Causal Reasoning with Autonomous Systems and Intelligent Machine Applications

Dr. Rusty Baldwin
Dr. Harold J. Arata III

ABSTRACT

In the field of Artificial Intelligence (AI), Machine Learning (ML) techniques and algorithms have been employed in a wide variety of domains and have demonstrated incredible capabilities as well as continued applicability to an ever-expanding number of areas and applications. Image and speech recognition, medical diagnosis, classification and prediction, information extraction (i.e., deep learning), commercial market and customer analysis, robotics, and self-driving vehicles are a few of the many areas where ML has either made possible or had a significant impact. Yet for all this progress, the field of AI has not yet approached what many consider the holy grail of AI: machines with human-like intelligence. Causal analysis is essential for realizing the vision of human-like reasoning: it brings the ability to determine cause-effect relationships and provides a basis for reasoning about interventions (i.e., doing), as well as what might have happened had events occurred differently (i.e., imagining/retrospection) which are fundamental characteristics of human reasoning. Causal analysis has seen widespread use and success in epidemiology, social science, and other fields for decades. Even so, its use in engineering, computer science, and AI has been limited and its potential is just beginning to be widely recognized and applied.

Dr. Rusty Baldwin is a Distinguished Research Professor in the Computer Science Department of the University of Dayton. He received a B.S. in Electrical Engineering (cum laude) from New Mexico State University in 1987, an M.S. in Computer Engineering from the Air Force Institute of Technology in 1992, and a Ph.D. in Electrical Engineering from Virginia Polytechnic Institute and State University in 1999. He served 23 years in the United States Air Force and retired in 2004. He is a registered Professional Engineer in Ohio and CISSP. He is a member of the Fellowship of Catholic Scholars, Eta Kappa Nu, Tau Beta Pi, and the Association for Computing Machinery. He has published over 100 journal and conference papers in various areas and is co-inventor on 3 patents and 7 USAF inventions. His research interests include resilient and trusted systems, autonomous systems, cyber security, computer communication networks, embedded and wireless networking, and side channel analysis.

INTRODUCTION

In the field of Artificial Intelligence (AI), Machine Learning (ML) techniques and algorithms have been employed in a wide variety of domains and have demonstrated incredible capabilities as well as continued applicability to an ever-expanding number of areas and applications. Image and speech recognition, medical diagnosis, classification and prediction, information extraction (i.e., deep learning), commercial market and customer analysis, robotics, and self-driving vehicles are a few of the areas that ML has either made possible or has had a significant impact on. The success of ML is indisputable and will continue to be an important technology for the foreseeable future.

Two grainy film shots taken at Bell Laboratories in 1952, highlight mathematician and Bell Labs researcher Dr. Claude Shannon's own construction of a robotic, maze-solving mouse known as Theseus, one of the world's first examples of machine learning (Figure 1).

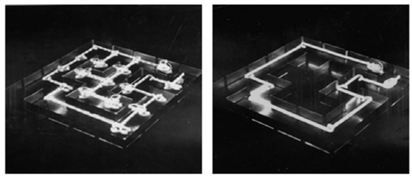
The Theseus of ancient Greek mythology navigated a minotaur's labyrinth and escaped by following a thread given to him by Mino's daughter, Ariadne, which he had used to mark his path. But Shannon's electromechanical marvel was able to "remember" its path with the help of telephone relay switches.[1]

Shannon's wheeled mouse methodically explored its surroundings—a 25-square maze. Shannon tells viewers that the maze's metal walls can be freely rearranged, so Theseus must use a small computing machine to learn the layout anew each time. But the mouse, a tiny wooden device containing a bar magnet and adorned with wire whiskers, is far too small to contain a computing machine. Instead, the machinery is hidden beneath the floor of the maze, a series of telephone relay circuits he has repurposed to do something that they had never done before: learn.[1]

Theseus was also ahead of its time, and "inspired the whole field of AI," says Dr. Mazin Gilbert, who

Dr. Harold J. Arata III, CISSP, is the Lead Systems Design Engineer at AT&T, supporting AT&T National Security and Defense. He has earned six university degrees, including a Ph.D. in computer science from the University of Tulsa. Most recently, Dr. Arata served as an Executive for Cybersecurity Strategy at Hewlett Packard Enterprise (HPE). Preceding his career in industry, Dr. Arata was the Director of the Air Force Cyberspace Technical Center of Excellence, Air Force Institute of Technology (AFIT), where he educated 650 joint cyber professionals a year. Dr. Arata also served as a Senior Military Professor at AFIT, conducting defense-focused research at the Master's and Ph.D. levels. Preceding Dr. Arata's federal civil service, he was an active duty, 2-year, below-the-zone, select to Full Colonel. Dr. Arata's military awards include being individually designated Best-in-Air Force as the Lt. General Leo Marquez Communications-Electronics award winner and being a recipient of the Legion of Merit.

was the Vice President of Advanced Technology with AT&T Laboratories. The mouse, who was featured in *Popular Science*, *Time*, and *Life* magazines the same year the film was made, learned purely through trial and error. Dr. Gilbert explained that "this random trial and error is the foundation of artificial intelligence."[1]



These photos, published in Life magazine in 1952, show the path Theseus took while learning a maze pattern and the direct path taken on its second trip through the same maze.

Figure 1: Theseus in Action[1]

Although there has been much progress made in AI since Theseus, the field of AI has not yet approached what many consider the holy grail: machines with human-like intelligence. In the 1950's, Alan Turing developed what became known as the "Turing Test" to determine if a machine had achieved intelligence. If an evaluator cannot tell whether they are interacting with a human or a machine over a text-only channel, the machine is said to have passed the test.[2] Whether the Turing Test is sufficient to demonstrate human-like intelligence has long been debated. What is not debatable is that systems exhibiting some level of intelligent behavior as well as the ability to learn complex and increasingly sophisticated tasks have been developed for decades. However, many feel this progress has plateaued and has failed to reach human-like intelligence.

A prominent voice in the AI community and the developer of Bayesian networks, Judea Pearl,[3] maintains that the ability to determine and reason about causality (i.e., cause and effect) is fundamental to human intelligence because it allows one to answer the question, "why?" Current ML techniques and algorithms cannot reach this level of inquiry because they are largely based on discovering associations in data (i.e., correlation) based on the passive observation of a system or post-hoc data analysis. This approach limits what can be achieved; it cannot determine cause-effect relationships because, fundamentally, ML algorithms use what statisticians call observational data. Observational data, except when carefully collected via randomized controlled experiments, cannot be used to uncover cause-effect relationships. What Pearl has dubbed the New Science of Cause and Effect[4] or causal analysis, is the ability of AI to determine cause-effect relationships from observational data under modest conditions in which actual systems operate. Furthermore, causal analysis provides the basis for reasoning about the effect of changing aspects of system operation without actually doing it (i.e., interventions), as well as reasoning about what might have happened had events occurred differently (i.e., imagining/retrospection). Causal analysis has had widespread use and success in epidemiology, social science, and other fields for over a decade.[4][5][6][7]

Its use in engineering, computer science, and AI, however, has been limited and its potential is just beginning to be widely recognized and applied. The background section that follows briefly discusses why typical inference systems (e.g., those using 1st/2nd order logic or constraint satisfaction) and data analysis alone is insufficient to determine causal relationships.

Background

Predicate and propositional logic has long been used to allow AI to reason about various combinations of propositions and the relations between them, as well as to determine whether a logical formula is true over a particular logical element or range of elements in the domain under consideration. This and other higher order logic systems constitute a fundamental basis for inference in computer science, mathematics, and other areas of science. They are essential and irreplaceable. Nevertheless, they do not provide a sufficient foundation for reasoning about cause and effect.

Consider as a simple example a naïve application of the chain rule which infers a conclusion from a set of implications. The chain rule for two implications can be shown symbolically as: $A \rightarrow B, B \rightarrow C \therefore A \rightarrow C$ or if A then B, if B then C, therefore if A then C. Though the conclusion is valid and the propositions are true, this type of reasoning fails to correctly assess causality when applied to ordinary everyday situations which even a child would be able to assess correctly. For when the symbols are said to represent actual objects the results can be nonsensical. For instance, let $A \rightarrow B$ be, "If we break the bottle, the grass will get wet." Let $B \rightarrow C$ be, "If the grass is wet, then it rained." An application of the chain rule would then produce $A \rightarrow C$, "If we break the bottle, then it rained." While simplistic, this example illustrates a fundamental limitation of many logic systems that are restricted to manipulating symbols. Causal or common-sense relationships between propositions cannot be specified because all propositions, as propositions, are interchangeable. This equivalence of propositions is what gives these kind of logic systems wide applicability but simultaneously limits their usefulness in causal analysis.

Consider another reasoning approach that had its genesis in AI and operations research: constraint satisfaction. Constraint satisfaction finds feasible solutions to achieve specified goal(s) under a given set of constraints while considering the capabilities of the agent(s) and the problem domain. The following example,[8] illustrates this approach. Suppose we have a suitcase with two locks: one on the left and the other on the right. The state of the suitcase, open or closed, depends on the position of the locks as shown in Table 1. If both locks are open the suitcase will open (#1), otherwise the suitcase will remain closed (#2-4). The constraint to be satisfied is the suitcase remaining closed. Consider the case where the suitcase is in state #2, the left lock is closed and the right lock is open. A query submitted to the constraint satisfaction system asks, "What would happen if the left lock were also opened?" This is a causal question and should result in the answer that the suitcase would open (#1). However, the response received from the constraint satisfaction inference engine was, "The right lock might get closed." Clearly an incorrect assessment of what should result! The reason for this is such systems are designed to ensure the specified constraint(s) are maintained, not to assess common sense causal effects.

Table 1: Suitcase State

	Left Lock	Right Lock	Suitcase
1.	Open	Open	Open
2.	Closed	Open	Closed
3.	Open	Closed	Closed
4.	Closed	Closed	Closed

Finally, consider a system from which we can observe/collect binary information from five entities labelled A through E that constitute the system as shown in Table 2.[4] The goal is to determine whether there is a causal relationship between entities A and E. That is, does A *cause* E? Clearly A is correlated with E (and with B, C, and D). In fact, all the pairwise entities are correlated. Equally clear is that additional data (given they remain all 0's or 1's) will not help clarify the situation. Causality cannot be determined in this situation because, as every Statistics 101 student learns, correlation does not *necessarily* imply causation. The fact remains, though, that the converse *is* true. Causation necessarily implies correlation. Human reasoning exploits this fact in the quest for knowledge and in search for the answer to the question, "why?"

Table 2: Binary System Data

		Entities		
A	B	C	D	E
1	1	1	1	1
0	0	0	0	0
1	1	1	1	1
1	1	1	1	1
0	0	0	0	0
0	0	0	0	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Vision and Objectives

The vision described here is a lofty one: to enable human-like reasoning (i.e., cognition/common sense) in Autonomous Systems (AS) and Intelligent Machines (IM). Achieving this vision will require the ability to make causal inferences and engage in causal reasoning in near real-time. The objective of this article is to take the next logical step towards enabling this vision by developing a Causal Reasoning Framework (CRF) that will provide the foundational framework and capability for causal reasoning.

Causal Reasoning

Causal reasoning is not complicated. Causal reasoning begins implicitly or explicitly every time the question, “why?” is asked. People want to know the cause of what happens; they inherently want to understand reality. “Why did this person die from lung cancer and that person live?”; “Why was this product profitable and that product a failure?”; “Why didn’t the firewall protect the network?”; “Why do rocks drop down instead of up?” Because causal reasoning is the ordinary method of inquiry for human beings, we typically do not even think about it.

Reasoning is more formal in the fields of science and engineering, but the end goal is the same: to answer the question, “why?” The typical approach is to systematically sample a population or system of interest, ***P***, and analyze the sample data as depicted in Figure 2. Statistical inference based on this sample data allows conclusions to be drawn about properties of ***P*** being measured, ***Q(P)*** at some level of confidence. This statistical inference process is sample data-centric.

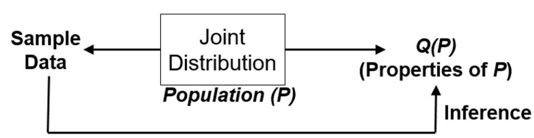


Figure 2: Statistical Inference[9]

As shown in Figure 3, the focus in causal inference shifts from ***P*** to the causal model ***M***, where the joint distribution of the data that comprises ***P*** is generated. That is, the goal of causal inference (as distinct from statistical inference) is to discover the causal model ***M*** that produced ***P.*** As Figure 3 indicates, sample data still plays a critical role in that it is used to make inferences about the properties, ***Q(M),*** of ***M***. But as will be shown, with ***M*** one can now reason about the effect on ***P*** of interventions (i.e., the effect of changing ***M***) or given that ***M*** is known or has been discovered, one can reason about what would have happened if ***M*** had been different even in the absence of any sample data from ***P***. Causal models are not data models, they are reality models.[4]

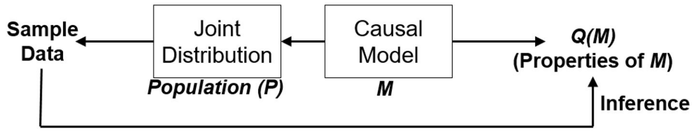


Figure 3: Causal Inference[9]

Pearl has developed a model of human reasoning that he calls the Ladder of Causation.[4] At the bottom rung of the ladder is *Association* whereby the probability of observing y given x was observed or $P(y|x)$ is ascertained. This corresponds to the human activity of observation. These probabilities are ascertained via data collection (i.e., passive observation). Except for the specialized case of randomized controlled experiments which are specifically engineered to uncover causal relationships, almost all of ML uses passive observation to produce its results by calculating conditional probabilities of an event at a given level of confidence. Typical questions that can be answered at this rung of the ladder include: "What does a symptom ***(x)*** tell me about a disease ***(y)***?" or "What does sales data ***(x)*** tell me about my customer ***(y)***?"

The next rung up the ladder is *Intervention*. This corresponds to the human activity of doing. Under intervention, the experimenter is no longer a passive observer but actively changes the data generating process ***M***. With intervention, the probability of observing y given I *do* ***x*** or $P(y|\mathbf{do}(x))$ is ascertained. The operator, $\mathbf{do}(x)$, signifies Pearl's do-calculus[3] has been applied. The **do** operator is one of the most significant results of Pearl's causal research because it enables one to use observational data to determine causal relationships under certain conditions. Previously, the main reason observational data could not be used to determine causal relationships was due to statistical confounding whereby multiple effects (possibly containing causal or merely correlated effects) were mixed together. When confounded, these effects can neither be distinguished nor separated from each other. Hence the term, "confounded." One of Pearl's main technical achievements is the development of the do-calculus, where causal effects can be determined from observational data in most situations under mild conditions.[3] Example questions that can be answered at this rung of the ladder include: "If I take aspirin ($\mathbf{do}(x)$), will my headache go away ***(y)***?" or "What would happen to the cancer rate (y) if smoking were banned ($\mathbf{do}(x)$)?"

The third and final rung of the ladder is *Counterfactuals*, which corresponds to the human activity of imagining or retrospection. On this rung, the experimenter can reason about the probability something would happen contrary to what actually occurred. Mathematically this is written as $P(y_x|x')$. "Would my headache have gone away (y_x), if I hadn't taken (x') that aspirin (x)?" or "What would the world be like (y_x) if gravity were different (x') than it is (x)?" are examples of the types of questions that can be asked at this level.

Causal reasoning is the distinguishing characteristic of human reasoning and inquiry. Many contend with Pearl that human-like reasoning in machines cannot be realized unless machines are able to operate at all three rungs of the Ladder of Causation.[4]

Causal Models

Figure 4, below, is an example of a simple causal model represented by a causal diagram. A causal diagram is nothing more than a directed acyclic graph (DAG) where the nodes are measurable outputs (i.e., variables) of a system and the directed edges indicate a causal relationship between them. Informally, the directed edges can be thought of using the metaphor "listens to." The directed edge from A to B indicates that B listens to A when determining its output value. Similarly, nodes C and D listen to B to set their output value, while E listens to both C and D.

The absence of an arrow is equally important as this indicates who a node does *not* "listen to." Thus, in Figure 4 the one edge from A to B asserts that Node B listens to A *and only* A in determining its output value. Thus, DAG models the invariant causal relationships that are either known or assumed for a given process or system. If the functional relationship between the nodes is known, this can be included in the causal analysis. The formal abstract model of Figure 4 is:

$A=f_A(U_A)$, $B=f_B(A, U_B)$,
$C=f(B,U_C)$, $D=fD(B, U_D)$, and
$E=f_E(C,D,U_E)$

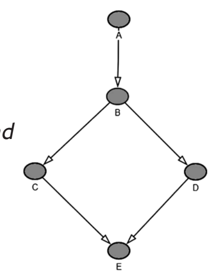


Figure 4: Causal Diagram of a Simple System[4]

where Ux is some unmeasured or unmeasurable latent variable (e.g., noise), and fx is a function that defines precisely how the node determines its output value. This function, fx, can be linear or non-linear, continuous, or discrete, parametric or non-parametric.

Even without knowledge of the actual functional relationships between nodes, the representation of who listens to whom shown in a simple DAG provides a significant amount of structural information. First, it makes explicit the known or assumed causal relationships within the system. Thus, the DAG forces an analyst to show their hand, thereby openly declaring assumptions and/or presenting their knowledge about how a system operates. Second, if the DAG accurately captures the actual causal relationships in a system, certain statistical relationships or testable implications will be reflected in the data. For example, if Figure 4 reflects the actual causal relationships of a given system then particular conditional independencies between nodes will be reflected in data collected from the system. These can be easily checked using virtually any statistical software package. In Figure 4 the following conditional independencies (i.e., $\perp$) must hold: $B \perp E \mid C, D$; $A \perp E \mid C, D$; $A \perp E \mid B$; $A \perp C \mid B$; $A \perp D \mid B$; and $C \perp D \mid B$.

Consider the first conditional independence, $B \perp E \mid C, D$. This asserts that given the values C and D are held constant, say via intervention, the variables B and E will exhibit statistical independence. If these conditional independencies do not hold, then the data and the DAG are incompatible. What follows from this is, even when the testable implications hold, that does not constitute a proof that the specified causal model is correct, but rather indicates that the specified model is not incorrect. This is akin to statisticians declaring that two systems are statistically not different. One cannot properly declare two different random variables the same because the observation period is necessarily finite. This means there may be several causal models compatible with the data. This should not be seen as a negative as it provides a ready basis for reasoning about plausible explanations for what has been observed.

Take as a concrete example the previously presented data from Table 2 and the causal diagram from Figure 4. Since edges represent causal relationships, the question that could not be answered before from the data alone, namely, does A *cause* E can now be answered affirmatively. A does cause E because E listens to D, D listens to B, and B listens to A—a chain of causality.

The causal diagram of Figure 4 is actually a causal diagram of a firing squad.[4] As shown in Figure 5, Node A represents the court which, when it takes on the value of 1, has issued an execution warrant. Node B is the Commander who without fail issues the order to fire (i.e., 1) upon receiving a warrant. The riflemen (Nodes C and D) are expert marksmen who always fire when ordered to do so by their Commander (i.e., 1) and always hit their target. Node E is the victim who dies (i.e., 1) whenever either (or both) C and D fire. The triangular symbol in Node A indicates it is the exposure variable while the "I" in Node E indicates it is the outcome variable. This graphically depicts the question, "Does A cause E?" or, "Is the Court issuing the warrant causally related to the death of the victim?" Of course, the answer is yes. But this was impossible to ascertain from the data in Table 2 without knowing the data generating process (i.e., $\boldsymbol{M}$).

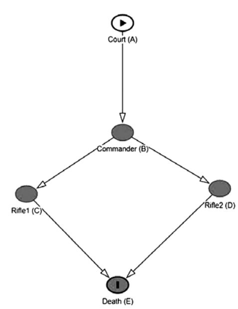


Figure 5: Causal Diagram of a Firing Squad[4]

The data in Table 2 reflects the firing squad operating as intended. However, now that the data generating process $\boldsymbol{M}$ is known, questions from rungs higher in the Ladder of Causation can be answered that before could not be determined using only the data collected from the system. For example, it was previously determined that A causes E. But what would the value of E be if, due to an intervention, C was set to 1? This situation is reflected in Figure 6. Note that C no longer listens to B; the arrow has been removed. The question being asked is essentially: What is the value of E if C is 1, independent of A, B, and D? The answer is, no matter what the other node values are, E will be 1. The rifleman is an expert and never misses. We can conclude this outcome even though the following combination of node values has not been observed (i.e., A-E being 0 0 1 0 1, respectively). In fact, if X represents "don't care" it can be concluded that if C = 1, E = 1 in a total of 7 situations that have not been observed (i.e., A-E being X X 1 X 1, respectively).

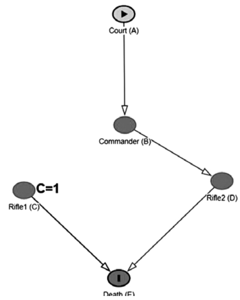


Figure 6: Firing Squad Intervention[4]

Moving to the final rung of the ladder, Counterfactuals, one can use ***M*** to analyze the situation where the firing squad operated as intended (i.e., A-E was 1 1 1 1 1, respectively) and imagine whether the outcome, E, would have been different if C had been 0 (i.e., A-E was 1 1 0 1, respectively). Given the causal model, ***M***, the answer is the outcome would have been the same. Node E would still be 1.

That this is completely obvious and even trivial is the fact that proves the point. For it is manifestly NOT obvious or trivial to a machine or algorithm that only has access to the data in Table 2. Furthermore, no amount of additional data (from a correctly operating system) would have helped. With a causal model though, reasoning about interventions and counterfactuals is readily accomplished.

An additional benefit of causal models in the form of DAGs is the ability to discover analogous situations across disparate domains. This situation is depicted in Figure 7, below. The causal model for the firing squad in the left-most section of the figure is from the legal/law enforcement domain but it describes a similar system from the aerospace domain in the middle of the figure. The aerospace system is a landing gear deployment system where an Aircraft Commander (A) initiates landing gear deployment by authorizing the uplock hook command via a relay (B) which signals two hydralic actuators (C and D) to lower the landing gear (E). Even more generally, the analogy extends to a dual-redundant command and control system from the even broader domain of system reliability as shown in the right-most causal model of Figure 7. Thus, this demonstrates how a system that works in one domain can potentially (and perhaps drastically) reduce the learning curve required to understand systems in a related domain. Given the readily available algorithms to quantify graph similarity, the DAG becomes an even more attractive representation of causality.

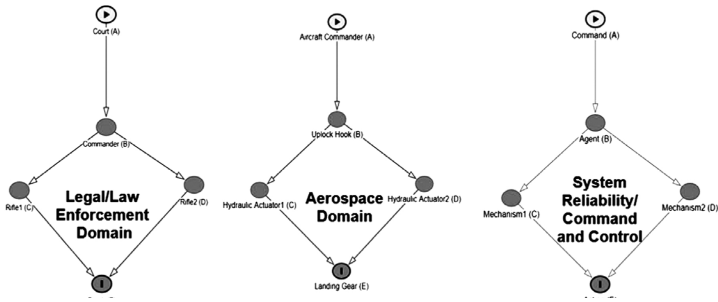


Figure 7: Analogies from Causal Models

Causal Reasoning in a Sports Medicine Scenario

A more complex and realistic example[5] comes from the field of epidemiology.[6][10] The causal model in Figure 8, below reflects a research team's consensus on causal factors related to participant injuries during a sports game. It is not directly based on data, but rather on their collective experience. The question considered is: Are Warm-Up Exercises (WUE) a causal factor of Injury (I)? and, is indicated by the light-gray arrows. Data for each one of these variables was collected and the testable implications were analyzed to verify compatibility between the data and the causal model using the statistical software, R, with the package daggity. Figure 9, below, shows the R program. In this case, analysis revealed the causal model and the data were indeed compatible; all required conditional independencies were reflected in the data.

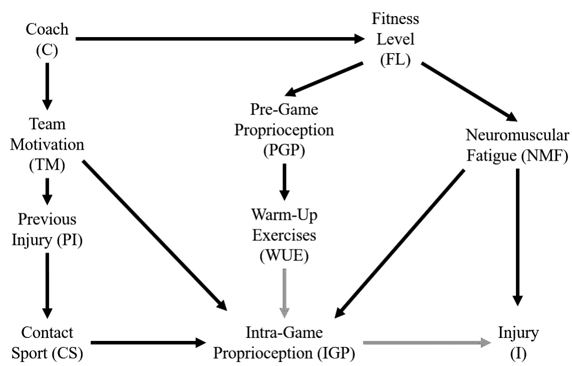


Figure 8: Sports Injury Causal Model

```
# load the R package `dagitty`
library(dagitty)
# load data from a text file
d <- read.csv("http://dagitty.net/sports.csv")
# download DAG from dagitty.net
g <- downloadGraph("dagitty.net/mN4IKjR")
# evaluate the d-separation implications of the DAG
r <- localTests(g, d)
# perform Holm-Bonferroni correction
r$p.value <- p.adjust(r$p.value)
# focus on tests with p-values below a threshold
r <- r[r$p.value<0.05,]
# plot results
plotLocalTestResults(r)
```

Figure 9: R "testable implications" Program[5]

Suppose, however, that the causal model was not specified correctly; suppose a causal relationship was inadvertantly omitted. This is the case in Figure 10 where the directed edge between Team Motivation (TM) and Warm-Up Exercises (WUE) has been removed. Now the resulting output from the R program of Figure 9 shown in Table 3 below indicates that in three instances the conditional independencies of the listed variables did not hold. Namely, Team Motivation (TM) should be conditionally independent of Warm-Up Exercises (WUE) given Pre-Game Proprioception (PGP), Fitness Level (FL), and Coach (C), respectively. However, the *p*-values (i.e., the values in the p.value column of Table 3) did not exceed the required threshold of 0.05 and therefore this is not the case.[1] Thus, an error in the causal model or, equivalently, a misunderstanding of causal relationships in the situation under consideration can be detected objectively and explicitly.

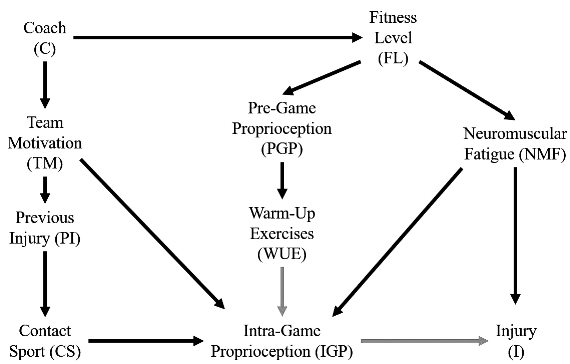


Figure 10: Sports Injury Causal Model with TM/WUE Edge Removed[5]

Table 3: R Output from "testable implications" Program

	estimate	std.error	p.value	2.5%	97.5%
TM _||_ WUE | PGP	0.2463031	0.04241572	7.273348e-07	0.1629669	0.3296393
TM _||_ WUE | FL	0.2397066	0.04189577	1.150645e-06	0.1573919	0.3220212
TM _||_ WUE | C	0.2287258	0.04109466	2.649499e-06	0.1479851	0.3094664

As a final result, estimates of the path coefficients can be determined from the causal model and the data.[2] That is, a numerical estimate of the causal effect of WUE on I can be determined. Figure 11 shows the simple R program used to calculate the coefficients, Figure 12 shows the resulting output which includes various summaries of fit indices on the left (i.e., quality metrics for the path coefficient estimates) as well as path coefficient estimates themselves on the right. The causal diagram with the path coefficients annotated is shown, below, in Figure 13.

1 The null hypothesis is that the tested variables are conditionally independent (i.e., using causal terminology, *d*-separated). Since the *p*-values are less than the specified threshold of 0.05 the null hypothesis is rejected: the variables are *not* conditionally independent.

2 Data has been scaled and normalized.

```
library(dagitty)
library(lavaan)
#load data from a text file
d <- read.csv("http://dagitty.net/sports.csv")
# CORRECTED DAG
g <- dagitty('dag {
+ C [pos="-4.000,-3.000"]
+ CS [pos="-4.000,3.000"]
+ FL [pos="1.000,-3.000"]
...
# Convert to lavaan object, fit model to data,
# Determine and display path coefficients
m=toString(g, "lavaan")
fit=sem(m, d)
summary(fit)
```

Figure 11: R Program to Calculate Path Coefficients[5]

```
lavaan 0.6-3 ended normally after 14 iterations

  Optimization method                           NLMINB
  Number of free parameters                         23

  Number of observations                           500

  Estimator                                         ML
  Model Fit Test Statistic                      22.771
  Degrees of freedom                                29
  P-value (Chi-square)                           0.787

Model test baseline model:

  Minimum Function Test Statistic              612.933
  Degrees of freedom                                44
  P-value                                        0.000

User model versus baseline model:

  Comparative Fit Index (CFI)                    1.000
  Tucker-Lewis Index (TLI)                       1.017

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -5370.189
  Loglikelihood unrestricted model (H1)      -5358.803

  Number of free parameters                         23
  Akaike (AIC)                               10786.378
  Bayesian (BIC)                             10883.314
  Sample-size adjusted Bayesian (BIC)        10810.311

Root Mean Square Error of Approximation:

  RMSEA                                          0.000
  90 Percent Confidence Interval          0.000  0.023
  P-value RMSEA <= 0.05                          1.000

Standardized Root Mean Square Residual:

  SRMR                                           0.026
```

```
Parameter Estimates:

  Information                                 Expected
  Information saturated (h1) model          Structured
  Standard Errors                             Standard

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  FL ~
    C                 0.242    0.041    5.857    0.000
  TM ~
    C                 0.189    0.043    4.378    0.000
  IGP ~
    CS                0.314    0.036    8.742    0.000
  PI ~
    CS                0.302    0.043    7.027    0.000
  NMF ~
    FL                0.346    0.045    7.739    0.000
  PGP ~
    FL                0.317    0.046    6.960    0.000
  I ~
    IGP               0.307    0.040    7.599    0.000
    NMF               0.256    0.040    6.463    0.000
  IGP ~
    NMF               0.219    0.036    6.061    0.000
  WUE ~
    PGP               0.184    0.043    4.320    0.000
  IGP ~
    TM                0.233    0.038    6.073    0.000
  PI ~
    TM                0.306    0.044    6.892    0.000
  WUE ~
    TM                0.258    0.044    5.824    0.000
  IGP ~
    WUE               0.272    0.037    7.371    0.000
```

Figure 12: R Path Coefficients Program Output

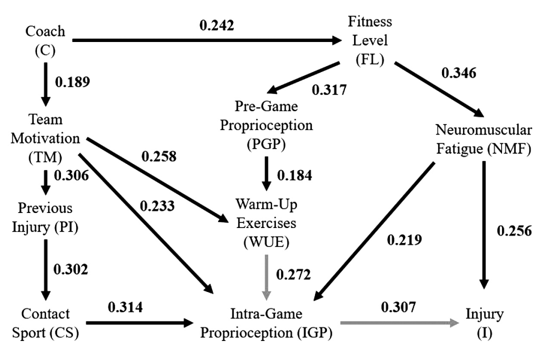


Figure 13: Sports Injury Causal Model with Path Coefficients

Using the path coefficients, the causal effect is easily determined. The effect of WUE on I, i.e., $f(WUE, IGP)$, is simply the product of the path coefficients on the light-gray causal path between WUE and I. That is $I = f(WUE, IGP) = 0.272 \cdot 0.307\ WUE = 0.0835\ WUE$. The significance of achieving a causal result, to say nothing of a numerical causal result, from observational data is considered astounding and even unbelievable or impossible by many statisticians. Nevertheless, this type of analysis is routine in the fields of biology, medicine, and social sciences and has been for decades.[4] [5] [6] [7]

Autonomous Systems and Intelligent Machine Applications

There are several fundamental areas where causal analysis can be used to advance AS and IM cognitive capabilities. Fortunately, much of the software needed to perform the critical manipulation, analysis, and testing of causal models and other data structures has already been implemented in statistical packages like daggity and lavaan from the R statistical software suite and are available via Application Programming Interface (API) calls from any number of languages. They are also readily available in equivalent statistical and structural equation modeling software suites. Thus, the required foundational computational and statistical tools are in place, mature, and ready to be used in the development of the capabilities described below. Some ideas of AS and IM applications include the following:

❖ ***Knowledge Storage and Retrieval***

This capability is fundamental to many, if not most, areas within AS and IM. Since the DAG serves as the core data structure for causal information and stores fundamental causal knowledge, the rich set of graph theory algorithms to analyze and characterize DAGs can be brought to bear. Furthermore, graphs, especially sparse graphs, can be very efficiently stored, retrieved, and compared. Some knowledge storage and retrieval capabilities include the following:

- Searching for similar causal models or those models similar (as measured by graph similarity metrics) to the situation reflected in the observed (and possibly real-time) data.
- This capability can additionally serve as a basis for discovering analogies by analyzing/comparing causal models that meet some minimum similarity threshold.

❖ ***Learning***

- Given two approaches to accomplish a task, evaluate multiple virtual “what if” scenarios to discover a more efficient or effective approach (i.e., different causal paths that achieve the same effect). Routines to simulate the operation of a causal model are readily available, relatively efficient, and fast.

❖ ***Discovery***

- Build causal model(s) of the operating environment via observations alone. Employ human experts to refine the models with feedback from the AS/IM as to whether the suggested refinements are compatible with the observed data.

- Similarly, develop an IM to observe the environment via sensors/other instruments and propose causal explanations for the collected data. That is, the IM will provide plausible explanations via causal diagrams that are readily interpretable by humans.

❖ ***Explainable AI***

- Given causal diagrams annotated with path coefficients that represent metrics such as cost or efficiency, an IM or AS can explain why it recommends a course of action (COA) X over Y, "It was more efficient than any alternative. Shall I list the other COAs I considered and explain why I didn't choose them?"
- By treating a causal diagram as a road map, existing routing and mapping algorithms and optimization routines can be brought to bear. Given a causal diagram annotated with an AS's or IM's ability to influence certain causal outcomes and the cost to do so (both of which may vary over time), the AS/IM can readily explain why a task was done the way it was at that particular time, "I would have accomplished the task in the preferred way, but at the time, my ability to modify this system parameter was disabled/malfunctioning."

❖ ***Experiments without (more) data***

- The question, "what would happen if I did this?" can be investigated by direct manipulation (i.e., intervention) of the causal model.
- The question, "what would happen if I had done this instead?" can be explored using the causal model to imagine alternative outcomes.
- The resulting causal models can be compared to the current/actual model of reality to evaluate alternative COAs.

❖ ***Policy evaluation***

- The question of whether a person/organization/system is conforming to a given policy can be determined by comparing the policy (i.e., a causal model of how the world "should" be) to data from the real world. If the causal model and the data are compatible, this indicates the specified policy is being followed.
- If they are incompatible, alternative models that are compatible with the data can be generated and compared to the should-be model to identify the possible areas of non-compliance.

There are many other potential AS/IM applications, but those identified above serve to demonstrate the rich and diverse areas in which causal reasoning is both applicable and can bring unique capabilities to AS's and IM's.

Causal Reasoning Framework (CRF)

The Causal Reasoning Framework (CRF), as shown in Figure 14, organizes the foundational components needed for causal inference and reasoning into a unified whole. The CRF is intended to be the basis for further experimentation and research into causal analysis and inference. To provide maximum flexibility, CRF was developed using open source, royalty-free components. The two main components of CRF are Soar[11] and R.[12] Soar will be explained in more detail below.

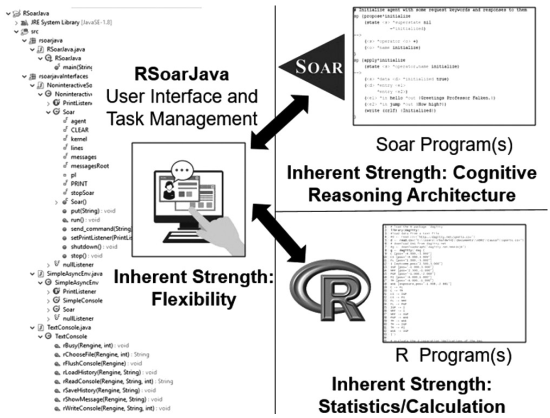


Figure 14: The Causal Reasoning Framework (CRF)

Briefly, however, Soar is an open-source cognitive architecture whose functional capabilities and architectural elements mimic the principle areas used in human cognition. Soar has been in development for over 35 years, is well-documented, and provides robust and stable computational building blocks for CRF. The inherent strength of Soar is its cognitive reasoning architecture.

R is a programming language and software environment for statistical computing. It is widely used in the area of AI, ML, causal analysis, and of course, statistics. It has a large and active user base and a core set of packages with over 15,000 additional packages available. It is supported on Windows, Linux, MacOS, and other platforms. R serves as the computational platform for Soar. The inherent strength of R is its rich statistical capabilities and robust API.

The final component of CRF is RSoarJava, which is a Java-based, "wrapper" application that is intended to provide user interface and task management functionality. It currently has a rudimentary capability to exchange data with both Soar and R via their respective APIs. RSoarJava's inherent strength is flexibility.

Soar, the cognitive architecture for CRF, is ideally suited for causal reasoning and analysis in that Soar presumes some initial state and a desired state and then applies operators to move towards the desired state. The Soar architecture includes the general capabilities and logic for automated decision-making, multiple types of learning, problem solving, and hierarchical planning. This greatly reduces the technical risk as development efforts can be directed to formulating the Soar rules, productions, and other procedures to develop causal applications rather than developing and debugging fundamental cognitive memories, capabilities, and learning mechanisms.

Figure 15 below shows the major architectural elements of Soar. Working Memory is a shared short-term representation of the current situation represented by a single, connected, directed graph. Production Memory stores knowledge about how to do things (e.g., procedures). Semantic Memory contains long-term contextual knowledge about objects or concepts as represented by disconnected graphs consisting of multiple directed sub-graphs. Episodic Memory captures temporally ordered information along with the context of when and how an episode was experienced. Reinforcement Learning (RL) can be used to guide operator selection based on a reward function, chunking captures general knowledge gained from impasse resolution. Semantic and episodic learning derives knowledge based on past experience. Functionally, the knowledge contained in Soar episodic and semantic memory is stored in a memory-based SQLite database. Soar supports all major platforms, is open source and has a domain independent API. It has bindings to many languages including C/C++, Java, Python, and TCL.

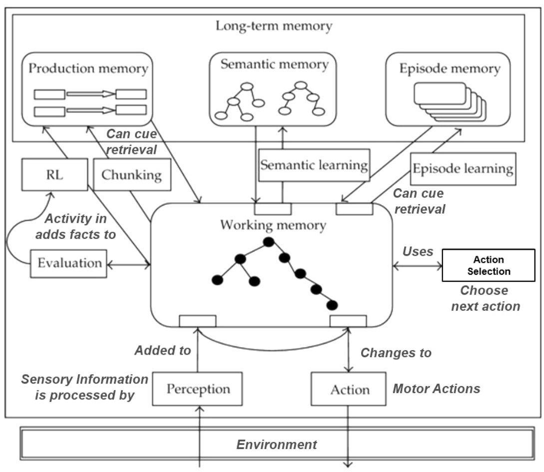


Figure 15: The Soar Cognitive Architecture[11]

In Figure 16 below, the CRF inference engine is shown. It has been adopted wholesale from Pearl's inference engine[13] and serves as the paradigm for CRF component interaction as well as for the presumed workflow resulting from a causal query, which the following description is based on.[13] The engine accepts three inputs and produces three outputs. The directed arrows in the figure indicate information flow between inference engine components. The icons in the blocks of the inference engine indicate the CRF component providing that functional capability. On the input side, the Query is presumed to be supplied by a CRF user via RSoarJava, while initial causal model(s) (i.e., Assumptions) and the Data is provided via domain experts and the domain, respectively. On the output side, Soar takes as input the causal model(s) and the query and formulates the "Estimand," $\boldsymbol{E_S}$. That is, the schema or recipe for answering the query. The Estimand, $\boldsymbol{E_S}$ and the data is used by R to calculate an Estimate or answer to the query, $\boldsymbol{\hat{E}_S}$. Fit Indices, $\boldsymbol{F}$, measure how well $\boldsymbol{\hat{E}_S}$ answers the query and is produced by R. As can be seen in the figure, these results are provided to Soar. As conceived below, Soar provides the heavy lifting with respect to acting on the causal inferences made by the inference engine to accomplish any tasks associated with the AS/IM applications.

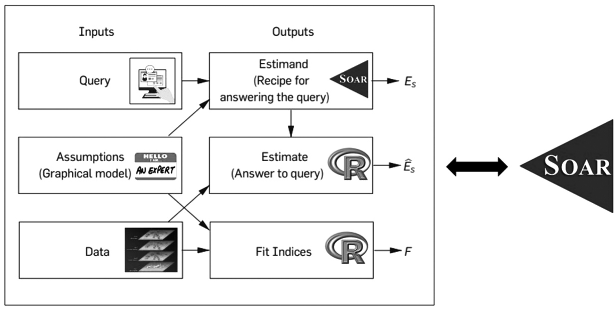


Figure 16: CRF Inference Engine based on[13]

Using Pearl's inference engine design as adopted and incorporated into the CRF, the "7 tools of Causal Inference"[13] can be realized and used to power new and unique applications for AS and IM. These tools include the following:[13]

1. Transparency and testability via encoding causal assumptions
2. Intervention and control of confounding via do-calculus
3. Answer "what if" questions via developing algorithimitization of counterfactuals,
4. Assess direct and indirect effects via causal mediation analysis
5. Robustness (i.e., adaptability, external validity, overcoming sample selection bias) via do-calculus
6. Recovery from missing data via assessing causal model(s) of the "missingness" process
7. Causal discovery via evaluation of models compatible with collected data.

Cyber Defense Strategy Observations

An increasing number of industry insiders believe more creative thinking, more research, more knowledge management and more causal reasoning with autonomous systems and intelligent machine applications—not just more technology—is needed. Dr. Thomas Homer-Dixon outlined this ingenuity gap, "in general, as the human-made and natural systems we depend upon become more complex, and as our demands on them increase, the institutions and technologies we use to manage them must become more complex too, which further boosts our need for ingenuity. The crush of information in our everyday lives is shortening our attention span, limiting the time we have to reflect."[14] It is these increasing demands, combined with today's greater network complexity, and rising social unpredictability, that make it more critical than ever that smart technical and social solutions be ready at a moment's notice. The MIT scientist Edward Lorenz's Chaos Theory is also used to describe how small changes can lead to widely varying results and path dependence.[15] As such, it is essential to leverage a new cyber situational awareness (SA) model that incorporates the aforementioned: causal reasoning with autonomous systems and intelligent machine applications.

Protecting enterprise networks and providing mission assurance without significant autonomous systems supporting cyber-SA and warning capabilities will continue to be a challenging mission. Without causal reasoning with autonomous systems, we are left with a fragmented, imperfect view into enterprise networks and how cyber assets map to tasks, objectives, and missions. This incomplete view thwarts threat detection, trend analysis, and preemptive actions which fosters slow or non-existent reactions to threats and changing conditions. An environment like this constricts a senior leader's decision-making space. Cyber-SA for most enterprises is presently disjointed, rudimentary, ad hoc, too focused on technical analysis, lacking important cyber threat intelligence data feeds from supporting providers, and missing actionable, contextual analytics provided by causal reasoning within autonomous systems. Moreover, personnel are currently delivering very limited strategic cyber-SA capabilities for senior leadership.

This flawed view can be operationally blinding to any organization. Initial progress has been made today by many organizations to increase their causal reasoning with autonomous systems capabilities to enhance their organizational cyber-SA capabilities, for example, security operations centers with advanced networks and AI algorithms. However, many organizations may further strengthen their cyber-SA and warning capabilities by weaving an empowered cyber-AI construct with causal reasoning attributes into their enabled mission assurance strategy. This construct has a high return on investment for any organization operating in today's high threat environment.

The time has arrived for a new model, more ingenuity, and the recognition of the importance of cyber-SA in defense of an organization's enterprise. What matters in transforming an organization's cyber–SA is causal reasoning with autonomous systems that increase intelligence, integration, speed, analytics, expertise, and resiliency. Enacting just such a cyber-AI causal reasoning with autonomous systems framework can and will enable an organization to more effectively protect itself today and in the future.

CONCLUSION

Causal analysis is essential for realizing the vision of human-like reasoning: it brings the ability to determine cause-effect relationships and provides a basis for reasoning about interventions (i.e., doing), as well as what might have happened had events occurred differently (i.e., imagining/retrospection) which are fundamental characteristics of human reasoning. Causal analysis has seen widespread use and success in epidemiology, social science, and other fields for decades. Even so, its use in engineering, computer science, and AI, has been limited and its potential is just beginning to be widely recognized and applied. For all the progress that has been made in the field of AI, machines with human-like intelligence are still not a reality. Like the story of Theseus and Dr. Shannon's electromechanical mouse, there is promise for those in the field of AI to find a path through the maze as well.

NOTES

1. Daniel Klein, "Mighty Mouse," MIT Technology Review, 2018, 1.
2. Robert Schalkoff, *Intelligent Systems: Principles, Paradigms, and Pragmatics*, Jones & Bartlett, 2011, 14.
3. Judea Pearl, *Causality (2nd Edition): Models, Reasoning, and Inference*, Cambridge University Press, 2009.
4. Judea Pearl, *The Book of Why: The New Science of Cause and Effect*, Basic Books, 2018.
5. Johannes Textor, et al., "Robust causal inference using directed acyclic graphs: the R package 'dagitty,'" *International Journal of Epidemiology*, 45:6, 2016, 1887-1894.
6. Mikel Aickin, *Causal Analysis in Biomedicine and Epidemiology*, Marcel Dekker, Inc., 2002.
7. Rex Kline, *Principles and Practice of Structural Equation Modeling*, Guilford, 4th Edition, 2016.
8. Fangzhen Lin, "Embracing Causality in Specifying the Indirect Effects of Actions," *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Quebec, Canada, August 1995, 1985-1991.
9. Judea Pearl, *The Mathematics of Cause and Effect: With Reflections on Machine Learning,* Microsoft Research, Video Presentation.
10. Bill Shipley, *Cause and Correlation in Biology 2nd Edition*, Cambridge University Press, 2016.
11. John Laird, *The SOAR Cognitive Architecture,* MIT Press, 2012.
12. R Core Team, *An Introduction to R*, available on-line at https://cran.r-project.org/ doc/manuals/r-release/R-intro.html, Version 3.6.2, 12 Dec 19.
13. Judea Pearl, "The Seven Tools of Causal Inference, with Reflections on Machine Learning," Communications of the ACM, March 2019, 54-60.
14. Homer Dixon, "How Can We solve the Problems of the Future?" The Ingenuity Gap, Knopf Publishing, 2000, 4.
15. Edward Lorenz, *Predictability: Does the Flap of a Butterfly's Wings in Brazil Set Off a Tornado in Texas?* Retrieved from https://mathsciencehistory.com/wp-content/uploads/2020/03/132_kap6_lorenz_artikel_the_butterfly_effect.pdf , 1972.