Sequences of Mechanisms for Causal Reasoning in Artificial Intelligence

Denver Dash
Intel Corporation and
Carnegie Mellon University
USA
denver.h.dash@intel.com

Mark Voortman
University of Pittsburgh
USA
mark@voortman.name

Martijn de Jongh
University of Pittsburgh
USA
mad159@pitt.edu

Abstract

We present a new approach to token-level causal reasoning that we call *Sequences Of Mechanisms (SoMs)*, which models causality as a dynamic sequence of active mechanisms that chain together to propagate causal influence through time. We motivate this approach by using examples from AI and robotics and show why existing approaches are inadequate. We present an algorithm for causal reasoning based on SoMs, which takes as input a knowledge base of first-order mechanisms and a set of observations, and it hypothesizes which mechanisms are active at what time. We show empirically that our algorithm produces plausible causal explanations of simulated observations generated from a causal model.

We argue that the SoMs approach is qualitatively closer to the human causal reasoning process, for example, it will only include relevant variables in explanations. We present new insights about causal reasoning that become apparent with this view. One such insight is that observation and manipulation do not commute in causal models, a fact which we show to be a generalization of the Equilibration-Manipulation Commutability of [Dash(2005)].

1 Introduction

The human faculty of causal reasoning is a powerful tool to form hypotheses by combining limited observational data with pre-existing knowledge. This ability is essential to uncovering hidden structure in the world around us, performing scientific discovery and diagnosing problems in real time. Enabling computers to perform this kind of reasoning in an effective and general way is thus an important sub-goal toward achieving Artificial Intelligence.

The theoretical development of causality in AI has up to now primarily been based on structural equation models (SEMs) [Strotz and Wold(1960); Simon(1954); Haavelmo(1943)], a formalism which originated in econometrics 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 this theory has been generalized by [Pearl(2000)] and others (e.g., [Spirtes et al.(2000)Spirtes, Glymour, and Scheines]) to include discrete propositional variables, of which (causal) Bayesian networks (BNs) can be viewed as a subset because they are acyclic. Yet, despite decades of theoretical progress, causal models have not been widely used in the context of core AI applications such as robotics.

Because causality in econometrics operates at the population level, important human causal faculties that operate on *single entities* may have been neglected in current causality formalisms. Such single-entity cases are common in AI: A robot gets stuck in a puddle of oil, a person has chest pains and we want to know the specific causes, etc. This distinction between population-level and individual-level causation is known in the philosophical literature as *type-level* versus *token-level* causation (e.g., [Eells(1991); Kleinberg(2012)]) or *general* versus *singular* causation (e.g., [Hitchcock(1995); Davidson(1967)]). For example, a type-level model for lung cancer might include all the possible causes, such as: *CigaretteSmoking*, *AsbestosExposure*, *GeneticFactors*, etc., whereas a token-level explanation contains only actual causes: "Bob's lung cancer was caused by that time in the 80s when he snorted asbestos."

Models in econometrics rely less on token causality and more on type-level reasoning. Causality-in-AI's evolution from these disciplines may explain why token-level causal reasoning has been less studied in AI. However, in many AI tasks such as understanding why a robot hand is stuck in a cabinet, this ability may be crucial to posing and testing concrete causal hypotheses. This disconnect may further explain why causal reasoning has up to now not been widely used in the context of robotics and AI applications.

In this paper, we consider the tasks of producing token-level explanations and predictions for causal systems. We present a new representation for these tasks which we call *Sequences of Mechanisms* (SoMs). We motivate SoMs with several examples for how they improve causal reasoning in typical AI domains. We present an algorithm for the construction of SoMs from a knowledge base of first-order causal mechanisms, and we show empirically that this algorithm produces good causal explanations. The first-order nature of our mechanisms facilitates scalability and more human-like hypotheses when the possible number of causes is large.

There has been some work on representation and al-

gorithms for token-level causal reasoning. In particular, [Halpern and Pearl(2005)] present a definition of *causal explanation* which uses the concept of *actual cause* from [Halpern and Pearl(2001)], based on functional causal models of [Pearl(2000)], to produce sets of variables which are deemed to be possible explanations for some evidence. We discuss some shortcomings of token causality with functional causal models such as BNs in Section 2. In particular, we show that in order for this representation to be general, it must essentially be reduced to the approach presented in this paper. [Kleinberg(2012)] discusses token causality explicitly and presents a measure of significance for a token-level immediate cause given logical formulae which are similar syntactically to our mechanisms; however she does not attempt to find optimal chains of causation which could serve as non-trivial hypotheses. Furthermore, both Halpern and Pearl's and Kleinberg's approaches are fundamentally propositional in nature, so lack our ability to scale when the number of possible causes is large.

2 Sequences of Mechanisms

To illustrate the type of reasoning we would like to enable, consider the following simple 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, which we will refer to as the Office Brawl 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 $T_1 < T_2 < T_3$. Humans, given their wealth of background knowledge might construct the graph of Figure 1(a). They may also be able to expand on these observed events to

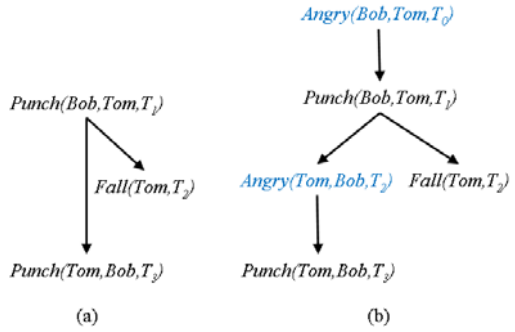

Figure 1: (a) A simple causal explanation. (b) A more elaborate causal explanation with hypothesized events.

include hidden events, such as *Angry*, of which a causal graph is displayed in Figure 1(b).

Although the explanations in Figure 1 take the form of a directed graph, there is much more to the causal explanation. The edges from *Angry*$\longrightarrow$*Punch* refers to a physical mechanism whereby a person succumbs to his sense of anger and punches someone. There could be many other mechanisms that result in a *Punch* without being *Angry*, such as a boxing match, and we would like to identify a specific mechanism that is becoming active when someone delivers a *Punch*. We argue in this section that mechanisms such as these are more fundamental than a state-space representation which only deals with sets of variables and their states.

In many real-world problems, having a first-order representation for token-level reasoning is essential. For example, the method of [Halpern and Pearl(2005)] can generate explanations, but only once a "complete" causal graph enumerating all possible causal ancestors of an event is specified. In the office brawl scenario as well as other real-world simple scenarios such as a pool table where there are an intractable or even infinite number of possible events that could cause other events to happen, building this "complete" causal model may be impossible or impractical.

This limitation aside, in BNs and SEMs, something similar to token-level reasoning can be performed by instantiating variables in the model to values based on the specific instance at hand. For example, in the lung cancer case of Section 1, one might instantiate *LungCancer* to *True* and *AsbestosExposure* to *True* to indicate the token-level hypothesis that Bob's asbestos snorting adventure caused his lung cancer. However, this *state-space* approach is incomplete: being able to reason only about states and possible causes is different from creating specific hypotheses about how an event was caused. For example, it could very well be that Bob was a smoker, but the smoking was not the cause of his lung cancer. In this case, the BN model is unable to distinguish (and score) between the three hypotheses: *Smoking*$\rightarrow$*LungCancer*, *AsbestosExposure*$\rightarrow$*LungCancer* and *Smoking & AsbestosExposure*$\rightarrow$*LungCancer*.

This problem becomes exacerbated in a more realistic causal model where the number of nodes would be much higher, and where nodes would combine in nontrivial ways. Consider, for example, the BN causal model of Figure 2(a) where all nodes are binary. Given some evidence we can obtain beliefs about the states of all the nodes in the graph (say dark represents *False* and light represents *True*). This representation of a type-level causal graph plus specific states does not necessarily provide us with a clear token-level picture of what is happening causally in this system. On the other hand, a token-level explanation represented by a subgraph (such as that shown in Figure 2(b) showing the likely causal ancestors of the event of interest provides a much clearer causal picture. If one wanted to consider, for example, which manipulations might change the outcome of the event of interest, the graph of Figure 2(b) would be much more informative. This suggests one possible algorithm to achieve such a token-level explanation which looks strikingly similar to a structure search given data, with two important differences: (1) the data of interest here is a single record, and (2) we have strong priors on the structure, being provided by the full type-level causal model.

Starting with a "complete" BN model and looking for active sub-graphs as possible token-level hypotheses still lacks the expressiveness required for producing many token-level hypotheses. As an example, if some effect can be caused by two mechanisms involving the same set of causes, then unless those mechanisms are explicitly represented, the state-space

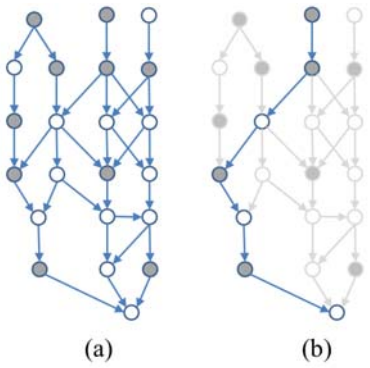

Figure 2: The state-space description of specific events (a) and a token-level explanation (b). Dark nodes are believed to be false and light nodes are believed to be true).

representation will lump both mechanisms together. Any hypothesis generated from this model will consist of some kind of mixture of two very different mechanisms. It may be possible to construct a BN in such a way that the individual mechanisms are preserved, but the point is, the mechanisms *must* be preserved to do token-level reasoning, and therefore, the mechanisms themselves should play an explicit role in the representation used.

2.1 Representation

We advocate a representation that consists of a collection of causal mechanisms that capture the causal interactions in a dynamic system. These causal mechanisms are formulae encoding and quantifying causal implication, i.e., what is the probability that an effect is true given that all its causes are true. The mechanisms are assumed to *work independently* of each other and the probability specificied with each formula encodes the likelihood of that mechanism causing the effect given that *all other mechanisms are absent*. We say that a formula *matches* when all its causes are present, and when these causes actually bring about the effect we say that it is *active*.

For the example of Figure 1, we have a set of predicates consisting of $Angry(Person, Person, Time)$, $Punch(Person, Person, Time)$, and $Fall(Person, Time)$. Each predicate is indexed by a discrete time index. The set of formulae could be chosen as

$$0.25 : Angry(P_1, P_2, T_1) \longrightarrow Punch(P_1, P_1, T_2) \quad (1)$$
$$0.4 : Punch(P_1, P_2, T_1) \longrightarrow Fall(P_2, T_2) \quad (2)$$
$$0.9 : Punch(P_1, P_2, T_1) \longrightarrow Angry(P_2, P_1, T_2) \quad (3)$$
$$0.95 : Punch(P_1, P_2, T_1) \longrightarrow \neg Angry(P_1, P_2, T_2) \quad (4)$$
$$0.2 : Angry(P_1, P_2, T_1) \longrightarrow \neg Angry(P_1, P_2, T_2) \quad (5)$$
$$1 : Fall(P_1, T_1) \longrightarrow \neg Fall(P_1, T_2) \quad (6)$$

The above formulae express causal relationships that govern the self-perpetuating cycle of violence between two individuals. We assume that times present on the left side of a formula occur at an earlier time than all times on the right, and in this paper we will assume that $T_2 = T_1 + 1$. For simplicity, we assume that 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) effect. We intend to relax these constraints on formulae in future work. It is assumed that the state of a predicate persists over time if there are no mechanisms actively influencing it, so Formula 6 prevents repeated falling behavior. A causal formula is called *promoting* if the effect predicate is not negated. A causal formula is called *inhibiting* if it is not promoting.

If no formula matches a predicate we assume that it will maintain its state. If there is exactly one formula that matches, the probability of the effect is simply defined by the probability associated with the formula; however if a set $\mathbf{F}$ of several mechanisms are active for the same effect, we use a combination rule to produce the probability of the effect. When all mechanisms $F_i \in \mathbf{F}$ being combined are of the same type (promoting or inhibiting), then we apply the well-known noisy-or [Good(1961); Peng and Reggia(1986)]:

$$P(E|\mathbf{F}) = 1 - \prod_{F_i \in \mathbf{F}} (1 - p_i),$$

where p_i is the probability associated with formula F_i. When $\mathbf{F}$ is made up of inhibiting formulae, then the noisy-or combination determines the complement of E. In the case where we are combining inhibiting formulae ($\mathbf{F}^-$) with promoting formulae ($\mathbf{F}^+$), we average the noisy-or combination of all promoters with the complement of the noisy-or combination of all inhibitors:

$$P(E|\mathbf{F}^+, \mathbf{F}^-) = \frac{P(E|\mathbf{F}^+) + 1 - P(\neg E|\mathbf{F}^-)}{2}.$$

As we make observations about our system, these formulae provide possible explanations that can tie those observations together causally. In Section 3 we present an algorithm that accomplishes this by searching for a structure that explains all the observations.

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] with many variants. Many of these variants might be adaptable to produce mechanistic interpretations simply by demanding that rules are comprised of isolated mechanisms, and by producing causal hypotheses that only include lists of ground formulae which relate predicates over time.

One representation in particular, the *CP-Logic* formalism of [Vennekens et al.(2009)Vennekens, Denecker, and Bruynooghe] combines logic programming with causality, and they explicitly discuss this representation in the context of counterfactuals and manipulation [Vennekens et al.(2010)Vennekens, Bruynooghe, and Denecker]. Our representation is very similar to CP-Logic, with temporal rules and slightly different syntax. To our knowledge CP-Logic has not been used for token-level explanation/prediction previously.

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

2.2 Reasoning

Causal reasoning as we define it produces hypothesized *sequences of causal mechanisms* that seek to *explain* or *predict* a set of *real or counterfactual* events which have been *observed* or *manipulated*. We therefore maintain that there are three independent dimensions to causal reasoning: *explanation/prediction*, *factual/counterfactual*, *observation/manipulation*. In this section, we look at causal explanation, prediction, counterfactuals, and manipulations. Although these types of reasoning have been discussed at length elsewhere (e.g., [Pearl(2000)]), here we relate these concepts to token-based causality and we raise several new issues that arise in this context such as the commutability between observation and manipulation.

In general, causal reasoning 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 causal model 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 sequence of mechanisms $\hat{S}$ given a set of information:

$$\hat{S} = \arg\max_{S} P(S|C, \mathbf{E_1}, \mathbf{E_2}, \ldots \mathbf{E_n}) \quad (7)$$

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 2.3 below.

Causal Explanation is the act of explaining a set of observations in terms of a set of mechanisms. This is defined by Equation 7 where the sequence of events $\mathbf{E}$ only contains observations and no manipulations. 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(a). 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 and will be discussed next.

2.3 Observation-Manipulation Commutability

One key concept in causal reasoning is understanding the effects of manipulating variables in the model. When a complete causal model is specified, then this can be accomplished with 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, but the effect on the structure is always the same.

However, when the complete model cannot be specified, we must re-generate the causal explanation after some manipulation is performed. Thus, in SoMs, rather than operating on graphs, manipulation in SoMs operates on formulae: if a variable X is manipulated to some value, then all formulae that normally would cause X to achieve that value get struck from the model, and formulae that required X as a cause are also removed. This can have very non-local effects on the most likely structure $\hat{S}$ that results.

For conciseness, in the rest of this section, we use, e.g., $P(B, T, t)$ to indicate $Punch(Bob, Tom, t)$, etc. Figure 3(a) shows a typical causal explanation when $P(B, T, 2)$ and $P(T, B, 5)$ (circular gray nodes) are observed given the Office Brawl system presented in Section 2.1. The circular clear nodes are hypothesized states that connect the observed states

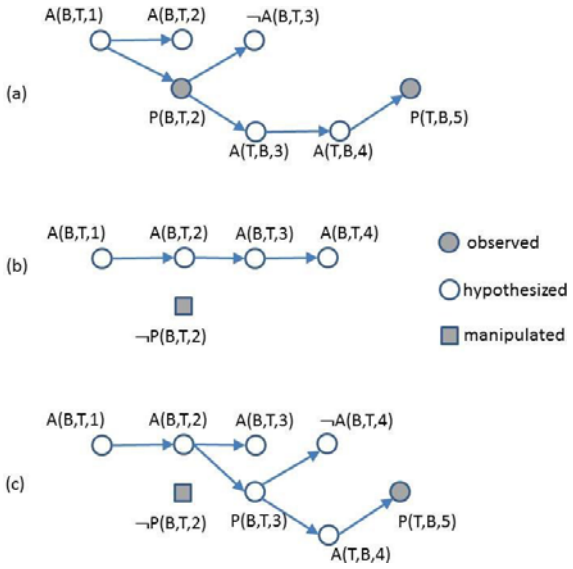

Figure 3: An example of three explanations given (a) two observed events, (b) two observed events followed by manipulation, and (c) two observed events, followed by manipulation, followed by a repeated observation of $P(T, B, 5)$. $A \equiv Angry$, $P \equiv Punch$, $B = Bob$ and $T = Tom$.

and thus increase the probability of the evidence. In Figure 3(b), we manipulate the system by setting $P(B, T, 2) = False$. 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 $P(B, T, 2) = True$ are now no longer valid, so all the children of $P(B, T, 2)$ are altered in addition to its parents. What is left is a causal structure that looks quite different from the one prior to manipulation. This feature of token-level causality that very different structures can be obtained from different manipulations seem to square much better with human causal reasoning than the method of fixed graphs.

Another important observation comes out of this example: manipulation and observation do not commute. This fact becomes apparent with SoMs, because we can observe the difference so vividly in the causal structure. To see this, imagine that after manipulating $P(B, T, 2) = False$

we then still proceeded to observe $P(T, B, 5)$, as in Figure 3(c). 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. Thus, the probability of say, $P(B, T, 3)$ given that we observed $O_a \equiv \{P(B, T, 2), P(T, B, 5)\}$ followed by manipulating $M_b \equiv \neg P(B, T, 2)$ is lower than if we observe O_a followed by manipulating M_b followed by observing *again* $P(T, B, 5)$. This lack of commutation is why we must specify events as a sequence instead of a set, as in Equation 7. It should also be emphasized that this lack of commutation holds in BNs as well, as a trivial exercise will reveal.

To our knowledge, the issues of the lack of commutability between observation and manipulation in general has not been addressed elsewhere. The issue of *Equilibration-Manipulation Commutability* presented in [Dash(2005)], is similar, and in fact is a special case of what we call *Observation-Manipulation Commutability*. In the former case, the observation being made results from the equilibration of certain variables. When a manipulation is made, it changes the downstream causality, which in turn can change the equilibrium values of some variables.

3 Algorithm

In this section we present an algorithm for causal reasoning based on SoMs, shown in Figure 4, which takes as input (a) a knowledge-base of mechanisms and a set of observations, and outputs a hypothesis (d) about which mechanisms were active at what time. The basic idea is to convert a causal model and a set of evidence into a BN (b), find (c) the Most Probable Explanation (MPE) for that evidence, and select all the formulae that are consistent with the MPE states to recover (d) the final sequence of grounded mechanisms. We will call this last step pruning because it effectively removes all formulae from the explanation/prediction that are not relevant. We will now describe each of the three steps in detail.

Step (a) to (b): A set of formulae and a set of evidence are converted into a Bayesian network in the following way. First, for each evidence predicate all the formulae that (partially) match that predicate are instantiated in all possible ways. Consequently, this results in an expanded set of predicates that can then be used to find additional formulae that match, just like in the first step. We call formulae that are added to the model instantiated formulae (because all their free variables are bound). This process continues until no more predicates can be added, and to make this finite, time bounds are used. The CPTs are constructed by following the procedure described in Section 2.1.

Step (b) to (c): The Most Probable Explanation is a state assignment for all variables that is the most likely out of all possible assignments. This is a well known problem and many algorithms have been developed to efficiently find solutions.

Step (c) to (d): The pruning step selects all the instantiations of formulae that are true given the states of the predicates identified in the MPE. Conceptually speaking, we could iterate through each family of nodes in the BN, and try to instantiate the assignment for the family in each formulae. If

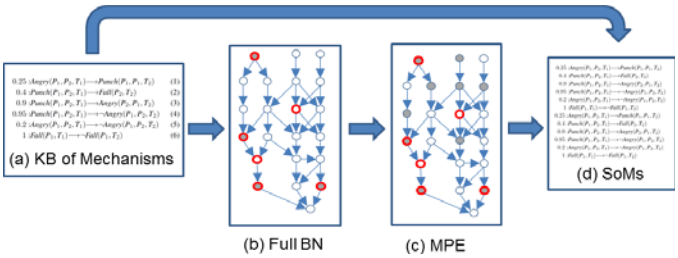

Figure 4: A high-level flow of our SoM algorithm.

all the variables in the formula are bound, we can assess the truth value of the formula (each predicate in the formula is either true or false because of the MPE step). If the formula is true it will be part of the causal explanation/prediction. If the formula is false or not all variables are bound it will not be part of the causal explanation/prediction.

In the next section we show that our algorithm produces plausible causal explanations/predictions of simulated observations generated from a causal model. Although an explanation/prediction found by our algorithm consists of a set of mechanisms instead of just a set of states, which is an improvement over existing algorithms, it always includes all the mechanisms that are consistent with the MPE states. It is thus possible that a subset of these mechanisms constitute an even better explanation in terms of Occam's Razor; some formulae may be removed from the explanation/prediction leading to fewer parameters in the model and retaining (almost) all the explanatory power. Thus there may be room for improvement of our first SoM algorithm.

4 Experiments

In this section we present an evaluation of our algorithm. The main idea is to start out with a full (ground-truth) causal explanation, and present parts of this explanation as evidence to our algorithm to fill in the gaps. More specifically, we constructed a causal model from which we generated a set of SoMs. For each of the SoMs we then selected a set of predicates that were presented to our causal explanation algorithm to recover the original SoMs. The reconstructed explanations were evaluated by using the precision-recall curve (PR-curve) as a measure of performance.

We examined the algorithm's performance on two levels: 1) recovering exact matches, requiring all the recovered formulae to exactly match the formulae in the original SoMs, and 2) time-invariant matches, where errors are allowed in the time variable, i.e., having the recovered formula occur earlier or later than in the correct SoMs.

4.1 The Airport Model

We evaluated our algorithm's performance using the *Airport* model. We used this model to describe several events at an airport, such as collisions, explosions, and terrorist threats. We model aircraft (A), vehicles (V), and time (T). We defined a set of formulae that link several events together to form SoMs, e.g., an aircraft colliding with a tug vehicle might cause a fuel leak to occur, possibly leading to an explosion.

Here are the main formulae:

$$0.01 : SameLocation(A_1, V_1, T_1) \longrightarrow Collision(A_1, V_1, T_2)$$
$$0.1 : Collision(A_1, V_1, T_1) \longrightarrow FuelLeak(T_2)$$
$$0.05 : FuelLeak(T_1) \longrightarrow Explosion(T_2)$$
$$0.2 : MaintenanceLapse(A_1, T_1) \longrightarrow MechDefect(A_1, T_2)$$
$$0.005 : MechDefect(A_1, T_1) \longrightarrow Explosion(T_2)$$
$$0.01 : Terrorist(T_1) \longrightarrow Threat(T_2)$$
$$0.01 : Terrorist(T_1) \longrightarrow Bomb(T_2)$$
$$0.95 : Bomb(T_1) \longrightarrow Explosion(T_2)$$

4.2 Methodology

All experiments were performed with SMILE, a Bayesian inference engine developed at the Decision Systems Laboratory and available at http://genie.sis.pitt.edu/. The simulation was written in Python. To evaluate the performance of our algorithm we used the following procedure:

1. Use the airport model to generate 1000 SoMs.
2. For each SoMs select a subset of predicates to present to the algorithm. These included *Explosion*, *Threat*, *SameLocation*, and *MaintenanceLapse*. The original SoMs were stored for evaluation.
3. Run the algorithm on the samples.
4. Calculate the PR-curve using the original SoMs and the SoMs constructed by the algorithm. SoMs are compared up until the first explosion in the orginal SoMs. We calculate 2 types of Precision/Recall scores: 1) *Exact Matches*: formulae in the recovered SoMs have to exactly match the original SoMs, and 2) *Time-Invariant Matches*: formulae from recovered SoMs are allowed to occur earlier or later than in the original SoMs.

The predicates that were available for selection as starting predicate were *SameLocation*, *MaintenanceLapse*, and *Terrorist*. We ran our evaluation procedure three times, varying the number of starting predicates from the set $[1, 2, 3]$ at each run.

4.3 Results

Figure 5 shows the precision-recall results for the three runs and for the two levels of comparison of our experiment. In some cases processed samples would result in identical precision-recall pairs. In our figure, a larger dot size corresponds to a larger number of samples that have the exact same precision-recall outcome.

We found that in any of the examined cases (number of starting predicates vs. Exact Match/Time-Invariant Match) the algorithm was able to achieve high Precision/Recall scores for the majority of the samples, scoring extremely well with Time-Invariant matching and very well with Exact matching, despite its increased difficulty. In cases where our algorithm did poorly, visual inspection showed that little or no informative evidence was actually presented to the algorithm. In those cases the algorithm picked the mechanism with the highest prior probability, as one would expect.

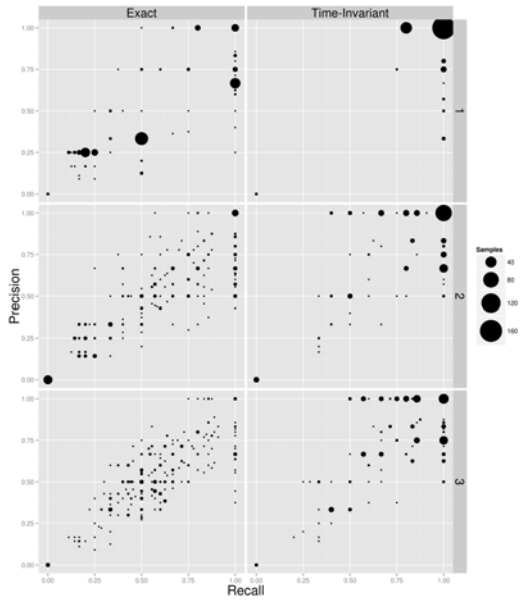

Figure 5: Precision-Recall for Exact Matches (left) and Time-Invariant matches (right) for 1,2, or 3 starting predicates (top, middle, and bottom).

5 Conclusions and Future Work

In this paper we presented a new approach to token-level causal reasoning that we call *Sequences Of Mechanisms (SoMs)*, which models causal interactions not as sequences of states of variables causing one another, but rather as a dynamic sequence of active mechanisms that chain together to propagate causal influence through time. This has the advantage of finding explanations that only contain mechanisms that are responsible for an outcome, instead of just knowing a set of variables that constitute many mechanisms that all could be responsible, which is the case for Bayesian networks. We presented an algorithm to discover SoMs, which takes as input a knowledge base of mechanisms and a set of observations, and it hypothesizes which mechanisms are active at what time. We showed empirically that our algorithm produces plausible causal explanations of simulated observations generated from a causal model.

We showed several insights about token causality and SoMs: Performing manipulations on SoM hypotheses leads to qualitatively different results than those obtained by the Do operator, possibly causing sweeping changes to the downstream causal mechanisms that become activated. This vivid change in structure made evident the fact that in general manipulation and observation do not commute, and we related this fact to the EMC condition. While only a first step, we hope this work will begin to bridge the gap between causality research and more applied AI such as that for robotics.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1260970 and the Intel Science and Technology Center on Embedded Computing.

References

[Dash(2005)] 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.

[Davidson(1967)] D. Davidson. Causal relations. *The Journal of Philosophy*, 64(21):691–703, 1967.

[Eells(1991)] Ellery Eells. *Probabilistic Causality*. CUP, 1991.

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

[Good(1961)] I.J. Good. A causal calculus. *British Journal of Philosophy of Science*, pages 305–318, 1961.

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

[Halpern and Pearl(2001)] Joseph Halpern and Judea Pearl. Causes and explanations: A structural-model approach — part 1: Causes. In *Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 194–202, San Francisco, CA, 2001. Morgan Kaufmann.

[Halpern and Pearl(2005)] Joseph Y. Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part ii: Explanations. *The British Journal for the Philosophy of Science*, 56(4):889–911, December 2005.

[Hitchcock(1995)] ChristopherRead Hitchcock. The mishap at reichenbach fall: Singular vs. general causation. *Philosophical Studies*, 78:257–291, 1995.

[Kleinberg(2012)] Samantha Kleinberg. *Causality, Probability, and Time*. 2012.

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

[Peng and Reggia(1986)] Yun Peng and James A. Reggia. Plausibility of Diagnostic Hypotheses: The Nature of Simplicity. In *Proceedings of the 5th National Conference on AI (AAAI-86)*, pages 140–145, 1986.

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

[Spirtes et al.(2000)Spirtes, Glymour, and Scheines] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, Cambridge, MA, second edition, 2000.

[Strotz and Wold(1960)] 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.

[Vennekens et al.(2009)Vennekens, Denecker, and Bruynooghe] Joost Vennekens, Marc Denecker, and Maurice Bruynooghe. Cp-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming*, 9(3): 245–308, 2009.

[Vennekens et al.(2010)Vennekens, Bruynooghe, and Denecker] Joost Vennekens, Maurice Bruynooghe, and Marc Denecker. Embracing events in causal modelling: Interventions and counterfactuals in cp-logic. In *The 12th European Conference on Logics in Artificial Intelligence (JELIA-2010*, pages 313–325, 2010.