws in the target domain c. By the balanced property, the running time is polynomially-related to $|c|$. □

A second aspect of the unlearnable domain classes in Theorems 1 and 2 is that multiple causal laws have the same fluent in their head, yet their preconditions are completely uncorrelated with each other. Such dissociated causal laws are useful when one wishes to model effectively unrelated circumstances that, nonetheless, cause the same effect to be brought about. There exist, however, scenarios where the causal laws are naturally much more tightly coupled. Such causal laws could result, for instance, when encoding exception lists.

An ***exception list*** e is a list of pairs $\langle C_j, Q_j \rangle$ of some set C_j of literals, and a literal $Q_j \in \{F, \overline{F}\}$ for some fixed fluent F, such that: given a truth-assignment, the conclusion of e is the highest-indexed Q_j for which $C_1 \cup C_2 \cup \ldots \cup C_j$ is satisfied. Thus, each pair refines the conditions under which a conclusion is to be drawn, and overrides the preceding conclusions.

Example 2 (Flying Bird) *Consider a scenario where a bird named Tweety starts or stops flying when shot at by a clumsy hunter (who often misses). This could be represented thus:*

if	*Tweety is not a penguin*	*then*	*start flying*
unless	*Tweety is hit by the shot*	*then*	*stop flying*
unless	*Tweety's injury is minor*	*then*	*start flying*

Note that exception lists (nested if-then-unless statements) are closely related to decision lists (nested if-then-else statements), a class of functions known to be learnable under the PAC semantics [Rivest, 1987]. It can be shown, in fact, that

[2] Borrowing this useful notion from Cryptography, a function is *non-negligible* in n if it is $\Omega(1/n^k)$ for any k that is constant in n.

LEARNEXCEPTIONLIST (Training set $\mathcal{T}$, Fluent F)

```
1:  Set T≠ to include ⟨st1, st2⟩ ∈ T s.t. st2[i] ≠ st1[i].
2:  Set T0 to include ⟨st1, st2⟩ ∈ T s.t. st2[i] = 0.
3:  Set T1 to include ⟨st1, st2⟩ ∈ T s.t. st2[i] = 1.
4:  Set h to be the empty exception list.
5:  Set all = cur = FindMaxConjunction(T≠, ∅).
6:  Repeat the following, until termination:
7:      Set neg = FindMaxConjunction(T0, all) \ all.
8:      Set pos = FindMaxConjunction(T1, all) \ all.
9:      If (neg ≠ ∅), then set Q = F and nxt = neg.
10:     If (pos ≠ ∅), then set Q = ¬F and nxt = pos.
11:     If (neg ∪ pos = ∅), then return h, and terminate.
12:     Set h = h ∘ ⟨cur, Q⟩, cur = nxt, all = all ∪ cur.
```

Algorithm 1. An algorithm that passively learns exception lists. Conjunctions of literals are represented as sets.

the latter class coincides with a *strict* subclass of exceptions lists: those whose first condition is the empty conjunction.

In the general case, and unlike decision lists, exception lists may abstain from making predictions in certain settings. This feature, although not applicable in the original PAC model, is particularly appropriate in a causal setting, as it accounts for cases where values persist without being caused to change.

Observation 1 (Modelling Exception Lists via Causal Laws) *For any exception list $\langle C_1, Q_1\rangle, \langle C_2, Q_2\rangle, \ldots, \langle C_m, Q_m\rangle$, there exists a domain c of size polynomial in the size of the exception list, such that $\texttt{st}_2$ satisfies the conclusion of the exception list when it is applied on the truth-assignment $\texttt{st}_1$, whenever $\langle \texttt{st}_1, \texttt{st}_2\rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$ with $t = 1$.*

The domain is this: For every $j \in \{1, 2, \ldots, m\}$, and every literal $L \in C_{j+1}$ (applicable if $j < m$), domain c includes the causal law "$C_1 \cup C_2 \cup \ldots \cup C_j \cup \{\overline{L}\}$ $\texttt{causes}$ Q_j".

We show that (even non-balanced) domains modelling exception lists are learnable, in the strong sense of Theorem 3.

Theorem 4 (Passive Learning of Exception List Domains) *Consider a domain class $\mathcal{C}$ comprising only domains where each of the fluents is caused according to some exception list. Assume the hypothesis class $\mathcal{H}$ is equal to $\mathcal{C}$, and that $\texttt{sense}(c, t \mid \texttt{fix})$ is called always with $t = 1$, and $\texttt{fix} = *^{|\mathcal{F}|}$. Then, $\mathcal{C}$ is learnable by strong generation by $\mathcal{H}$.*

Sketch of Proof: Draw a number of pairs of states to populate a set $\mathcal{T}$ of training instances. For each fluent F_i, invoke Algorithm 1 with inputs $\mathcal{T}$ and F_i, to obtain a hypothesis h_i. Following Observation 1, convert h_i into a set h'_i of causal laws. Set $h = \bigcup_{F_i} h'_i$, return the hypothesis h, and terminate.

Subroutine $FindMaxConjunction(\mathcal{T}', S')$ is based on an algorithm in [Valiant, 1984]. Each iteration introduces at least one new literal in all, which guarantees termination.

Assume that $h_i = \langle C_1, Q_1\rangle, \langle C_2, Q_2\rangle, \ldots, \langle C_m, Q_m\rangle$. By a careful case analysis it can be shown that h_i cannot err:

(i) if C_1 is not satisfied, since that would contradict Step 5.

(ii) if $C_1 \cup C_2 \cup \ldots \cup C_j$, $j < m$ is maximally satisfied, since Step 12 would imply that C_{j+1} should also be satisfied.

(iii) if $C_1 \cup C_2 \cup \ldots \cup C_m$ is satisfied, since the target exception list would imply that h_i should have been longer. □

Of course, even if only some of the fluents are caused according to an exception list, those can be learned as above, and other approaches can be used for the remaining fluents.

Learning through Active Selection of the Initial State

The established negative results point to two more scenarios where learning may be possible: use sensing with unrestricted fixing *and* focus on either domains of monotone causal laws, or uniform sensing. We prove next that both results hold.

Theorem 5 (Active Learning of Monotone Domains) *Consider a domain class $\mathcal{C}$ comprising only domains of monotone causal laws. Assume the hypothesis class $\mathcal{H}$ is equal to $\mathcal{C}$, and that $\texttt{sense}(c, t \mid \texttt{fix})$ is called always with $t = 1$, and $\texttt{fix}$ may be chosen from $\{0, 1, *\}^{|\mathcal{F}|}$. Then, $\mathcal{C}$ is learnable by strong generation by $\mathcal{H}$.*

Sketch of Proof: By adapting the algorithm for PAC learning monotone DNF formulas in [Angluin, 1988]. Start by setting $h = \emptyset$, and test h on a training set $\mathcal{T}$. Whenever h errs on $\texttt{st}_1$, use fixing to find a minimal subset S of the fluents satisfied in $\texttt{st}_1$ that preserve the value of fluent F_i in $\texttt{st}_2$. Add "S $\texttt{causes}$ F_i" in h, and repeat. □

It is straightforward to extend Theorem 5 to a certain class of non-monotone domains, where it holds that all causal laws with the same fluent F_i in their head (possibly negated), comprise simultaneously either all positive or all negative literals.

Assuming that the models of the target domain are sensed uniformly at random, and doing without the requirement for hypotheses to be represented as sets of causal laws, it is possible to learn general non-monotone non-balanced domains.

Theorem 6 (Active Learning under Uniform Sensing) *Consider a domain class $\mathcal{C}$. Assume that $\texttt{sense}(c, t \mid \texttt{fix})$ is called always with $t = 1$, the models of the target domain $c \in \mathcal{C}$ are sensed uniformly at random, and $\texttt{fix}$ may be chosen from $\{0, 1, *\}^{|\mathcal{F}|}$. Then, there exists a hypothesis class $\mathcal{H}$ such that $\mathcal{C}$ is learnable by strong generation by $\mathcal{H}$.*

Sketch of Proof: By multiple reductions to the problem of learning DNF formulas, which can be achieved by adapting the celebrated result in [Jackson, 1997]. The DNF formulas encode whether fluent values *change* as a result of the action execution. Hence, for each causal law "S $\texttt{causes}$ L" that affects a fluent F_i, the corresponding DNF formula will include a term $S \cup \{\overline{L}\}$. Each $\langle \texttt{st}_1, \texttt{st}_2\rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$ will give rise to a positive learning example for the DNF formula if $\texttt{st}_1[i] \neq \texttt{st}_2[i]$, and a negative example otherwise. □

6 Learnability when Sensors are Slow

In many cases, the agent's sensors are slow, and result in the agent sensing some state following the action's execution that may incorporate some or all of the action's *indirect* effects. In this setting, the action's execution and the sensed state are interjected by one or more states, which encode the intermediate materialization of the action's effects. Fluents may have their values change back and forth, without being sensed so.

It is, thus, unsurprising that slow sensors hinder learnability further. In fact, hidden states can be exploited to simulate the workings of circuits, certain classes of which are unlearnable under reasonable assumptions, *even if causal laws are restricted to have order* 2 [Michael, 2007]. Although many relevant negative results could be presented, we focus on what is, admittedly, a more interesting direction: establishing that under certain natural assumptions, learnability is reinstated.

We achieve this not by restricting the structure of domains, but by imposing constraints on the sensors through which their models are sensed. The constraints aim to alleviate the hiding of complex computations that take place in the intermediate states, and which ultimately leads to unlearnability.

Simple Sensing as a Counterbalance to Slow Sensing

A sensor $\texttt{sense}(c, t \mid \texttt{fix})$ is (α, β)-***simple*** if for every $F_i \in \mathcal{F}$ there exists an α-term β-DNF formula such that for every $\langle \texttt{st}_1, \texttt{st}_2 \rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$, the formula evaluates on $\texttt{st}_1$ to true exactly when $\texttt{st}_1[i] \neq \texttt{st}_2[i]$ holds.

Although the agent is not necessarily aware of α and β during learning, the learning time is bounded by those parameters. This, in turn, allows us to carry positive results from the case of fast sensing, under certain assumptions on α and β.

Theorem 7 (Passive Learning of Simply Sensed Domains) *Consider a domain class $\mathcal{C}$. Assume the hypothesis class $\mathcal{H}$ is equal to $\mathcal{C}$, and that $\texttt{sense}(c, t \mid \texttt{fix})$ is (α, β)-simple for β independent of $|\mathcal{F}|$, is called always with the same t, and $\texttt{fix} = *^{|\mathcal{F}|}$. Then, $\mathcal{C}$ is learnable by strong generation by $\mathcal{H}$.*

Sketch of Proof: The algorithm is similar to that in the proof of Theorem 3. The value of k is β. The causal laws in the hypothesis h initially are those with the negation of their head in their conditions. This ensures that during the k-th iteration, the conditions of the casual laws in h are always a superset of the terms of the respective DNF formulas whose existence is guaranteed by the definition of (α, β)-simple. □

Theorem 7 has two immediate corollaries. The first corollary relates to domains whose causal laws include at most one literal in their conditions. These domains are not overly simple, since a chain of such causal laws effectively allows fluents to be caused according to some disjunction of literals. It is immediate that for any such domain c, any given sensor $\texttt{sense}(c, t \mid \texttt{fix})$ is $(O(|\mathcal{F}|), 1)$-simple. Hence:

Corollary 8 *A domain class comprising only domains whose causal laws include at most one literal in their conditions, is learnable in the sense of Theorem 7.*

The second corollary relates to a situation that could, presumably, be common when a teacher (or parent) guides a student (or child) to learn the effects of an action. In such a setting, then, one could envision the teacher as choosing the initial state $\texttt{st}_1$ so that if some effect is caused after the execution of the action, the effect would have also been caused had the initial state been heavily perturbed. Put in other words, the teacher chooses the initial state so that only a small part of it is essential, which, arguably, makes learning easier.

Formally, a sensor $\texttt{sense}(c, t \mid \texttt{fix})$ is γ-***friendly*** if for every $F_i \in \mathcal{F}$ and every $\langle \texttt{st}_1, \texttt{st}_2 \rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$ such that $\texttt{st}_1[i] \neq \texttt{st}_2[i]$, there exists a γ fraction of the fluents in $\mathcal{F}$ so that any state $\texttt{st}'_1$ resulting from $\texttt{st}_1$ by changing the truth-values of those fluents is such that $\texttt{st}'_1[i] \neq \texttt{st}'_2[i]$ whenever $\langle \texttt{st}'_1, \texttt{st}'_2 \rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$.

It is immediate that any sensor $\texttt{sense}(c, t \mid \texttt{fix})$ that is γ-friendly is necessarily $(\alpha, (1-\gamma)|\mathcal{F}|)$-simple. Hence:

Corollary 9 *A domain class is learnable in the sense of Theorem 7 whenever the sensor is $(1 - O(|\mathcal{F}|^{-1}))$-friendly.*

Friendly sensing is related to the general problem of learning in the presence of irrelevant information [Blum and Langley, 1997], progress in which could lead to new learnability results in our framework with certain less friendly sensors.

If we, now, allow active learning in addition to simple sensing, we can establish a generalization of Theorems 5 and 6.

Theorem 10 (Active Learning of Simply Sensed Domains) *Consider a domain class $\mathcal{C}$. Assume that $\texttt{sense}(c, t \mid \texttt{fix})$ is (α, β)-simple, and is called always with the same t. Then, Theorems 5 and 6 apply mutatis mutandis for the learnability of $\mathcal{C}$, assuming that extra time polynomial in α is allowed.*

There are natural conditions under which α is polynomial in $|\mathcal{F}|$, and the extra time in Theorem 10 can be eliminated.

The ***associated graph of*** a domain c comprises a node for each fluent, and a directed edge between two fluents when the former and latter appear, respectively, in the body and head of a causal law in c. It is possible to show that the number of directed paths in the associated graph of c of length at most t, upper-bounds the value of α in any (α, β)-simple sensor $\texttt{sense}(c, t \mid \texttt{fix})$. The next corollary follows from above:

Corollary 11 *A domain class is learnable in the sense of Theorem 10 whenever the associated graphs of the domains contain polynomially in $|\mathcal{F}|$ many directed paths of length at most the sensor's speed t. This holds, in particular, under the very natural assumption that t is independent of $|\mathcal{F}|$.*

Transparency for Learning Width-Bounded Domains

Another way to stop intermediate states from hiding complex computations, is to consider *transparent* sensing, where any change can be attributed to something that the agent observes.

Formally, a sensor $\texttt{sense}(c, t \mid \texttt{fix})$ is ***transparent*** if for every $F_i \in \mathcal{F}$ and every $\langle \texttt{st}_1, \texttt{st}_2 \rangle \leftarrow \texttt{sense}(c, t \mid \texttt{fix})$ such that $\texttt{st}_1[i] \neq \texttt{st}_2[i]$, assuming $L \in \{F_i, \overline{F_i}\}$ is satisfied by $\texttt{st}_2$ (i.e., L indicates $\texttt{st}_2[i]$), there exists a causal law "S $\texttt{causes}$ L" $\in c$ whose body is satisfied by $\texttt{st}_1$ *or* $\texttt{st}_2$.

An analogous notion of transparency has been employed in the context of PAC learning CP-networks [Dimopoulos *et al.*, 2009]. At its basis, the learning algorithm developed in that context receives preferences over pairs of outcomes that are transparently entailed by an acyclic CP-network of bounded in-degree, and efficiently identifies one such CP-network. We suggest that some adaptation of that learning algorithm could be used to establish the following result in our framework.

Conjecture 1 (Passive Learning with Transparent Sensing) *Consider a domain class $\mathcal{C}$ comprising only domains with an acyclic associated graph of in-degree at most k. Assume the hypothesis class $\mathcal{H}$ is equal to $\mathcal{C}$, and that $\texttt{sense}(c, t \mid \texttt{fix})$ is transparent, is called always with the same t that is upper bounded by some polynomial in $|\mathcal{F}|$, and $\texttt{fix} = *^{|\mathcal{F}|}$. Then, $\mathcal{C}$ is learnable by strong recognition by $\mathcal{H}$, assuming that extra time polynomial in $|\mathcal{F}|^k$ is allowed.*

We believe that causal learnability and preference elicitation (e.g., in PAC learning CP-networks) share some key features that can be formally characterized, and that further research may reveal deep connections between the two areas.

7 Conclusions

We have presented a formal theory and a comprehensive collection of results in causal learning, delineating the scenarios that are learnable from those that are not. Related work often diverges from ours on not adopting our two premises.

Otero [2005], for instance, considers indirect effects of actions that all come about in the state immediately following the action execution. Hence, there are no hidden states, nor a rippling effect as indirect effects materialize over time; properties which are a necessary implication of Premise *(P1)*.

Some works seek to identify causal laws consistent with a training set. Inoue *et al.* [2005], for instance, show how to do this for Language $\mathcal{A}$, by reducing domains to Deterministic Finite Automata. The known unlearnability of the latter class when guarantees are expected [Kearns and Vazirani, 1994], exemplifies the non-adoption of Premise *(P2)* by that work.

More recently, Amir and Chang [2008] consider an online setting of partially observing a sequence of states and interjected actions, and investigate the problem of simultaneously identifying the current state of the world, and the direct effects of all involved actions. Indirect effects, predictive guarantees, and the use of queries are not considered in that work.

We find the partial state observability and the online setting intriguing for future investigation, alongside the investigation of sensors that occasionally return incorrect inputs. Relevant work in Learning Theory can aid in this line of research (e.g., [Aldous and Vazirani, 1995; Kearns, 1998; Michael, 2010b]).

In an effort to strengthen our positive results, albeit at the expense of learning autonomy, one could investigate learnability with the aid of teachers [Goldman and Mathias, 1996], or more powerful sensors [Angluin, 1988]. We have touched upon these issues when discussing how the use of active sensing can be best interpreted, but further research is warranted.

Additional extensions can be considered to account for scenarios where, for instance: the speed of sensing varies during the training and testing phases; additional actions occur while the effects of certain preceding actions are still being realized; the causal models being learned are non-deterministic; etc.

In terms of applications, beyond offering a computational toolbox for the autonomous acquisition of causal knowledge, the developed causal learnability framework can also be used for the formal study of questions such as whether learning by generation is strictly harder than learning by recognition, or even for the identification of the context under which certain stories are to be interpreted and understood [Michael, 2010a].

References

[Aldous and Vazirani, 1995] David Aldous and Umesh Vazirani. A Markovian Extension of Valiant's Learning Model. *Information and Computation*, 117(2):181–186, 1995.

[Amir and Chang, 2008] Eyal Amir and Allen Chang. Learning Partially Observable Deterministic Action Models. *Journal of Artificial Intelligence Research*, 33:349–402, 2008.

[Angluin, 1988] Dana Angluin. Queries and Concept Learning. *Machine Learning*, 2(4):319–342, 1988.

[Blum and Langley, 1997] Avrim Blum and Pat Langley. Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence*, 97(1–2):245–271, 1997.

[Dimopoulos *et al.*, 2009] Yannis Dimopoulos, Loizos Michael, and Fani Athienitou. Ceteris Paribus Preference Elicitation with Predictive Guarantees. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1890–1895, 2009.

[Goldman and Mathias, 1996] Sally Goldman and David Mathias. Teaching a Smarter Learner. *Journal of Computer and System Sciences*, 52(2):255–267, 1996.

[Inoue *et al.*, 2005] Katsumi Inoue, Hideyuki Bando, and Hidetomo Nabeshima. Inducing Causal Laws by Regular Inference. In *Proceedings of the 15th International Conference on Inductive Logic Programming (ILP'05)*, pages 154–171, 2005.

[Jackson, 1997] Jeffrey Jackson. An Efficient Membership-Query Algorithm for Learning DNF with Respect to the Uniform Distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997.

[Kakas *et al.*, 2011] Antonis Kakas, Loizos Michael, and Rob Miller. Modular-E and the Role of Elaboration Tolerance in Solving the Qualification Problem. *Artificial Intelligence*, 175(1):49–78, 2011.

[Kearns and Vazirani, 1994] Michael Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, Cambridge, MA, U.S.A., 1994.

[Kearns, 1998] Michael Kearns. Efficient Noise-Tolerant Learning from Statistical Queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[McCarthy and Hayes, 1969] John McCarthy and Patrick Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence*, 4:463–502, 1969.

[Michael, 2007] Loizos Michael. On the Learnability of Causal Domains: Inferring Temporal Reality from Appearances. In *Working notes of the 8th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'07)*, 2007.

[Michael, 2010a] Loizos Michael. Computability of Narrative. In *Working notes of the 2nd Symposium on Computational Models of Narrative (Narrative'10)*, 2010.

[Michael, 2010b] Loizos Michael. Partial Observability and Learnability. *Artificial Intelligence*, 174(11):639–669, 2010.

[Miller and Shanahan, 2002] Rob Miller and Murray Shanahan. Some Alternative Formulations of the Event Calculus. *Lecture Notes in Artificial Intelligence*, 2408:452–490, 2002.

[Otero, 2005] Ramón Otero. Induction of the Indirect Effects of Actions by Monotonic Methods. In *Proceedings of the 15th International Conference on Inductive Logic Programming (ILP'05)*, pages 279–294, 2005.

[Rivest, 1987] Ronald Rivest. Learning Decision Lists. *Machine Learning*, 2(3):229–246, 1987.

[Thielscher, 1998] Michael Thielscher. Introduction to the Fluent Calculus. *Electronic Transactions on Artificial Intelligence*, 2(3–4):179–192, 1998.

[Valiant, 1984] Leslie Valiant. A Theory of the Learnable. *Communications of the ACM*, 27:1134–1142, 1984.