On the Learnability of Causal Domains: Inferring Temporal Reality from Appearances*

Loizos Michael
Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138, U.S.A.
loizos@eecs.harvard.edu

Abstract

We examine the feasibility of learning causal domains by observing transitions between states as a result of taking certain actions. We take the approach that the observed transitions are only a macro-level manifestation of the underlying micro-level dynamics of the environment, which an agent does not directly observe. In this setting, we ask that domains learned through macro-level state transitions are accompanied by formal guarantees on their predictive power on future instances. We show that even if the underlying dynamics of the environment are significantly restricted, and even if the learnability requirements are severely relaxed, it is still intractable for an agent to learn a model of its environment. Our negative results are universal in that they apply independently of the syntax and semantics of the framework the agent utilizes as its modelling tool. We close with a discussion of what a complete theory for domain learning should take into account, and how existing work can be utilized to this effect.

Introduction

Mathematical logic has established itself as a means of formalizing commonsense reasoning about actions and change. Numerous frameworks (McCarthy & Hayes 1969; Harel 1984; Gelfond & Lifschitz 1992; Thielscher 1998; Doherty *et al.* 1998; Miller & Shanahan 2002; Giunchiglia *et al.* 2004; Kakas, Michael, & Miller 2005) have been proposed for modelling the various intricacies of our environment, addressing to various extents the fundamental problems inherent in such an endeavor. For these frameworks to be widely accepted as useful tools in the design of autonomous agents that employ common sense when deliberating about their actions, one needs to go beyond programming knowledge into agents, and towards endowing agents with the capability of learning this knowledge through interactions with their environment. The reasoning mechanisms developed by the Commonsense Reasoning community over the years can then be employed to put the acquired knowledge into good use, allowing agents to draw sound conclusions about their environment, and the effects of their actions.

In this work we take a first step in examining the feasibility of undertaking such a learning task. Two main premises underlie our study. First, that the goal of learning should not be to identify domains that are simply consistent with learning examples the agent has observed, but rather domains that can *provably* make highly accurate predictions in *future* situations that the agent will face. Second, that the time granularity at which the state of the environment evolves is *finer* than that at which the agent takes actions and makes observations. What the agent perceives as consecutive states in its environment is not necessarily so in the underlying dynamics that cause the state transitions. Thus, while at the observable time granularity the push of a button causes the light to be on immediately afterwards, the environment in fact transitions through multiple unseen intermediate states, during which the electric current comes on, the wire in the light bulb heats up, and so on. The macro-level manifestation of the micro-level dynamics of the agent's environment resembles a temporal analog of McCarthy's "Appearance and Reality" dichotomy (2006); what appears to be the case does not necessarily fully match or explain the underlying reality.

We model the macro/micro granularity discrepancy via a simple framework of causal change. We assume the environment is described by a set of causal laws, which get triggered (as a result of an agent's actions) in the current state of the environment, and subsequently get resolved, possibly triggering new causal laws. The environment transitions through a set of micro-states until it eventually stabilizes to a final state, which the agent gets to observe. Our approach is a stripped-down version of recent work (Kakas, Michael, & Miller 2005) that has shown that such a treatment enables one to naturally model a variety of domains in a modular and elaboration tolerant manner, providing a clean solution to the Ramification and Qualification Problems. Our emphasis here, however, is on the learnability of domains, and not on their reasoning semantics; a minimal framework of causal change suffices for our purposes.

The learning problem is formalized as that of inferring a model of the environment by observing transitions between macro-states. As per our premises, *(i)* the agent does not observe the micro-states that interject between the initial and final macro-states, and *(ii)* the agent is expected to be confident that its inferred model is highly accurate in predicting macro-state transitions in future and previously unseen situations. The learning setting that the agent is faced with is made precise through an extension of the Probably Approximately Correct learning semantics (Valiant 1984). A number

*This work was supported by grant NSF-CCF-04-27129.

of possible extensions are considered to account for varying degrees of stringency on the learning requirements and the amount of information that is available to the agent.

We examine the feasibility of learning in the described setting, and establish rather severe limitations on the learnability of domains (under certain cryptographic assumptions). Our negative results hold even under a number of simplifying assumptions on the complexity of the underlying dynamic model of the environment, even when the learning requirements are significantly relaxed, and even if the agent is allowed to experiment and actively choose its learning examples. More surprisingly, and perhaps more importantly, our results hold *independently* of the means the agent employs to build its model, and do not hinge on the syntax or semantics of any particular framework. The framework of causal change we present is only utilized to describe the environment from which learning examples are drawn, and need not be employed by the agent when learning. In fact, in view of our negative learnability results, our simple framework of causal change only serves to further strengthen the established intractability of learning causal domains.

We close with a discussion of the implications of our results. We review some related work and explain how existing positive results in domain learning should be interpreted, in view of the severe limitations on learnability that we establish. We discuss the potential of deriving domain learning algorithms that do not sacrifice predictive guarantees, and what such a task would necessitate. We also consider relaxations of assumptions made in this work, and briefly mention how existing work in Computational Learning Theory can offer the tools necessary to develop a complete treatment of domain learnability, along the lines presented in this work.

A Simple Framework of Causal Change

We live in an arguably complex environment, and one can never hope to fully describe all the intricacies that surround us. Even under the simplifying assumption that the environment can be described as a collection of discrete attributes, which assume discrete values, and which change in discrete time steps, a lot remains to be modelled: transitions in attribute values might occur non-deterministically, values might oscillate across time, triggered processes might get resolved in an arbitrary order, and actions exogenous to any subsystem of interest might affect it (see (Kakas, Michael, & Miller 2005) for a treatment of such issues). These facts notwithstanding, it is often useful to consider an environment that is well-behaved with respect to these issues, in an effort to understand and appreciate the complexity of "simpler environments". In this section we develop a framework for modelling such a simple environment, borrowing some ideas from (Kakas, Michael, & Miller 2005).

Definition 1 (States and Satisfaction) *Consider any non-empty finite set $\mathcal{F}$ of **fluent constants**. A **state over** $\mathcal{F}$ is a vector $\mathtt{st} \in \{\mathtt{0},\mathtt{1}\}^{|\mathcal{F}|}$. A set of fluent literals S is **satisfied in** a state $\mathtt{st}$ if $\mathtt{st}[i] = \mathtt{0}$ for every negative literal $\overline{F_i} \in S$, and $\mathtt{st}[i] = \mathtt{1}$ for every positive literal $F_i \in S$. A **state transition over** $\mathcal{F}$ is simply a pair $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle$ of states over $\mathcal{F}$; we call $\mathtt{st}_1$ the **initial**, and $\mathtt{st}_2$ the **final** state.*

A state transition occurs when the initial state "evolves" to the final state. We assume that this transition process is explained by virtue of a set of causal laws.

Definition 2 (Domains of Causal Laws) *A **causal law (of order** k**)** is a statement of the form "S causes L", where S is a set (of cardinality at most k) of fluent literals, and L is a fluent literal; the causal law is **monotone** if all fluent literals in $S \cup \{L\}$ are positive. A **domain** c is a (finite) collection of causal laws.*

The intended meaning of a causal law "S causes L" is that whenever the preconditions S are satisfied in a state, the effect L holds in a subsequent state. Thus, the transition from an initial to a final state results from causal laws that get triggered and resolved in a series of intermediate states.

Definition 3 (Successor and Stable States) *A state $\mathtt{st}_2$ is the **successor of** a state $\mathtt{st}_1$ **w.r.t.** a domain c if $E(\mathtt{st}_1; c) \triangleq \{L \mid$ "S causes L" $\in c$; S is satisfied in $\mathtt{st}_1\}$ is such that: (i) $E(\mathtt{st}_1; c)$ is satisfied in $\mathtt{st}_2$, and (ii) $\mathtt{st}_2[i] = \mathtt{st}_1[i]$ for every fluent constant $F_i \in \mathcal{F}$ such that $F_i, \overline{F_i} \notin E(\mathtt{st}_1; c)$. A state $\mathtt{st}_2$ is **reachable (in** m **steps) from** a state $\mathtt{st}_1$ **w.r.t.** a domain c if $\mathtt{st}_2$ is the successor w.r.t. c of either $\mathtt{st}_1$ or a state reachable (in $m-1$ steps) from $\mathtt{st}_1$ w.r.t. c. A state $\mathtt{st}$ is **stable w.r.t.** a domain c if $\mathtt{st}$ is the successor of itself w.r.t. c. A domain c is **consistent with** $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle$ if $\mathtt{st}_2$ is the (unique) stable successor of $\mathtt{st}_1$ w.r.t. c.*

Our semantics captures the minimal, perhaps, set of principles necessary for domains with causal laws, namely that the effects of causal laws are instantiated (condition *(i)*), and that default inertia applies to properties that are not affected by causal laws (condition *(ii)*).

Learning from Observing State Transitions

An agent wishing to learn the dynamic properties of its environment presumably does so by observing the current state, taking certain actions, and then observing the resulting final state. For ease of exposition we only consider the case of learning from state transitions with complete information, and assume that each initial state is associated with exactly one final state, as per the semantics of the previous section.

We formalize the learning setting as follows. An agent observes state transitions $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle$ over some fixed set of fluent constants. The initial state $\mathtt{st}_1$ is thought of as being drawn from an underlying fixed probability distribution $\mathcal{D}$. The probability distribution is arbitrary and unknown to the agent, and aims to capture the complex interdependencies of fluents in the agent's environment, as well as the lack of control over the current state of affairs in which the agent is executing its actions. The final state $\mathtt{st}_2$ results when a set of causal laws (triggered by the agent's actions) apply on the initial state $\mathtt{st}_1$. In order to facilitate the learning process, one needs to make a minimal assumption on the set of causal laws that apply on the initial state, namely that they are fixed across observations. In the same spirit, we also assume that whatever actions the agent is taking to trigger a state transition also remain fixed across observations.

More precisely, we assume that there exists a domain $c \in \mathcal{C}$ that is consistent with all observed state transitions.

The actual domain c is not made known to the agent; still, the agent has access to the class $\mathcal{C}$ of possible domains and the set of fluent constants $\mathcal{F}$ over which the domains in $\mathcal{C}$ are defined. The domain class $\mathcal{C}$ should be thought of as a prior bias that the agent has on the structure of its environment. Depending on the circumstances, one might restrict $\mathcal{C}$ to certain subsets of all syntactically valid domains, increasing thus the prior bias and making the learning task easier.

Given the domain class $\mathcal{C}$, and access to randomly drawn state transitions consistent with some domain $c \in \mathcal{C}$, an agent enters a *training* phase, during which it uses the available state transitions to construct a hypothesis $h \in \mathcal{H}$ about its environment. Note that allowing the agent exponential time in the relevant problem parameters essentially trivializes learning, since the agent can practically observe all possible state transitions. To avoid this situation, we require that training be carried out *efficiently*, in time that is only polynomial in the relevant problem parameters. Following the training phase, an agent enters a *testing* phase, where the agent is faced with possibly *previously unseen* state transitions, drawn however from the *same* underlying probability distribution and consistent with the *same* domain c. Learning is said to be successful if *with high probability* h is *sufficiently often* consistent with these new state transitions. The testing phase aims to exemplify two key points. First, that the agent is tested under the same conditions that it faced during the training phase; this corresponds to the fact that the agent was trained and tested in the same environment. Second, that the returned hypotheses are expected to be accompanied by *predictive guarantees*; the agent needs to be able to make predictions in new situations and be confident in the accuracy of these predictions. To see why this requirement is not overly optimistic, observe that an accurate hypothesis need only be returned with high probability. This acknowledges the fact that the agent might be unlucky during the training phase, and not be able to obtain a good sample of state transitions. Furthermore, even if a good sample was obtained, we only require that the returned hypothesis be approximately correct, acknowledging the fact that during testing an agent may be faced with rare state transitions that need not be accurately predicted.

It is important to note at this point that the syntax and semantics of the returned hypotheses are not a priori restricted in any manner. In particular, we do not expect an agent to return a domain in the syntax and under the semantics of the framework presented in the preceding section. Our framework of causal change only serves as a model of the environment from which state transitions are drawn. The agent attempting to learn the structure of its environment is free to model it in any manner it sees fit. Thus, for example, an agent might choose to return a domain description in the syntax and under the semantics of either Situation Calculus (McCarthy & Hayes 1969) or Language $\mathcal{ME}$ (Kakas, Michael, & Miller 2005), presumably even employing additional constructs these frameworks might offer to model other aspects of the environment beyond causal laws. Even more generally, a hypothesis might be any efficiently evaluatable function that given an input produces a corresponding output. If one wishes to explicitly restrict the class of possible hypotheses, one can do so by defining $\mathcal{H}$.

The learning setting we employ is an extension of the *Probably Approximately Correct* learning model (Valiant 1984), and a number of possible variations appropriate for domain learning are formalized in the rest of this section.

Passive Learning through Observations

We start with a rather strong definition of learnability.

Definition 4 (State Transition Exact Oracle) *Given a probability distribution $\mathcal{D}$ over states, and a domain c, the* ***exact oracle*** *$\mathcal{E}(\mathcal{D}; c)$ is a procedure that runs in unit time, and on each call $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle \leftarrow \mathcal{E}(\mathcal{D}; c)$ returns a state transition $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle$, where $\mathtt{st}_1$ is drawn randomly and independently from $\mathcal{D}$, and c is consistent with $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle$.*

Definition 5 (Learnability by Generation) *Given a set of fluent constants $\mathcal{F}$, a class $\mathcal{C}$ of domains is* ***learnable from transitions by*** *a class $\mathcal{H}$ of* ***generative hypotheses*** *if there exists an algorithm $\mathcal{L}$ such that for every probability distribution $\mathcal{D}$ over states, every domain $c \in \mathcal{C}$, every real number $\delta : 0 < \delta \leq 1$, and every real number $\varepsilon : 0 < \varepsilon \leq 1$, algorithm $\mathcal{L}$ has the following property: given access to $\mathcal{E}(\mathcal{D}; c)$, δ, and ϵ, algorithm $\mathcal{L}$ runs in time polynomial in $1/\delta$, $1/\varepsilon$, $|\mathcal{F}|$, and the size of c, and with probability $1 - \delta$ returns a hypothesis $h \in \mathcal{H}$ such that*

$$Pr\left(h(\mathtt{st}_1) = \mathtt{st}_2 \mid \langle \mathtt{st}_1, \mathtt{st}_2 \rangle \leftarrow \mathcal{E}(\mathcal{D}; c)\right) \geq 1 - \varepsilon.$$

Note that Definition 5 asks that returned hypotheses are *generative* in the sense that given an initial state they are expected to generate a final state that is accurate with respect to an agent's observations. Weaker notions of learnability are of course possible.

Definition 6 (State Transition Noisy Oracle) *Given a probability distribution $\mathcal{D}$ over states, and a domain c, the* ***noisy oracle*** *$\mathcal{N}(\mathcal{D}; c)$ is a procedure that runs in unit time, and on each call $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle \leftarrow \mathcal{N}(\mathcal{D}; c)$ returns a state transition $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle$, where $\mathtt{st}_1$ is drawn randomly and independently from $\mathcal{D}$, and c is consistent with $\langle \mathtt{st}_1, \mathtt{st}_2 \rangle$ with probability $1/2$.*

Definition 7 (Learnability by Recognition) *Given a set of fluent constants $\mathcal{F}$, a class $\mathcal{C}$ of domains is* ***learnable from transitions by*** *a class $\mathcal{H}$ of* ***recognitive hypotheses*** *if the same provisions hold as in Definition 5, except that $h \in \mathcal{H}$ is such that*

$$Pr\left(h(\langle \mathtt{st}_1, \mathtt{st}_2 \rangle) = c(\langle \mathtt{st}_1, \mathtt{st}_2 \rangle) \mid \langle \mathtt{st}_1, \mathtt{st}_2 \rangle \leftarrow \mathcal{N}(\mathcal{D}; c)\right) \geq 1 - \varepsilon.$$

Intuitively, we have shifted the requirement for hypotheses from that of accurately generating (or predicting) the final state given only an initial state, to that of recognizing whether a given state transition is consistent with the hidden target domain c. This latter requirement is presumably weaker, since an algorithm is only expected to decide on the validity of a state transition, rather than to predict a final state among the exponentially many possible candidates. Indeed, whereas a recognitive hypothesis can trivially achieve accuracy $1/2$ (by simply classifying all state transitions as valid), this is not possible for generative hypotheses.

Note that an agent still receives only valid state transitions during the training phase, since invalid transitions do not naturally occur in an agent's environment. Besides, invalid state transitions can be trivially simulated by replacing the final state in a valid state transition by an arbitrary state (in the same way that the noisy oracle does so). The use of the exact oracle is in fact strictly more beneficial in that the valid state transitions can be identified by the agent.

Learning to Weakly Recognize Consistency

Both of our domain learnability definitions so far require that a target domain can be approximated to arbitrarily high accuracy $1-\varepsilon$ and with arbitrarily high confidence $1-\delta$ (albeit with an appropriate increase in the allowed resources) in order for a class of domains to be characterized as learnable. The PAC learning literature has considered a notion of learnability that relaxes these requirements to the maximum extent possible (Kearns & Valiant 1994). The corresponding definition of domain learnability is as follows.

Definition 8 (Weak Learnability by Recognition) *Given a set of fluent constants $\mathcal{F}$, a class $\mathcal{C}$ of domains is **weakly learnable from transitions by** a class $\mathcal{H}$ of **recognitive hypotheses** if there exists an algorithm $\mathcal{L}$ and polynomials $p(\cdot,\cdot)$ and $q(\cdot,\cdot)$ such that for every probability distribution $\mathcal{D}$ over states, and every domain $c \in \mathcal{C}$, algorithm $\mathcal{L}$ has the following property: given access to $\mathcal{E}(\mathcal{D};c)$, algorithm $\mathcal{L}$ runs in time polynomial in $|\mathcal{F}|$, and the size of c, and with probability $1/p(|\mathcal{F}|, size(c))$ returns a hypothesis $h \in \mathcal{H}$ such that*

$$Pr\left(h(\langle \mathtt{st}_1, \mathtt{st}_2\rangle) = c(\langle \mathtt{st}_1, \mathtt{st}_2\rangle) \mid \langle \mathtt{st}_1, \mathtt{st}_2\rangle \leftarrow \mathcal{N}(\mathcal{D};c)\right) \geq 1/2 + q(|\mathcal{F}|, size(c)).$$

Thus, not only have we relaxed the requirement for arbitrarily high confidence and accuracy, but we have also allowed them to diminish as the size of the problem increases. In particular, we now only require that accuracy is slightly better than the trivially obtainable accuracy of $1/2$.

Active Learning through Experimentation

The final relaxation of the learning requirements we consider is on the information that is made available to an agent during the training phase. We have, thus far, assumed that an agent obtains learning examples by simply taking actions in the current state of the environment, and then observing the resulting final state. Conceivably, an agent might first take other actions (whose effects might already be known or previously learned by the agent) to bring the environment to a chosen state, and then attempt to learn from state transitions with this particular state as their initial state. This setting allows the agent to have some control over the types of learning examples it utilizes during the training phase. In the learning literature this type of learning is called learning with queries (see, e.g., (Angluin 1988)), since the agent can be thought of as asking questions and receiving answers.

Definition 9 (State Transition Query Oracle) *Given a domain c, the **query oracle** $\mathcal{Q}(\cdot;c)$ is a procedure that runs in unit time, and on each call $\langle \mathtt{st}_1, \mathtt{st}_2\rangle \leftarrow \mathcal{Q}(\mathtt{st}_1;c)$ returns a state transition $\langle \mathtt{st}_1, \mathtt{st}_2\rangle$, where $\mathtt{st}_1$ is given as input to the oracle, and c is consistent with $\langle \mathtt{st}_1, \mathtt{st}_2\rangle$.*

It is natural to assume that although the agent might wish to bring the environment to a chosen state, its lack of knowledge or physical abilities do not allow the agent to choose the entire state of the environment. Indeed, an agent that can bring its environment to a chosen state presumably already knows the causal model of its environment, leaving nothing to be learned. We consider, therefore, the situation where the agent is able to set a significant known portion of the state to certain chosen values, albeit in doing so it loses any guarantees it might have on the status of the rest of the state. This situation can be modelled through restricted queries.

Definition 10 (State Transition Restricted Query Oracle) *Given a domain c, the **restricted query oracle** $\mathcal{R}(\cdot;c)$ is a procedure that runs in unit time, and on each call $\langle \mathtt{st}_1, \mathtt{st}_2\rangle \leftarrow \mathcal{R}(\mathtt{st}_0;c)$ returns a state transition $\langle \mathtt{st}_1, \mathtt{st}_2\rangle$, where $\mathtt{st}_0$ is given as input to the oracle, $\mathtt{st}_1$ is some state that agrees with $\mathtt{st}_0$ on an inverse polynomial size fixed subset of $\mathcal{F}$, and c is consistent with $\langle \mathtt{st}_1, \mathtt{st}_2\rangle$.*

Definition 11 (Learnability with (Restricted) Queries) *Given a set of fluent constants $\mathcal{F}$, a class $\mathcal{C}$ of domains is **weakly learnable from transitions with queries (resp., restricted queries) by** a class $\mathcal{H}$ of **recognitive hypotheses** if the same provisions hold as in Definition 8, except that algorithm $\mathcal{L}$ is also given access to $\mathcal{Q}(\cdot;c)$ (resp., $\mathcal{R}(\cdot;c)$).*

With the use of query oracles an agent is no longer *passively* observing state transitions, but can *actively experiment* with its environment to obtain information on specific situations that could be too rare to observe passively. This setting brings up the possibility of requiring that domains are learned exactly (Angluin 1988), rather than simply approximately. We will not, however, examine this alternative and more stringent learning model in this work.

Analogously to the extension of Definition 8, one can extend Definitions 5 and 7 to employ (restricted) query oracles.

Negative Results in Domain Learning

The learnability of various classes of problems has been extensively studied under the PAC semantics, and many positive and negative results have been established, often under certain complexity or cryptographic assumptions. Perhaps the best-studied classes are those of boolean functions over boolean inputs, usually viewed as digital circuits over the standard logic gates.

Circuit learning is very close to domain learning as stated in Definition 5, requiring the same type of learnability guarantees. Roughly speaking, an algorithm observes randomly chosen inputs to a hidden target circuit, and for each input the corresponding boolean output. The algorithm is then expected to produce a hypothesis that on randomly chosen inputs predicts with high accuracy the corresponding output of the hidden circuit. Weak learning is defined in circuit learning in a similar fashion as domain learning. Queries can also be employed to obtain the value of the hidden circuit on an input chosen by the algorithm; such queries are called *membership* queries, since they essentially ask if a given input is a member of the inputs on which the circuit evaluates to true. We call the resulting setting *weak PAC learning with*

membership queries. We reduce the problem of circuit weak PAC learning with membership queries to that of domain weak learning from state transitions with restricted queries.

Theorem 1 (Reduction of Circuit to Domain Learning) *Consider the class $\mathcal{C}_0$ of polynomial size circuits with n inputs, fan-in at most k, and depth at most m. Consider the class $\mathcal{C}_1$ of all domains over some set $\mathcal{F}$ of fluent constants with polynomial cardinality in n, such that all domains in $\mathcal{C}_1$: (i) are of size polynomial in $|\mathcal{F}|$, (ii) only contain causal laws of order k out of which only one is not monotone, and (iii) only explain transitions between states reachable in $m+1$ steps. If $\mathcal{C}_1$ is weakly learnable from transitions with restricted queries by any class of (polynomially evaluatable) recognitive hypotheses, then $\mathcal{C}_0$ is weakly PAC learnable with membership queries.*

Proof: We first establish certain correspondences between the two learning problems. Let $W = \{w_1, w_2, \ldots, w_t\}$ denote the set of wires over which the circuits in $\mathcal{C}_0$ are defined, and let w_t correspond to the output wire. Construct the set of fluent constants $\mathcal{F} = \{F_i^-, F_i^+ \mid w_i \in W\} \cup \{F_0\}$.

• For each circuit $\texttt{ckt} \in \mathcal{C}_0$ construct a domain $\texttt{dom}(\texttt{ckt}) \in \mathcal{C}_1$ over $\mathcal{F}$ such that: *(i)* for each output w_{i_0} of an AND-gate over $\{w_{i_1}, \ldots, w_{i_k}\}$ in $\texttt{ckt}$, $\texttt{dom}(\texttt{ckt})$ includes the causal laws "$\{F_{i_1}^+, \ldots, F_{i_k}^+\}$ causes $F_{i_0}^+$", and "$\{F_{i_j}^-\}$ causes $F_{i_0}^-$" for each $j \in \{1, \ldots, k\}$; *(ii)* similarly for every OR-gate and NOT-gate in $\texttt{ckt}$; *(iii)* $\texttt{dom}(\texttt{ckt})$ includes the causal law "$\{\overline{F_t^-}, F_t^+\}$ causes F_0"; and *(iv)* $\texttt{dom}(\texttt{ckt})$ includes the causal laws "$\{F_t^+\}$ causes F" and "$\{F_t^-\}$ causes F", for each fluent constant $F \in \mathcal{F} \setminus \{F_0\}$.

• For each circuit input $\texttt{in} \in \{\texttt{0},\texttt{1}\}^n$ construct a state $\texttt{st}(\texttt{in}) \in \{\texttt{0},\texttt{1}\}^{|\mathcal{F}|}$ such that: *(i)* $\texttt{st}(\texttt{in})$ satisfies $\{F_i^-, \overline{F_i^+}\}$ for each circuit input wire w_i set to $\texttt{0}$ under $\texttt{in}$; *(ii)* $\texttt{st}(\texttt{in})$ satisfies $\{\overline{F_i^-}, F_i^+\}$ for each circuit input wire w_i set to $\texttt{1}$ under $\texttt{in}$; and *(iii)* $\texttt{st}(\texttt{in})$ satisfies $\{\overline{F}\}$ for each fluent constant $F \in \mathcal{F} \setminus \{F_i^-, F_i^+ \mid w_i \text{ is a circuit input wire}\}$.

• For each circuit output $\texttt{out} \in \{\texttt{0},\texttt{1}\}$ construct a state $\texttt{st}(\texttt{out}) \in \{\texttt{0},\texttt{1}\}^{|\mathcal{F}|}$ such that: *(i)* $\texttt{st}(\texttt{out})$ satisfies $\{F_0\}$ if and only if the circuit output wire w_t is set to $\texttt{1}$ under $\texttt{out}$; and *(ii)* $\texttt{st}(\texttt{out})$ satisfies $\mathcal{F} \setminus \{F_0\}$.

• For each query state $\texttt{st}_0 \in \{\texttt{0},\texttt{1}\}^{|\mathcal{F}|}$ construct a state $\texttt{alt}(\texttt{st}_0) \in \{\texttt{0},\texttt{1}\}^{|\mathcal{F}|}$ such that: *(i)* $\texttt{alt}(\texttt{st}_0)$ satisfies $\{F_i^-, \overline{F_i^+}\}$ for each circuit input wire w_i such that $\{\overline{F_i^+}\}$ is satisfied by $\texttt{st}_0$; *(ii)* $\texttt{alt}(\texttt{st}_0)$ satisfies $\{\overline{F_i^-}, F_i^+\}$ for each circuit input wire w_i such that $\{F_i^+\}$ is satisfied by $\texttt{st}_0$; and *(iii)* $\texttt{alt}(\texttt{st}_0)$ satisfies $\{\overline{F}\}$ for every fluent constant $F \in \mathcal{F} \setminus \{F_i^-, F_i^+ \mid w_i \text{ is a circuit input wire}\}$. Clearly, $\texttt{alt}(\texttt{st}_0)$ agrees with $\texttt{st}_0$ on a fixed polynomial size subset of $\mathcal{F}$, and equals $\texttt{st}(\texttt{in})$ for a unique circuit input $\texttt{in}$.

All constructions are polynomial-time computable in n, and $\texttt{ckt}$ on input $\texttt{in}$ computes output $\texttt{out}$ if and only if domain $\texttt{dom}(\texttt{ckt})$ is consistent with $\langle \texttt{st}(\texttt{in}), \texttt{st}(\texttt{out}) \rangle$.

Algorithm $\mathcal{L}_0$ for learning $\mathcal{C}_0$ executes algorithm $\mathcal{L}_1$ for learning $\mathcal{C}_1$: Whenever algorithm $\mathcal{L}_1$ requests an example from the exact oracle, algorithm $\mathcal{L}_0$ draws a random circuit input $\texttt{in}$ with the corresponding output $\texttt{out}$, and returns $\langle \texttt{st}(\texttt{in}), \texttt{st}(\texttt{out}) \rangle$ to algorithm $\mathcal{L}_1$. Whenever algorithm $\mathcal{L}_1$ requests an example from the restricted query oracle with input state $\texttt{st}_0$, algorithm $\mathcal{L}_0$ asks a membership query on the unique circuit input $\texttt{in}$ that corresponds to $\texttt{alt}(\texttt{st}_0)$ to obtain the corresponding output $\texttt{out}$, and returns $\langle \texttt{st}(\texttt{in}), \texttt{st}(\texttt{out}) \rangle$ to algorithm $\mathcal{L}_1$. When algorithm $\mathcal{L}_1$ returns a hypothesis h satisfying the conditions of Definition 11, algorithm $\mathcal{L}_0$ employs this hypothesis to make accurate predictions on input $\texttt{in}$ of the hidden target circuit by selecting uniformly at random an output $\texttt{out} \in \{\texttt{0},\texttt{1}\}$, and replying with $\texttt{out}$ if and only if h is consistent with $\langle \texttt{st}(\texttt{in}), \texttt{st}(\texttt{out}) \rangle$. This concludes the proof. □

Theorem 1 establishes a precise connection between properties of the circuit class one considers and properties of the domain class to which one reduces. Since the known negative results of PAC learning circuit classes hold not only on the general class of all polynomial size circuits, but also on certain special subclasses, Theorem 1 allows us to carry these negative results to special subclasses of domains.

Corollary 2 (Transitions from Simple Causal Laws) *Consider the class $\mathcal{C}$ of all domains that: (i) are of size polynomial in the number of available fluent constants $\mathcal{F}$, (ii) only contain causal laws of order 2 out of which only one is not monotone, and (iii) only explain transitions between states reachable in $O(\lg |\mathcal{F}|)$ steps. Then, $\mathcal{C}$ is not weakly learnable from transitions with restricted queries by any recognitive hypothesis class, given that the Factoring Assumption is true.*

Proof: Kharitonov (Kharitonov 1993, Theorem 6) shows that the class $\texttt{NC}^1$ of polynomial size circuits with n inputs, fan-in at most 2, and depth at most $O(\lg n)$, is not weakly PAC learnable with membership queries if the Factoring Assumption holds. The claim now follows from Theorem 1, by observing that the theorem guarantees that $|\mathcal{F}|$ is polynomial in n and therefore that $O(\lg n) + 1 = O(\lg |\mathcal{F}|)$. □

The *Factoring Assumption* states that factoring Blum integers is hard; that is, given a natural number N of the form $p \cdot q$, where both p and q are primes congruent to 3 modulo 4, it is intractable to recover the factors of N. The Factoring Assumption is one of the most widely accepted and used cryptographic assumptions. In fact, a proof that the assumption is false would completely undermine the presumed theoretical security of the well-known RSA cryptosystem (Rivest, Shamir, & Adleman 1978). It is believed, therefore, that for all practical purposes the assumption is true.

Corollary 3 (Transitions with Few Intermediate States) *Consider the class $\mathcal{C}$ of all domains that: (i) are of size polynomial in the number of available fluent constants $\mathcal{F}$, (ii) contain causal laws out of which only one is not monotone, and (iii) only explain transitions between states reachable in $O(1)$ steps. Then, $\mathcal{C}$ is not weakly learnable from transitions with restricted queries by any recognitive hypothesis class, given that the "Strong Factoring Assumption" is true.*

Proof: Kharitonov (Kharitonov 1993, Theorem 9) shows that the class $\texttt{AC}^0$ of polynomial size circuits with n inputs,

and depth at most $O(1)$, is not weakly PAC learnable with membership queries if factoring Blum integers of length ℓ is $(2^{-\ell^{\varepsilon}})$-secure for some $\varepsilon > 0$; we call this condition the "Strong Factoring Assumption". The claim now follows from Theorem 1, by observing that $O(1) + 1 = O(1)$. $\square$

Corollary 3 relies on what we call the *"Strong Factoring Assumption"*. Roughly speaking, this stronger version of the Factoring Assumption states that there exists $\varepsilon > 0$ such that factoring an ℓ-bit integer remains intractable even if we allow an adversary running time $2^{\ell^{\varepsilon}}$ (Kharitonov 1993), as opposed to some polynomial in ℓ. Although less likely to be true, this stronger assumption on the intractability of factoring is still a plausible one.

We have thus established that irrespective of how an agent represents its hypotheses, it is impossible (under the stated assumptions) to learn certain classes of domains. The negative results hold even for classes of domains with practically only monotone causal laws that either have at most two preconditions, or do not form "long chains" in state transitions; that is, the result holds even if the number of intermediate micro-states in observed state transitions is small. These results leave little room for considering simpler domains where learning might not be impaired, without sacrificing the expressivity of domains, and without making unrealistic assumptions on the learning model (e.g., the use of unrestricted query oracles).

Discussion and Conclusions

The induction of domains from observations has recently received an increased interest, especially within the Inductive Logic Programming community. To the extent such frameworks relate to ours, the following remarks can be made as regards to the two premises of this work.

First, causal knowledge is often represented through ramification statements whose preconditions and effects apply on the same state (see, e.g., (Otero 2005)), essentially collapsing all micro-states that follow an action occurrence into the single final macro-state that the agent observes. This arguably less realistic model of causal change provides an agent with much more information than our framework, by explicitly encoding all changes in fluent truth-values in the observed state transitions. Although the use of "flat" ramification statements excludes the possibility of providing natural representations for many real-world domains, one might wish to ask whether learnability is enhanced if one restricts one's attention to the subset of domains that are representable in this manner. In general, the answer to this question might depend on the exact semantics associated with the ramification statements, and it is outside the scope of this work to provide a comprehensive study of this problem.

Second, learnability is taken to correspond to the efficient identification of a domain consistent with training examples, with no guarantees accompanying the predictive power of learned domains on future situations. One can, in fact, identify this as an explanation of the apparent discrepancy between our strongly negative results, and the positive results presented in other frameworks. Especially representative is the case of learning Language $\mathcal{A}$ through a reduction to the problem of learning Deterministic Finite Automata (Inoue, Bando, & Nabeshima 2005); the latter problem is known not to be PAC learnable (Kearns & Vazirani 1994). Despite the fact that the intractability of learning Language $\mathcal{A}$ does not follow from this reduction (although it can be easily shown to follow from our results), it nonetheless illustrates the lack of concern for the predictive power of learned domains.

How should one interpret the current status of our knowledge on domain learnability? Are we trapped in a situation where we either dismiss learnability as infeasible, or give up any formal guarantees on its usefulness? In order to answer these questions one needs to understand that our results only establish intractability in a worst-case scenario. In practice, an agent's environment might not be adversarial, although the extent to which this happens can be determined only through experimentation. Nonetheless, we believe, it is important that theoretical models of such more benign environments be developed, and guarantees of the effectiveness of learning be provided under these environments. The Computational Learning Theory community has examined learnability under various prisms, including restricted probability distributions, the use of teachers during the training phase (see, e.g., (Goldman & Mathias 1996)), and the use of more powerful oracles (see, e.g., (Angluin 1988)). Clearly such assumptions weaken our hope to design fully autonomous agents that develop their own dynamic models of their environment, but perhaps this is not such a big drawback given that some basic knowledge can be feasibly programmed.

In this more optimistic frame of mind, one might wish to question some of the simplifying assumptions we made in this work, albeit doing so will only result in even harder learning problems. In a realistic scenario, not only the intermediate micro-states are not observed, but even the macro-states themselves are only partially observable. McCarthy's "Appearance and Reality" dichotomy arises once more, this time in the static setting of a single state. Can learning be meaningfully defined and carried out in such situations? Can an agent provably learn to make accurate predictions on attributes of its environment that are not always visible even during the training phase? Such questions were studied in recent work (Michael 2007), where a framework that formalizes learning in situations with *arbitrary* missing information was proposed, modelling thus the fact that what is observable is often beyond an agent's control. In that framework various natural classes of concepts were shown to be learnable. An extension to the case of learning domains can be carried out in a manner similar to the extension of the PAC framework in the present work. Orthogonally, one can employ the techniques developed in (Michael 2007) to predict missing information in the observed macro-states before attempting to employ the (now more complete) macro-states for learning the dynamic behavior of the environment.

The related assumption of accurate sensing can also be relaxed, to account for an agent's noisy sensors, or for exogenous factors that affect state transitions. Again, a wealth of results in the Computational Learning Theory literature can be employed to study this problem. As one might expect, noise makes learning harder, and in the case of adversarial noise learning is practically impaired (Kearns & Li 1993).

However, in certain situations of random noise, as the ones we expect an agent to be faced with, learning is still possible, and without a large additional overhead (Kearns 1998).

Finally, we may reconsider our assumption that state transitions are drawn independently from each other. Given the dynamic nature of an agent's environment, observed states might more appropriately be thought of as being drawn according to a Markovian process (Aldous & Vazirani 1995), that itself transitions from state to state as observations are drawn. Another possibility is to employ the Mistake Bounded Model (Littlestone 1988), where learning guarantees are stated in terms of the maximum number of mistakes an agent will make in all its predictions. This model offers a worst-case scenario learning guarantee, since it assumes that the order of observations is adversarially selected. Interestingly enough, learning in this model implies learning in the PAC model that we employ in this work (Littlestone 1989).

Our goal in this work was not to present a comprehensive collection of results on the learnability (or lack thereof) of domains from state transitions, but rather to emphasize the need for formal guarantees in the study of learnability, to illustrate that the problem is far from being tractable even under a number of simplifying assumptions, and to highlight certain key aspects of and possible approaches to this problem that warrant further investigation. We hope that this work will help attract more interest in this exciting endeavor.

Acknowledgments

The author would like to thank Leslie Valiant for his advice and encouragement of this research.

References

Aldous, D., and Vazirani, U. 1995. A Markovian extension of Valiant's learning model. *Information and Computation* 117(2):181–186.

Angluin, D. 1988. Queries and concept learning. *Machine Learning* 2(4):319–342.

Doherty, P.; Gustafsson, J.; Karlsson, L.; and Kvarnström, J. 1998. TAL: Temporal action logics language specification and tutorial. *Electronic Transactions on Artificial Intelligence* 2(3–4):273–306.

Gelfond, M., and Lifschitz, V. 1992. Representing actions in extended logic programming. In *Tenth Joint International Conference and Symposium on Logic Programming*, 559–573.

Giunchiglia, E.; Lee, J.; Lifschitz, V.; McCain, N.; and Turner, H. 2004. Nonmonotonic causal theories. *Artificial Intelligence* 153(1–2):49–104.

Goldman, S. A., and Mathias, H. D. 1996. Teaching a smarter learner. *Journal of Computer and System Sciences* 52(2):255–267.

Harel, D. 1984. Dynamic logic. In *Handbook of Philosophical Logic Volume II — Extensions of Classical Logic*. 497–604.

Inoue, K.; Bando, H.; and Nabeshima, H. 2005. Inducing causal laws by regular inference. In *Fifteenth International Conference on Inductive Logic Programming*, 154–171.

Kakas, A. C.; Michael, L.; and Miller, R. 2005. Modular-E: An elaboration tolerant approach to the ramification and qualification problems. In *Eighth International Conference on Logic Programming and Nonmonotonic Reasoning*, 211–226.

Kearns, M. J., and Li, M. 1993. Learning in the presence of malicious errors. *SIAM Journal on Computing* 22(4):807–837.

Kearns, M. J., and Valiant, L. G. 1994. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM* 41(1):67–95.

Kearns, M. J., and Vazirani, U. V. 1994. *An Introduction to Computational Learning Theory*. Cambridge, Massachusetts, U.S.A.: The MIT Press.

Kearns, M. J. 1998. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM* 45(6):983–1006.

Kharitonov, M. 1993. Cryptographic hardness of distribution-specific learning. In *Twenty-fifth ACM Symposium on the Theory of Computing*, 372–381.

Littlestone, N. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning* 2:285–318.

Littlestone, N. 1989. From on-line to batch learning. In *Second Annual Workshop on Computational Learning Theory*, 269–284.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence* 4:463–502.

McCarthy, J. 2006. Appearance and reality. *http://www-formal.stanford.edu/jmc/appearance.html*.

Michael, L. 2007. Learning from partial observations. In *Twentieth International Joint Conference on Artificial Intelligence*, 968–974.

Miller, R., and Shanahan, M. 2002. Some alternative formulations of the Event Calculus. *Lecture Notes in Artificial Intelligence* 2408:452–490.

Otero, R. P. 2005. Induction of the indirect effects of actions by monotonic methods. In *Fifteenth International Conference on Inductive Logic Programming*, 279–294.

Rivest, R. L.; Shamir, A.; and Adleman, L. M. 1978. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM* 21(2):120–126.

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

Valiant, L. G. 1984. A theory of the learnable. *Communications of the ACM* 27:1134–1142.