Causal vs. Temporal Learning

Bo Morgan
MIT Media Lab

September 26, 2011

1 The Planning Task

[To do: Make this definition of the planning task allow for state dependent effects of operators.]

Definition 1. *A planning task is a 4-tuple* $\Pi = \langle V, O, I, G \rangle$*, where*

- V *is a finite set of* propositional state variables *(also called* propositions *or* facts*),*
- O *is a finite set of* operators*, each with associated* preconditions $pre(o) \in V$*,* add effects $add(o) \in V$ *and* delete effects $del(o) \in V$*,*
- $I \in V$ *is the* initial state*, and*
- $G \in V$ *is the set of* goals*.*

A *state* is a subset of facts, $s \in V$, representing the propositions which are currently true. *Applying* an operator o in s results in state $(s \setminus del(o)) \cup add(o)$, which we denote as $s[o]$. The notation is only defined if o is *applicable* in s, i.e., if $pre(o) \in s$. Applying a sequence $o_1, \ldots, o_{n+1}$ of operators to a state is defined inductively as $s[\epsilon] := s$ and $s[o_1, \ldots, o_{n+1}] := (s[o_1, \ldots, o_n])[o_{n+1}]$. A plan for a state s (*s-plan* or *plan* when s is clear from context) is an operator sequence π such that $s[\pi]$ is defined and satisfies all goals (i.e., $G \in s[\pi]$).

2 Learning the Effects of Planning Operators

We would like our system to be able to adapt to novel environments. Toward this end, we allow for $s[o]$ and $pre(o)$ to be learned during the run-time of the system.

3 Concept Learning and Hypothesis Spaces

If some instance x satisfies all the constains of hypothesis h, then h classifies x as a positive example ($h(x) = 1$). The set of items over which the concept is defined is called the set of *instances*, which we denote by X. The concept or function to be learned is called the *target concept*, which we denote by c. In general, c can be any boolean-valued function defined over the instances X; that is, $c : X \rightarrow \{0, 1\}$. When learning the target concept, the learner is presented a set of *training examples*, each consisting of an instance x from X, along with its target concept value $c(x)$. Instances for which $c(x) = 1$ are called *positive examples*, or members of the target concept. Instances for which $c(x) = 0$ are called *negative examples*, or nonmembers of the target concept. We use the symbol D to denote the set of available training examples.

Definition 2. *Let h_j and h_k be boolean-valued functions defined over X. Then h_j is* more general than or equal to *h_k (written $h_j \geq_g h_k$) if and only if*

$$(\forall x \in X)[(h_k(x) = 1) \rightarrow (h_j(x) = 1)] \tag{1}$$

4 Learning to make planning decisions

Learning to make planning decisions is an example of a delayed learning task, where the category to be learned is not available at the same time that the features are available. For example, plans that are successful in accomplishing types of goals will also tend to be developed through similar types of planning. Heuristics can be learned to predict actual success or failure from features of the planning space. Then, we use these heuristics to guide the planning search, less completely toward a static heuristic of plan distance and more toward a learned heuristic based on types of execution successes and failures.

4.1 The problem with pyramids

In our block building toy problem we have cubic blocks and pyramidal blocks. The gripper is able to pick up cubic blocks, while pyramidal blocks are impossible for the gripper to pick up. The gripper has two sources of knowledge for accomplishing the given goal. Knowledge is a causal relationship between states, or in other words, an action trans-frame. These two knowledge sources have features that can be used for training our heuristics.

The block building domain is shown in Figure 1, where four panels (a, b, c, d) illustrate a scenario of learning to make planning decisions.

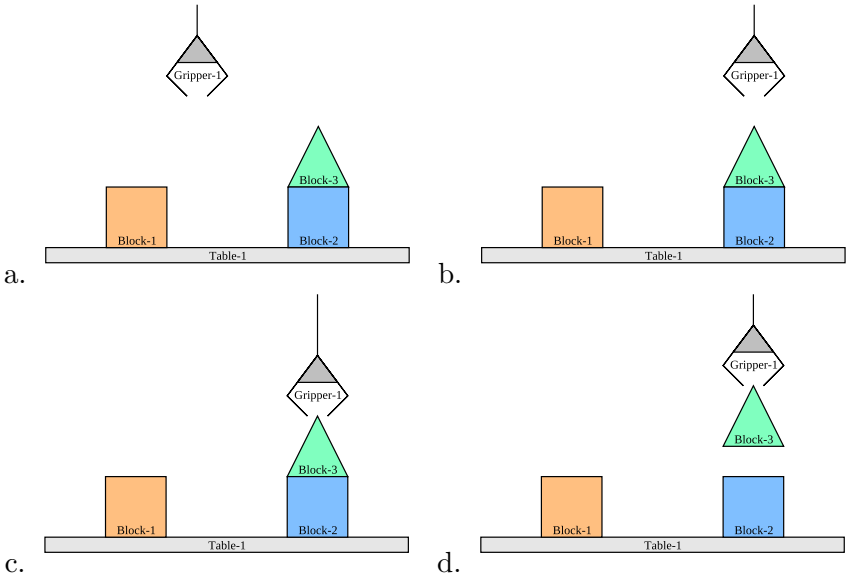

Figure 1: Block building domain. (a) the initial starting physical state of the gripper in the block building domain. (b) how a planning process could represent the result if the gripper were to *move right*. (c) how a planning process could represent the result if the gripper were to try to *grab* the block below. (d) how a planning process could misrepresent the later result if the gripper were to try to *grab* the block below.

4.2 Correct Knowledge

4.3 Incorrect Knowledge

5 A Concrete Plan

5.1 Credit Assignment Metric

First, I will develop a metric by which I can measure the performance of my credit assignment algorithm versus other algorithms. At one extreme, we have the brute force approach to learning, in which all possible concepts are re-learned at every opportunity (e.g. at every step in a Markov stepped system, or at every knowledge event in my real-time event driven system).

At the other extreme, we re-learn only those concepts whose fundamental training knowledge has changed (instances, hypothesis space, or features). My approach to tracing the provenance of deliberative knowledge through the reactive plan execution agencies will be somewhere between these two extremes of efficiency.

– Evaluation Metrics

– credit assignment: the process of choosing a subset from the total possible set of concepts that should be focused on as being responsible for the occurrence of a given event.

EXACTLY WHAT IS THE METRIC HERE? HOW WILL IT BE MEASURED? YOU HINT AT SPEED BELOW, AND IF THIS IS TRUE, WILL IT BE MEASURED IN DECISION-CYCLE TIME? DEFINE AND OPERATIONALIZE YOUR METRIC.

The credit assignment process can be more or less efficient in focusing the learning resources toward concepts that can be re-learned.

5.2 Temporal Credit Assignment

I can measure the performance of my causal credit assignment algorithm versus a typical temporal credit assignment algorithm, such as an adaptation of the Temporal Difference (TD) learning algorithm commonly used in the field of reinforcement learning.

HAVE YOU ALREADY IMPLEMENTED THIS ALGORITHM? WILL YOU USE AN EXISTING IMPLEMENTATION?

5.3 Causal Credit Assignment

The goal of these tests are to show that learning by provenance speeds the standard temporal credit assignment method, which assigns credit to the sequence of immediately previous states and actions.

ABOVE YOU MENTION YOUR CREDIT ASSIGNMENT *VERSUS* A STANDARD ALGORITHM. HERE YOU STATE THAT YOUR VERSION WILL ACCELERATE THE STANDARD ALGORITHM. DO YOU MEAN THAT YOURS IS FASTER THAT THE STANDARD IN THE LIMIT? OR DO YOU INTEND TO COMBINE THEM TO SPEED UP THE TEMPORAL ALGORITHM?

– I will focus on re-learning category concepts given new training instances that would change those concepts.

– Now, given the above metric for what I will actually be measuring as a performance metric, here are two specific deliberative learning tasks

where my causal learning will be superior to a temporal credit assignment method. This is primarily due to the delayed time between deliberation about knowledge and the execution of that knowledge, resulting in a precondition check form of failure. Specifically, the precondition check could simply be what would be Removed by a trans-frame, but doesn't necessarily exist in the physical knowledge at the time of execution. The following are two examples of scenarios:

5.4 The Social Knowledge Trust Task

Some plans can be learned through a social communication language. Physical plans are added to the deliberative knowledge with the added relationship with the appropriate social provenance markers, e.g. Gripper-1 may represent that a given plan is told to it by The-User (or another social source).

The Decision to use a plan that accomplishes a given goal from one knowledge source or another can be learned, but the learning of this type of action is difficult because of the often indirect temporal connection between this deliberative Decision involved in choosing or creating a plan and the actual execution failure, resulting from executing a plan that has derived from that deliberative decision.

THIS LONG RUN-ON SENTENCE IS TYPICAL OF YOUR WRITING AND PLACES THE BURDEN OF UNDERSTANDING UPON THE READER. WHAT IS LEARNED? A DECISION OR AN ACTION? FURTHERMORE, YOU PREVIOUSLY DISCUSSED LEARNING CONCEPTS. IMPLEMENT A SINGLE LEARNING ALGORITHM AND USE IT TO ILLUSTRATE YOUR THEORY. AGAIN, DECIDE WHAT YOU WILL NOT SAY.