Cause & Effect Algorithm

When we speak of intelligence, we speak of purposefully intervening in the natural flow of events through *actions* in order to further some goal. Generally speaking, the task of intelligence is deciding whether to allow history to unfold unimpeded or whether to intervene. Additionally, if intervention is chosen, then the correct action must be selected.

This algorithm builds an intelligent system out of primitives: *states, events, actions and goal.*

1. The driver of activity in this algorithm is the goal. Simply speaking, the task of the system is to maximize an *a priori* (and external) *goal*. The goal is quantified by a goal objective function *G(t).*
2. The only way to influence the goal is through the intervention of some *action*. Without actions, the goal will vary outside the system’s control. Through actions, the system attempts to influence *G(t)* in order to maximize it.
3. For an action to favorably and reliably influence *G(t)*, it is necessary that the right action be done at the right time. The mechanism that links the right time and the right action is called an *event.* All actions have associated events that trigger them.
4. *States* are partitionings of some space – probably we need a state machine with an associated space derived from the input space of the system and/or a memory trace of past activity. Whichever way we implement it (either a state machine or some kind of state boundary adjustment), states and state transitions ultimately need to be *deterministic.* This seems intuitive because once the system hones down a way to do something, it is hard to imagine a reason to do it unreliably. It seems a better choice to keep randomness out of this part. (Maybe some highly advanced system needs to be “shifty and unpredictable” but we won’t consider it here.)

Thus the algorithm is driven in a specified order:

1. Actions are explored randomly to assess their effects on the goal.
2. Actions that have effects on the goal function are linked to events in order to trigger or prevent triggering of them. This implies that we require a state transition in these cases in order to define the event.
3. If there is no suitable transition to create/adjust an event, then the states themselves must be adjusted in order to create one.

Cause and Effect

Item #1 of the algorithm requires using the observed effects of an action on the goal function to determine if there is a *causal link* between them. We say that a causal link exists if we deem that the action can generate a sufficiently reliable prediction about the change in *G(t)*. Since the effect of an action might depend on the particular time and circumstance in which it occurred, our model needs to consider that causes produce effects with a probability that is *conditioned* on a certain time and circumstance. We identify a time and circumstance with an *event*.

Assume the system is considering action A “at” event E, the system must have knowledge concerning the outcomes of A|E and not-A|E in the form of probability distributions. This knowledge can be crudely hardwired in some cases but generally must be learned through the experience it gains gradually over time. In other words, the system must *learn.*

Since learning involves assessing the conditional probabilities above, and assessing probabilities perfectly requires infinite samples, the system can never determine the *true* probability of any outcome. Instead, it perpetually operates in the regime of having too small of a sample, leading to only an approximate probability. Any practical intelligent system must be based on the assumption that knowledge is always imperfect. From this we conclude that the choice of representation for outcomes must reflect not only the conditional probabilities of outcomes but also the *quality* of such probabilities.

Intuitively, this makes sense. The huge number of inputs and outputs of any useful system guarantees that most apparent *causal links* between events and their outcomes are mere coincidences. It is critical that the system be able to extract reliable predictions from the immense chaff of spurious coincidences.

Insert Link to Statistics Paper (to be written) **– paper is about how to derive probabilities from samples and how to associate a confidence level.**

Let us consider a general problem. We have an event E and its complement (i.e. lack of E) not-E, an action A and its complement not-A (i.e. lack of A), and an outcome score (i.e. a goal objective function) G. The goal of the system is to maximize the goal function G. In fact, G is the only guidance the system has.

If we fine-grain the system enough and arbitrarily rescale it such that there are only three possible changes in G:, it follows that in order to maximize G, the system must seek +1 outcome as often as possible and the -1 outcome as rarely as possible such that over time, the function grows as quickly as possible towards its maximum. The 0 outcome does not change G and need not be sought or avoided explicitly – doing so adds nothing. In the rest of the document below, we will refer to the ordered list of changes in G as.

Similarly, we can define a function *e(t)* that is binary and has value +1 when a certain state transition occurs that corresponds to the event E and 0 when some other transition (or none at all) . When *e(t)* is 0, we say that the event not-E has occurred. We also define a binary function *a(t)* that is +1 when the action A is triggered and 0 when the not-A action occurs.

SAVED FOR REWRITE

So, upon detection of an event E, the next step is to determine whether a particular action A should be triggered by this event. The system must attempt to *predict* whether action A will likely have a good or bad outcome with respect to some goal. In other words, it needs to decide, or *predict*, whether A, or not-A is the better choice towards furthering the system’s goal.

In order to make this prediction, the system must have knowledge concerning the outcomes of A|E and not-A|E in the form of probability distributions. This knowledge can be crudely hardwired in some cases but generally must be learned through the experience it gains gradually over time. In other words, the system must *learn.*

A general limitation of any realizable system is due the finite amount of resources available to “remember” past events in order to associate them with future effects. Intuitively, given a constant level of resources, there must be some sort of tradeoff between the precision of the timing of remembered events and the “epoch” of such events. By epoch we the maximum interval a memory can last. Not to put too fine of a point on it, basically we can choose at one extreme a memory system with highly precise timing but with a short temporal capacity or, at the other extreme, a “fuzzy” remembrance of events and/or their timing but with a long temporal capacity.

This choice of this tradeoff depends greatly on the task being addressed by a particular system. Depending on the task, an argument can be made for tradeoffs close to either extreme. For example, a fine motor control task is critically dependent on precise timing. Such a system would best be served by the precise/short combination. On the other hand, a more abstract task may very well be better served by reducing timing precision drastically down to simple ordering and storing a long history. This kind of system leans toward the fuzzy/long end of the tradeoff spectrum.

We bring this tradeoff up because it is a basic a priori “design decision” that affects even the simplest tasks. We will however, set the intricacies of this choice aside for now and focus only on precise/short systems. These systems will hopefully lead to simpler mathematics than the alternative.

the system needs to assess whether A or not-A is the better choice at this moment in time. We say that the event E or not-E is the *cause* of a change in G, if and only if E or not-E has some influence on the change of G. Note that this is true w.l.o.g., even in very complex systems with huge numbers of events and possible actions. The fact remains that *every* action needs to be assessed for suitability at the moment the event E occurs. Therefore, we will first deal with a single possible action and then consider the effects of the presence of others.

As an example, consider a system that remembers the events of the T previous time steps, additionally, let us say the events are remembered exactly and with perfect timing. As mentioned above, we would call this a precise/short system with epoch T. Let us consider an arbitrary realization of such a system with T=3:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *t* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
|  | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
|  | ? | ? | ? | +1 | 0 | +1 | 0 | 0 | 0 |

Let us start by identifying the existing changes in *g(t).* They occur at t=3 and t=5. Let us consider the change at t=3 first. Since our system is of epoch 3, there are only 3 *possible* time points that will be considered as the *possible* and *knowable* causes of the change at g(3). We use the term *knowable* to make clear that the change may have a cause we are not able to detect. If the cause is unknowable (perhaps because it beyond the epoch or even because it is something the system does not measure), then *no* algorithm will be able to recover it. In this toy problem, we will assume the case where the cause *IS* knowable which implies we must have an identifying event for the cause.

ARGH .. get his action/event thing squared away. Which one is the root? I think it should be the action? The event is the timing mechanism.

Let us consider every possibility relating to the change *g(3):*



Any one of these *could* be a *causal link* but could just as easily be an irrelevant coincidence. We need to determine if we have a causal relationship. For there to be a causal relationship we need to determine whether each of them is *necessary* and also whether it is *sufficient.*

Our definition of *necessary* is:



where *n* is +1 for *necessary(+)* , i.e. necessary for increase, and -1 for *necessary(-)* , i.e. necessary for decrease*.* The definition for *sufficient* is:



where *n* is +1 for *sufficient(+)* and -1 for *sufficient(-).* And together, *necessary* AND *sufficient* is a combination of both constraints:



Using these terms, we can categorize all events that are deterministic and knowable causes. We stress *deterministic* at this point because we have not considered probabilistic aspects yet and we stress *knowable* because for a cause to be knowable it must be detectable and by extension must have an associated event:

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Necessity | Sufficiency | Comment |
|  | Effect cannot happen without this cause | Nothing else is needed | Cause is the *only* possible cause of an effect and that its presence alone guarantees the effect |
|  | Effect cannot happen without this cause | Something else is needed | Cause is a required cause of an effect and but at least one other event is also necessary |
|  | Effect can happen without this cause | Nothing else is needed if this cause is present | Event is one of more possible causes but when present it is sufficient to guarantee the effect. |
|  | Effect can happen without this cause | Something else is needed | Event is not a required cause of an effect but it can be in conjunction with other causes. |
| Null | N/A | N/A | Event is never a cause of the effect. |

1. **:** A cause that is *necessary* and t.
2. : A cause that is *not* necessary (because the effect can happen without it) nevertheless is *sufficient* to guarantee the effect.