Causal Logic Models

Denver Dash
ISTC*on Embedded Computing and
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
denver.h.dash@intel.com

Mark Voortman
Decision Systems Laboratory
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, 15260, USA
mark@voortman.name

Abstract

Despite their success in transferring the powerful human faculty of causal reasoning to a mathematical and computational form, causal models have not been widely used in the context of core AI applications such as robotics. In this paper, we argue that this discrepancy is due to the static, propositional nature of existing causality formalisms that make them difficult to apply in dynamic real-world situations where the variables of interest are not necessarily known a priori. We define Causal Logic Models (CLMs), a new probabilistic, first-order representation which uses causality as a fundamental building block. Rather than merely converting causal rules to first-order logic as various methods in Statistical Relational Learning have done, we treat the causal rules as basic primitives which cannot be altered without changing the system. We provide sketches of algorithms for causal reasoning using CLMs, preliminary results for causal explanation, and explore the significant differences between causal reasoning in CLMs and fixed causal graphs, including the non-locality of manipulation and the non-commutability between observation and manipulation.

1 Introduction

Most existing causal models used in AI are based on structural equation modelling [Strotz and Wold, 1960, Simon, 1954, Haavelmo, 1943], a formalism which originated in the econometrics literature and which is still used commonly in the economic and social sciences. The models used in these disciplines typically involve real-valued variables, linear equations and Gaussian noise distributions. In AI these models have been generalized by Pearl [2000] and others [e.g., Spirtes et al., 2000] to include discrete propositional variables, of which Bayesian networks can be viewed as a subset.

There has also been a good deal of awareness that in some systems, the complexity of Bayesian networks can be reduced by exploiting context-specific independence [Boutilier et al., 1996]. This observation typically leads to more efficient learning and inference algorithms, but can also lead to more intuitive explanation of causal systems. For example the causal chain event graphs of Thwaites et al. [2010] generalize Bayesian networks by explictely representing context-specific asymmetries in the network structure.

The overwhelming majority of these causal representations in AI are based on *propositional* logic. One of the key consequences of this fact is that important variables must be known at the time the model is created. In the economic and macroscopic social sciences, this constraint is not a burden. The quantities of interest are typically known and fixed, such as inflation, GDP, unemployment, etc. However, despite their success in transferring the powerful human faculty of causal reasoning to a mathematical and computational form, such causal models have not been widely used in the context of core AI applications such as robotics. In this paper, we argue that this discrepency has to do with their propositional nature, and we provide a probabilistic *first-order* representation which we call *Causal Logic Models (CLMs)* that is more suitable for the real world when events of interests are not known in advance.

Probabilistic first-order representations have been widely studied in the past decade in the context of graphical models, giving rise to an entire sub-field of AI called *statistical relational AI.* [1] While these meth-

*Intel Science and Technology Center

[1] A good overview of this field is provided by Getoor and Taskar [2007].

ods often use rules that are based in causality, they typically convert those rules to first-order logic formulae and thereafter ignore their causal nature. Thus these methods treat logic and probability as the fundamental elements and do not attempt to peform explicit causal reasoning. CLMs, by contrast, are first-order models that treat causality as the fundamental building block. Finally, like causal chain event graphs, CLMs represent context-specificity directly in the structure of the graphical model; however CLMs produce these structures dynamically, in response to observed variables, and produce much simpler structures than causal chain event graphs.

In this paper our contributions are as follows:

- We define a probabilistic first-order representation for causality as an alternative to fixed causal models and build intuition for why these models can handle dynamic situations much better.
- We define methods for performing explicit causal inference such as explanation, prediction and counterfactual reasoning.
- We show that manipulations on CLMs are much different than manipulations in fixed causal models, possibly resulting in global changes to the causal graphs.
- We show that manipulation and observation do not necessarily commute in CLMs.

The paper is organized as follows: in Section 2 we present a motivating example which illustrates the shortcomings of propositional causal models for real-world AI applications. In Section 3 we define CLMs, in Section 4 we discuss several types of causal reasoning we would like to enable with CLMs, and in Section 5 we present sketches of algorithms using CLMs to achieve these types of reasoning, and some preliminary empirical results. Finally, in Section 6 we summarize and discuss future work.

2 Causal Reasoning in Dynamic Situations

To illustrate the type of reasoning we would like to enable, consider the following simplistic running example of "human-like" causal inference:

> *While you are in a business meeting with Tom, Bob suddenly bursts into your office and punches Tom in the face. Tom falls to the ground, then gets up and punches Bob back.*

In this example, there are three main events spaced out in time: *Punch(Bob,Tom,*T_1*)*, *Fall(Tom,*T_2*)*, and *Punch(Tom,Bob,*T_3*)* with $T_1 < T_2 < T_3$. Most humans, given their knowledge of human behavior and physical interactions would have little trouble constructing a causal graph that relates these three events. They might construct the graph of Figure 1(a). They

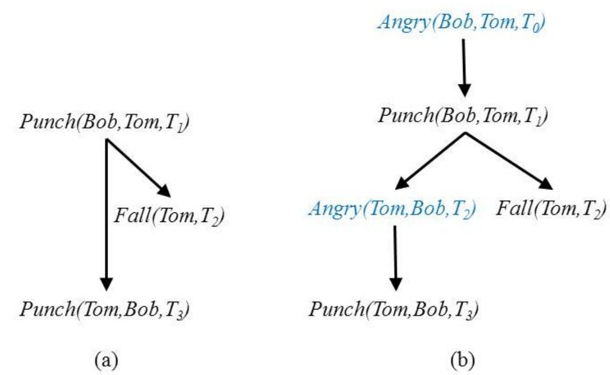

Figure 1: (a) a simple causal explanation (b) a more elaborate causal explanation

may also be able to expand on these observed events to include hypothetical (unobserved) causes as well. For example, they may infer that Bob was Angry prior to T_1 and that Tom was angry after T_2, as shown in Figure 1(b).

In order to model this very simple example using a standard propositional causal model, it quickly gets overwhelmingly complicated. One would first need to have a model that included the variables shown in Figure 1. This would not be possible if Bob was a person that up until now one had no knowledge of. In addition, hypothetically it could have been Tom who barged in on Bob, in which case we would need to essentially mirror the structure of Figure 1 only with Bob and Tom reversed. If we have tens or hundreds of people in the office, this type of model will explode combinatorially. The point is, to have a propositional model pre-built is not a practical solution when the events of interest are part of a dynamic environment where "anything" could happen.

This type of reasoning–taking a general knowledge base about the world, applying it to specific (temporal) facts that have been observed, and reasoning about causality between those facts–is useful, and we would like to create algorithms that can automate it. This particular case is an example of *causal explanation*, and in Section 4 we formalize causal explanation as well as other types of reasoning in the context of our first-order representation. First we define Causal Logic Models.

3 Representation

A Causal Logic Model (CLM) is defined by a set of predicates and a set of formulas. A predicate is specified by a name and a set of argument types. A formula is a causal statement that has a probability associated with it. In Section 3.1 we elaborate on the Office Brawl example given previously, and in Section 3.2, we present formal definitions.

3.1 The Office Brawl System

In Section 1 we presented a simple example of an office brawl system to build intuition for the dynamic environment we wish to handle. Here we provide more details for this example system. The predicates for this system might look like:

$$\begin{aligned}&\mathit{Angry}(\mathit{Person}, \mathit{Person}, \mathit{Time})\\&\mathit{Punch}(\mathit{Person}, \mathit{Person}, \mathit{Time})\\&\mathit{Fall}(\mathit{Person}, \mathit{Time}).\end{aligned}$$

Each predicate is indexed by a discrete time index. The set of formulas could be chosen as

$$0.25 : \mathit{Angry}(P_1, P_2, T_1) \longrightarrow \mathit{Punch}(P_1, P_1, T_2) \quad (1)$$
$$0.4 : \mathit{Punch}(P_1, P_2, T_1) \longrightarrow \mathit{Fall}(P_2, T_2) \quad (2)$$
$$0.9 : \mathit{Punch}(P_1, P_2, T_2) \longrightarrow \mathit{Angry}(P_2, P_1, T_2) \quad (3)$$
$$0.9 : \mathit{Punch}(P_1, P_2, T_2) \longrightarrow \neg\mathit{Angry}(P_1, P_2, T_2) \quad (4)$$
$$0.8 : \mathit{Angry}(P_1, P_2, T_1) \longrightarrow \mathit{Angry}(P_1, P_2, T_2) \quad (5)$$
$$0.1 : \mathit{Angry}(P_1, P_2, T_1) \quad (6)$$
$$0.01 : \mathit{Punch}(P_1, P_2, T_1) \quad (7)$$
$$0.01 : \mathit{Fall}(P_1, T_1). \quad (8)$$

Formula 1 indicates that if someone is angry then they are more likely to punch. Formula 2 embodies the principle that someone who gets punched may fall down. Formula 3 and 4 expresses the idea that the person who gets punched will get more angry, but the puncher will get less angry. Formula 5 makes anger persistent over time. Formula 6, 7, and 8 are simply prior probabilities for each of the predicates.

We define two types of formulaes: *causal formulae* and *prior formulae*. Causal formulae express a causal relationship between a set of causes and an effect. All predicates that match the set of causes in a formula must have the same time index T_1, and the predicate matching the effect must have a time index T_2 such that $T_1 < T_2$. Causal formulae are drawn from the subset of first-order logic that includes formulae containing a conjunction of (possibly negated) causes relating to a single (possibly negated) single effect. Prior formulas are composed of a single predicate with no causal claim, and are meant to express the prior probability of events occurring. Prior formulae are necessary in CLMs to be able to model cases where causal formulae are not present, and we demand that each predicate contain a prior formulae.

Restricting the form of the formulas is common in computational logic systems (e.g., Prolog makes the even more restrictive assumption of only allowing Horn clauses), and we do it here mainly for convenience. We intend to relax these constraints on formulas in future work.

The predicates and formulas given above define a CLM. As we make observations about our system, these formulae provide possible explanations that can tie those observations together causally. The algorithm that we present in Section 5 accomplishes this by searching for the the structure with maximum likelihood given the evidence. In order to score the likelihood of structures, we first need to convert the CLM into a Bayesian network, and calculate the probability of the evidence given the structure. Figure 1(b) gives a possible explanation for the office brawl scenario.

Figure 1(b) shows the cause and effect relationships between predicates but it does not specify how probabilities for the given predicates could be calculated. In order to construct a Bayesian network that can be used to score a hypothesis, it is necessary to convert the formulae from a CLM into conditional probability tables. In other statistical relational methods, similar conversions are typically done using *combination rules*, of which *noisy-or* is the most popular. We also use noisy-or in the work presented here, but we feel this combination rule is not fully adequate, and we are actively looking for better ones.

As an example, the CPT for the root node $P(\mathit{Angry}(\mathit{Bob}, \mathit{Tom}, T_0))$ is given by the prior probability of a person being angry (Formula 6). The CPT for predicate $\mathit{Punch}(\mathit{Bob}, \mathit{Tom}, T_1)$ is defined by combining Formula 1, that is, $P(\mathit{Punch}(\mathit{Bob}, \mathit{Tom}, T_1) | \mathit{Angry}(\mathit{Bob}, \mathit{Tom}, T_0)) = 0.7$, with the prior formula for Punch.

3.2 Formal Definitions

Here we make a first attempt at formally defining CLMs.

The key constituent of CLMs are *predicates* and *formulas*. Predicates in CLMs are n-ary and typed. We define two types of formulae, *causal formulae* and *prior formulae*.

Definition 1 (causal formula)**.** *A causal formula is a* *formula of the form:*

$$p : C_1 \wedge C_2 \wedge \ldots, C_n \longrightarrow E,$$

where the C_i and E are (possible negated) predicates and $0 \leq p \leq 1$. The symbol $\longrightarrow$ indicates that the C_i variables cause E in the future. The time index T_1 for each cause must agree with the other causes, and the time index T_2 for E must obey $T_1 < T_2$.

Because $\longrightarrow$ denotes causality and not merely logical implication, it is not possible in general to apply rules of logic to rewrite the form of this equation. This restriction is similar to the restriction placed on structural equation models where systems of structural equations cannot be algebraically manipulated into equivalent systems without altering the structure, and it is this structural restriction that makes CLMs unique compared to other probabilistic first-order methods.

Definition 2 (prior formula)**.** *A prior formula is a formula of the form:*

$$p : X,$$

where X is a non-negated predicate and $0 \leq p \leq 1$.

In the rest of this paper, we assume that p is defined as $P(E|C_1, C_2, \ldots, C_n)$ for causal rules and as $P(X)$ for prior rules under the assumption that there are no other causes. This assumption possibly impacts and is impacted by the choice of combination rule, but for noisy-or, we can ensure it to be the case.

Using these definitions, we can now define a CLM:

Definition 3 (Causal Logic Model)**.** *A Causal Logic Model is a pair $\langle \mathbf{P}, \mathbf{F} \rangle$, where $\mathbf{P}$ is a set of predicates, $\mathbf{F}$ is a set of formulas over $\mathbf{P}$.*

An constant is a primitive symbol over which we wish to reason. A *ground predicate* $\hat{P}$ is a predicate with all variables instantiated with constants. Similarly, a *ground formula* $\hat{F}$ is a formula with all predicates grounded.

Definition 4 (Causal Logic Database)**.** *A causal logic database is a pair $\langle \mathbf{C}, \hat{\mathbf{P}} \rangle$ where $\mathbf{C}$ is a set of constant and $\hat{\mathbf{P}}$ is a set of predicates grounded with constants in $\mathbf{C}$.*

We call a set of grounded formulas a *Causal Logic Hypothesis* or *Hypothesis* for short. Finally, a Hypothesis defines a directed acyclic graph which we call a *Causal Logic Network* as follows:

Definition 5 (Causal Logic Network (CLN))**.** *Given a Causal Logic Hypothesis $\hat{\mathbf{F}}$ a CLN $G_C = \langle \mathbf{N}, \mathbf{E} \rangle$ is a directed acyclic graph, where $\mathbf{N}$ is a set of nodes and $\mathbf{E}$ is a set of directed edges, such that a node exists for each ground predicate appearing in a formula $F \in \hat{\mathbf{F}}_{\hat{\mathbf{P}}}$ and an edge $E_{12} \equiv P_1 \rightarrow P_2 \in \mathbf{E}$ iff P_1 is a cause of P_2 in a formula $F \in \hat{\mathbf{F}}_{\hat{\mathbf{P}}}$.*

4 Causal Reasoning with CLMs

In this section, we define the concept of *causal reasoning.* Often this concept is partitioned into different sub-classes such as *explanation*, *prediction* and *counterfactual reasoning* using operators such as *observation* and *manipulation.* Although these types of reasoning have been discussed at length elsewhere [c.f. Pearl, 2000], here we relate these concepts to CLMs and we raise several new issues that arise in this context such as the commutability between observation and manipulation.

In general, causal reasoning with CLMs is the act of inferring a causal structure relating events in the past or future. The events themselves can be observed, hypothesized (i.e., latent) or manipulated. Given a CLM C and a sequence of events $\mathbf{E_1}, \mathbf{E_2}, \ldots, \mathbf{E_n}$, all causal reasoning can be cast into the problem of finding a most-likely structure $\hat{S}$ given a set of information:

$$\hat{S} = \arg\max_S P(S|C, \mathbf{E_1}, \mathbf{E_2}, \ldots \mathbf{E_n})$$

We consider sequences of events rather than one big set of events $\mathbf{E} = \cup_i \mathbf{E_i}$ because when we consider manipulation of the system, then it will sometimes be the case that manipulation does not commute with observation. So we need to preserve the sequence in which events are observed and manipulated. We discuss this issue more in Section 4.3 below, but first we show how our definition of causal reasoning performs explanation, prediction and counterfactual reasoning.

4.1 Causal Explanation

We have already shown an example of causal explanation in Figure 1. Roughly speaking, causal explanation seeks to explain a set of *observed events* in terms of the *hypothesized latent events* that caused those observations. If we have multiple observed events spaced out in time, causal explanation may produce hypotheses about how those events are related causally. Figure 2(a) shows a schematic example of causal explanation.

Formally, we define causal explanation as the most likely causal graph given a CLM and a set of evidence:

Definition 6 (causal explanation)**.** *Given a CLM C and a set of observed events $\mathbf{E} = \{[E_1, T_1], [E_2, T_2], \ldots, [E_n, T_n]\}$, a causal explanation is a directed acyclic graph $\hat{S}$ such that $\hat{S} = \arg\max_s P(S = s|C, \mathbf{E})$.*

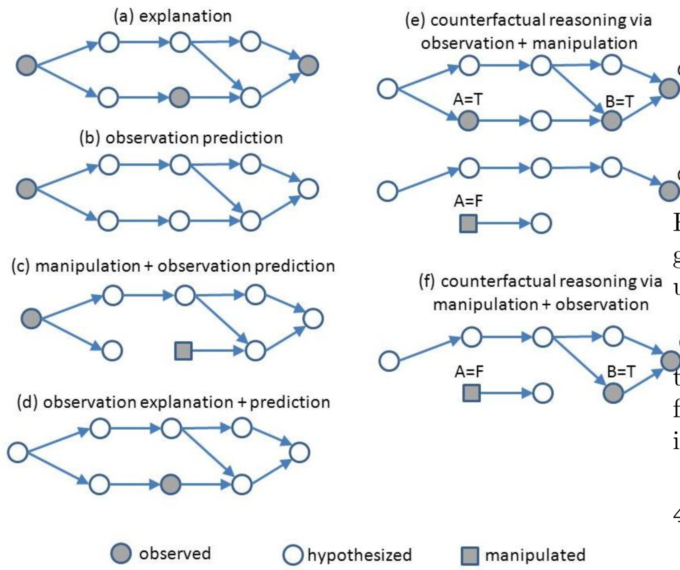

Figure 2: Different types of causal reasoning generated by combining explanation, prediction, observation and manipulation.

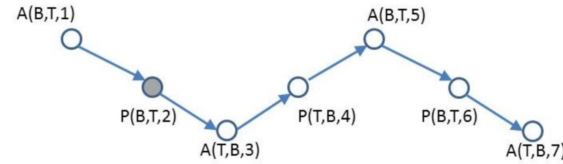

Figure 3: Causal prediction produces the most likely graph relating known events in the present or past to unknown (hypothesized) events in the future.

Compare this definition to that used in standard causal modeling where an explanation is a joint state of the variables represented by a fixed causal graph. Thus in standard causal modeling, no matter what was observed, the causal structure does not change, only likely states of variables change. In CLMs a causal explanation is the graph itself. Not only does this feature allow us to reason about previously unknown constants, but it produces explanations that are more like the types of explanations humans would give, i.e., a causal graph that changes depending on the observations.

4.2 Causal Prediction

Causal Prediction is the act of predicting what sequences of cause and effect will occur in the future given evidence observed in the past. For example, we may predict, given that Bob punches Tom at time 2 that at time 3 Tom will be angry with Bob and at time 4 Tom will punch Bob. This may in turn cause Bob to get Angry at Tom, thus repeating the cycle indefinitely. This graph is shown in Figure 3. Formally, we define causal prediction in a way similar to causal explanation as the most likely graph extending into the future given a CLM and a set of evidence in the present or past. Prediction is not restricted to only inferring events in the future. In practice, the events in the past that led to the observations in the present may be relevant for predicting future variables as well, so we must perform inference on past events in order to better predict the future. In general, the distinction between explanations, predictions, and counterfactuals is somewhat arbitrary and can be combined in various ways as seen in Figure 2.

4.3 Counterfactuals and Manipulation in CLMs

One key concept in causal reasoning is understanding the effects of manipulating variables in the model. In the propositional directed graph representations of causality, this is accomplished, for example, by the *Do* operator of Pearl, which modifies the fixed causal structure by cutting all arcs coming into a node that is being manipulated to a value. Inference results can change depending on whether the state of some evidence is determined by mere observation or by active manipulation.

Manipulation in CLMs is quite different in terms of its effect on the causal structure. Rather than operating on graphs, manipulation in CLMs operates on formulae: if a variable X is manipulated to some value, then all formula that normally would cause X to achieve that value get struck from the model. This can have very non-local effects on the most likely structure $\hat{S}$ that results.

To see the non-local effects of manipulation in CLMs, consider the example in Figure 4. Figure 4(a) shows a typical causal explanation when $Punch(B,T,2)$ and $Punch(T,B,5)$ (circular gray nodes) are observed given the Office Brawl system presented in Section 3. The circular clear nodes are hypothesized states that connect the observed states and thus increase the probability of the evidence. In Figure 4(b), we manipulate the system by setting $Punch(B,T,2) = False$. In this example, since Bob does not punch Tom, then his anger persists based on the "persistence" formula given in Equation 5. Furthermore, all the formulae that were fulfilled by $Punch(B,T,2) = True$ are now no longer valid, so all the children of $Punch(B,T,2)$ are altered in addition to its parents. What is left is a causal structure that looks little like the original one prior to manipulation.

Another important observation comes out of this ex-

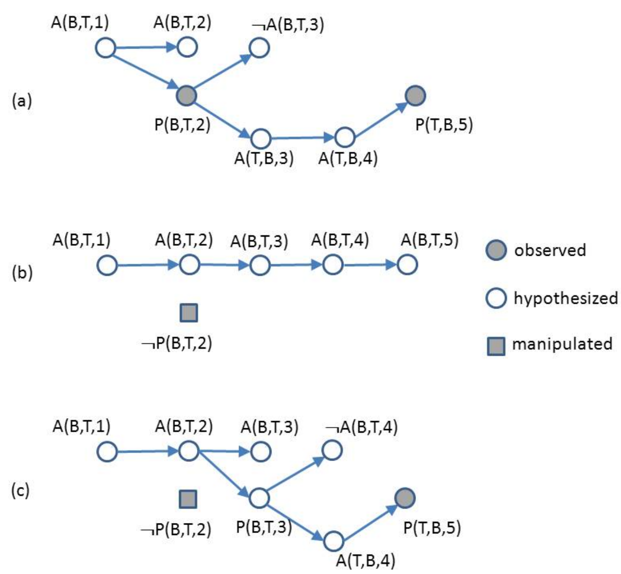

Figure 4: An example of counterfactual reasoning with CLMs using the Office Brawl model. $A \equiv Angry$, $P \equiv Punch$, $B = Bob$ and $T = Tom$. (a) The causal explanation when Bob punches Tom at time 2 and Tom punches Bob at time 5. (b) Unlike with fixed causal graphs, manipulation of variables can cause sweeping global changes to the structure. (c) Manipulation and observation do not commute: Manipulation followed by observing $Punch(Tom, Bob, 5)$ produces a much different graph than observing $\{Punch(Bob, Tom, 2), Punch(Tom, Bob, 5)\}$ and then manipulating $Punch(Bob, Tom, 2) = False$.

ample: unlike for fixed-structure causal models, manipulation and observation in CLMs do not commute. To see this, imagine that after manipulating $Punch(Bob, Tom, 2) = False$ we then still proceeded to observe $Punch(Tom, Bob, 5)$. In this case, given Bob's persistent state of anger, a likely explanation may very well be that Bob punched Tom in a later time causing Tom to get angry and punch back. This feature of very different causal structures resulting based on different combinations of observation and manipulation seem to square much better with human causal reasoning than the method of fixed graphs.

To our knowledge, the issues of the lack of commutability between observation and manipulation has not been addressed elsewhere. This result is similar to the violation of *Equilibration-Manipulation Commutability* presented in Dash [2005]; however in our case no equilibration is required for violation of commutability.

5 Algorithm and Preliminary Results

In this section we describe an algorithm that finds an explanation given an evidence set, and we then apply this algorithm to the office example. The algorithm is a greedy search algorithm that first thickens the graph and then thins it based on a score function that is defined below. The basic steps are the following:

1. Start with an empty graph.
2. Iteratively add the predicate that increases the score most and continue until adding a predicate decreases the score.
3. Iteratively remove the predicate that increases the score most and continue until removing a predicate decreases the score.
4. Find the states that constitute the most likely explanation and find all grounded formulae $\hat{\mathbf{F}}$ consistent with those states.
5. Prune from the graph all variables that are not part of some $F \in \hat{\mathbf{F}}$.

Our goal is to find the structure with the highest posterior probability $P(G|e)$ given the evidence e. We approximated this quantity by the likelihood $P(e|G)$, which amounts to assuming flat priors on graph structure. This quantity can be calculated with standard Bayesian network inference algorithms. The underlying idea of this score function is that variables should only be included in the graph if they somehow affect the evidence, which is what we are trying to explain, or otherwise they should be left out.

Steps 4 and 5 of our algorithm apply a post-pruning step to the graph by finding the most likely configuration of all the variables in the network, keeping the evidence variables fixed. This configuration already constitutes an explanation but it can easily be simplified. Instead of looking at the states of the variables individually we can look at formulas that are instantiated in the graph, and we remove all the variables that are not part of such an instantiated formula. The remaining graph is the final explanation. Note that the graphical form of this explanation can be easily translated back into formulas, making it also relatively easy to generate a textual explanation.

As a preliminary test of this algorithm, we applied it to the office scenario introduced in an earlier section. We used $Punch(Bob, Tom, T_1) = True$, $Fall(Tom, T_2) = True$, and $Punch(Tom, Bob, T_3) = True$ as evidence. The explanation returned by the algorithm corresponds exactly to the network in Figure 1(b), with all states being true. It is now also possible to determine the posterior probabilities of the hidden variables (we clear their map states first), which are $Angry(Bob, Tom, T_1) = 0.741$ and $Angry(Tom, Bob, T_3) = 0.996$. The second probability is much higher, because both the preceding and following punch increase the likelihood of being angry.

Without the search and pruning of variables the output would be much more verbose. One would have to create a complete network of all predicates and then calculate the most likely explanation. But this would contain a lot of information that is not relevant in the explanation. In addition, when our set of constants grows very large, instantiating an entire network over them may be prohibitively memory intensive. Thus, our algorithm finds a succinct yet complete explanation of a given scenario with reasonable memory constraints.

6 Summary and Future Work

In this paper we introduced CLMs which are a novel representation for causal reasoning. To our knowledge, CLMs are the first probabilistic first-order system to model causality as the basic primitive rather than logic or probability. This approach is analagous to that of structural equation models which encode causality in the form of equations and cannot be algebraically manipulated without changing the semantics of the model. In the same way, our formulae cannot be undergo arbitrary logical transformations without a similiar violation of semantics.

We discussed several types of causal reasoning with CLMs such as explanation, prediction and counterfactuals, and defined a general version of causal reasoning that subsumes all of these. We showed that manipulation in CLMs is qualitatively different from the *Do* operator applied to fixed propositional causal models; in particular, manipulations can cause sweeping global changes to the causal graph, and manipluation may not commute with observations. We also presented an algorithm that can generate causal explanations from a given evidence set, and verified on a simple test case that it produces expected results.

Future work includes expanding our explanation algorithm to general causal reasoning, developing more intuitive combination rules when transforming CLMs to Bayesian networks for scoring, and learning CLM formulae from data.

References

Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)*, pages 115–123, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

Denver Dash. Restructuring dynamic causal systems in equilibrium. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AIStats 2005)*. Society for Artificial Intelligence and Statistics, 2005.

Lise Getoor and Ben Taskar, editors. *Introduction to Statistical Relational Learning*. Adaptive Computation and Machine Learning. The MIT Press, 2007.

Trygve Haavelmo. The statistical implications of a system of simultaneous equations. *Econometrica*, 11(1):1–12, January 1943.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2000.

Herbert A. Simon. Spurious correlation: A causal interpretation. *Journal of the American Statistical Association*, 49(267):467–479, September 1954.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, Cambridge, MA, second edition, 2000.

Robert H. Strotz and H.O.A. Wold. Recursive vs. nonrecursive systems: An attempt at synthesis; Part I of a triptych on causal chain systems. *Econometrica*, 28(2):417–427, April 1960.

Peter Thwaites, Jim Q. Smith, and Eva Riccomagno. Causal analysis with chain event graphs. *Artificial Intelligence*, 174(12-13):889–909, August 2010.