UDT WITH KNOWN SEARCH ORDER

TSVI BENSON-TILSEN

Abstract. We consider logical agents in a predictable universe running a variant of updateless decision theory. We give an algorithm to predict the behavior of such agents in the special case where the order in which they search for proofs is simple, and where they know this order. As a corollary, "playing chicken with the universe" by diagonalizing against potential spurious proofs is the only way to guarantee optimal behavior for this class of simple agents.

Contents

1. Introduction

Updateless decision theory (UDT) was developed to deal with Newcomb-like scenarios—situations where the behavior of the environment may depend on the logical fact of what decision an agent makes in certain situations, as opposed to depending only on events that were physically caused by the agent's actions. See [Altair, 2013] or [Hintze, 2014] for an exposition of this flavor of decision theory.

In Wei Dai's original formulation of UDT, there is an agent A deciding between possible observation-action policies π that it may follow [Dai, 2009, 2010]. The agent picks the policy π for which the expected utility of the universe is highest, where the expectation for π is computed after conditioning on the statement, "The algorithm A always follows π." In this paper we explore an issue with one formulation of UDT in a logical setting, due to Vladimir Slepnev.

Section 2 explains that issue.

Section 3 gives the formal setting for proof-searching agents.

Section 4 proves the main results characterizing the behavior of proof-searching UDT agents with too much self-knowledge.

Section 5 discusses other work on the problem of logical counterfactuals.

Date: August 22, 2014.

2. The problem of spurious counterfactuals

Vladimir Slepnev provided a precise model for UDT in the setting of an agent algorithm A that lives in a deterministic, known universe U [Slepnev, 2011]. For simplicity, we will assume that A takes a single action, and then the universe evaluates to a utility. The agent uses a formal system such as PA to search for proofs of statements of the form $A(U) = a \to U(A) = u$, for a utility $u \in \mathbb{R}$ and an action a. When the agent has proved such a statement for each available action, it chooses the action with the highest resulting u.

We assume that the agent's encoding of the environment is clear enough that basic theorems are easily proven, like $\neg(A(U) = a \wedge A(U) = b)$ for $a \neq b$. To avoid non-halting proof searches, we assume that A has access to a halting oracle. That is, A can make queries in the form of a statement in the language of PA, and receive "yes" if that statement is provable in PA and "no" otherwise.

This decision algorithm is doing counterfactual reasoning, where $A(U) = a \to U(A) = u$ is intended to represent the counterfactual, "If the algorithm A were to output a in universe U, a utility of u would result." However, logical implication does not necessarily correspond to the intuitive notion of counterfactual, and indeed we may have spurious counterfactuals. Indeed, if A knows enough about itself, so to speak, we may have that $\mathsf{PA} \vdash A(U) \neq a$. Then $A(U) = a \to U(A) = u$ is vacuously true for any u. But this "counterfactual" statement is seemingly an accident, or at least irrelevant to decision making, because its truth was not a result of any relationship between U and agents that take action a.

To be concrete, say the universe is very simple: it gives utility 5 if A does a_5 and it gives utility 10 if A does a_{10}. Those are the only actions available to A, and A has access to the source code of U and A. We would like for A to prove $A(U) = a_5 \to U(A) = 5$ and $A(U) = a_{10} \to U(A) = 10$, in which case A will actually take action a_{10}. By the assumption of a transparent universe, there are proofs of both of these statements; the worry is that A might fail to find these proofs, instead finding proofs of some other statements that lead it take action a_5.

To see how this might happen, recall Löb's theorem:

$$\forall \varphi : \mathsf{PA} \vdash \Box_{\mathsf{PA}}[\varphi] \to \varphi \implies \mathsf{PA} \vdash \varphi$$

Here $[\varphi]$ is the Gödel number of φ, and $\Box_{\mathsf{PA}}[\varphi]$ is the provability predicate for PA, which states that there exists a proof in PA of φ. We usually denote $\Box_{\mathsf{PA}}$ by just $\Box$. We will also use $\Box[\varphi]$ in the meta-language as a shorthand for $\mathsf{PA} \vdash \varphi$.

Now, the agent A might reason as follows:

$\mathsf{PA} \vdash A(U) = a_5 \to U(A) = 5$

$\mathsf{PA} \vdash \Box[A(U) = a_5 \to U(A) = 5]$

$(*)$ $\mathsf{PA} \vdash A$ finds the proof of $[A(U) = a_5 \to U(A) = 5]$

$\mathsf{PA} \vdash \Box[A(U) = a_5] \to \Box[A(U) = a_5]$

$\mathsf{PA} \vdash \Box[A(U) = a_5] \to \Box[A(U) \neq a_{10}]$

$\mathsf{PA} \vdash \Box[A(U) = a_5] \to \Box[A(U) = a_{10} \to U(A) = 0]$

$(*)$ $\mathsf{PA} \vdash \Box[A(U) = a_5] \to A$ finds the proof of $[A(U) = a_{10} \to U(A) = 0]$

$\mathsf{PA} \vdash \Box[A(U) = a_5] \to A(U) = a_5$

$\mathsf{PA} \vdash A(U) = a_5$

In words, this says that under the assumption that the agent provably does a_5, spurious counterfactuals about the outcome of a_{10} can lead A to conclude that it in fact takes action a_5. Then in the final step of the derivation, we deduce that by Löb's theorem, the agent will actually conclude that it takes action a_5. Then by soundness, A does in fact do a_5. This undesirable behavior is known as "the 5-and-10 problem" or "the problem of spurious counterfactuals."

The steps marked by $(*)$ cannot always be carried out (otherwise the agent could also prove $A(U) = a_{10}$ in the same way, which would be an outright contradiction in PA). For A to go from "there exists a proof" to "A will find that proof" requires that A have significant information about the order in which it searches for proofs. In particular, it needs to know that, for instance, if a proof of $A(U) = a_{10} \to U(A) = 0$ exists, then it comes before any proof of $A(U) = a_{10} \to U(A) = u$ for $u \neq 0$.

Thus it seems that this particular kind of self-knowledge is dangerous; if a proof-searching agent knows in what order it would find proofs *if* they existed, then that agent may fail to reason correctly about certain actions. As reported in [Slepnev, 2011], Vladimir Nesov patched this problem by inserting a clause into the decision procedure of a UDT agent. This clause says that before A starts looking for any proofs of $A(U) = a \to U(A) = u$, it first checks if there are any proofs of the form $A(U) \neq a$. If it finds such a proof, it immediately takes action a.

The idea is that, assuming PA is sound, A will never actually prove $A(U) \neq a$ and then do a. So A will always proceed to the usual proof-search step. It will not be vulnerable to spurious counterfactuals because the implication $A(U) = a \to U(A) = u$ is never proved by first proving $A(U) \neq a$.

This strategy has been termed "playing chicken against the universe"; it forces itself and the universe to be opaque by diagonalizing against any transparency. Playing chicken is not a wholly satisfactory solution to the problem of spurious counterfactuals because it seems that the agent is deliberately sabotaging its own self-knowledge. Also, it is not clear whether the diagonalization clause is reflectively consistent—i.e. whether or not a UDT agent would delete the clause from its source code if given the opportunity. Furthermore, playing chicken doesn't extend so straightforwardly to probabilistic settings; we will discuss more in section 5.

With this in mind, it might be helpful to understand more closely how spurious counterfactuals occur. In sections 3 and 4 we specify a certain model for a UDT agent with finitely many possible actions, and investigate its behavior.

3. Setting up the universe and the agent

The agent, an algorithm A, is acting in an arbitrary fixed universe, an algorithm U. The agent picks a certain action $a_i \in \{a_1, \ldots, a_n\}$, after which the universe returns some utility. Let u_i denote the actual utility resulting from action a_i. This utility should depend only on A's action; the universe is behavioral, i.e. for all agents A and B, if $A(U) = B(U)$ then $U(A) = U(B)$. Assume that the universe is transparent in the following sense:

$$(A(U) = a_i \implies U(A) = u_i) \implies \Box[A(U) = a_i \to U(A) = u_i]$$

The agent will search for proofs of statements of the form $A(U) = a \to U(A) = u$ using a halting oracle—it knows, for any given statement, whether or not that statement is provable in PA. This is a formalization of the step in the spurious

counterfactual described above where the agent goes from "there exists a proof" to "A will find a proof"; with a halting oracle, these statements are the same.

Remark 3.1. It may seem overcomplicated to search for proofs of $A(U) = a \to U(A) = u$. Indeed, under the assumption that the universe is behavioral, we could just as well search for proofs of $\pi(U) = a \to U(\pi) = u$, where π ranges over the possible action policies. This would give correct evaluations of the consequences of different policies.

However, our motivation for investigating these counterfactuals is to understand agents capable of, for instance, using logical deduction to recognize other copies of themselves. Deducing and acting on the logical consequences of $A(U)$ being one thing or another is closer to capturing the reasoning of such an agent.

Define the universe U:

Data: the agent A
Result: a utility u_i
for *each action* $a_i, i \leq n$ **do**
  if $A(U) = a_i$ **then**
    return u_i
  end
end
return 0

Algorithm 1: The universe U

It will not matter in what follows, but we allow U access to a double halting oracle so that it can run A and return a utility even if A does not halt. Now we define the agent A:

Data: the universe U, and for each a_i an order $\{y_1^i, y_2^i, \dots\}$ of the distinct possible utilities
Result: an action a_i
for *each action* $a_i, i \leq n$ **do**
  for *each outcome* $u \in \{y_1^i, y_2^i, \dots\}$ **do**
    if $\square_{PA}[A(U) = a_i \to U(A) = u]$ **then**
      store $\leftarrow (a_i, u)$;
      break for
    end
  end
end
let $k = \max\{i \mid (a_i, u) \in \text{store and } (\forall (a_j, u') \in \text{store})(u' \leq u)\}$;
return a_k

Algorithm 2: The agent A

This agent has access to a complete description of the universe, including itself, and uses a halting oracle to find proofs. Since the universe is transparent, A will always find at least one proof for every action, and will store exactly one (a_i, u) for each action. This also allows for infinitely many possible utilities, since no search will go on forever.

The return statement gives the tiebreak rule "take the highest numbered action that gives maximal payoff". These tiebreaks are necessary but inconvenient, so we abstract away the outcomes as A sees them. Recall from the definition of A that the $y^i_\bullet$ are the possible outcomes for a_i.

Definition 3.2 (Ordering on outcomes with tiebreak)**.** Write $y^i_s \succ y^j_t$ if $i > j$ and $y^i_s \geq y^j_t$, or if $i < j$ and $y^i_s > y^j_t$.

Then A takes the a_i with the $\prec$-greatest u with $(a_i, u) \in \mathit{store}$. We are abusing the notation by writing $y^i_s \succ y^j_t$; technically we should write $(y^i_s, i) \succ (y^j_t, j)$, since the y^i_s are just numbers. (Note that since A never compares y^i_s to y^i_t, we won't need to either.)

4. Playing chicken is the only way to win

Let $f_i = y^i_1$ be the first outcome searched for a_i, and let $u_i = U(A)$ be the true outcome given $A(U) = a_i$.

Lemma 4.1. *The action $A(U)$, as a function of the search order, depends only on the f_i and the u_i for $i \leq n$. That is, $A(U)$ always stores either (a_i, f_i) or (a_i, u_i). The latter statement is provable in PA for fixed i.*

Proof. Fix $i \leq n$ and say $u_i = y^i_s$. By transparency of U we have $\Box[A(U) = a_i \to U(A) = y^i_s]$. So either A will store (a_i, y^i_s), or else for some $t < s$ we have

$$\Box[A(U) = a_i \to U(A) = y^i_t]$$

In the latter case, since $\Box[\neg(U(A) = y^i_s \wedge U(A) = y^i_t)]$, we have $\Box[A(U) \neq a_i]$. Then by a vacuous counterfactual, $\Box[A(U) = a_i \to U(A) = y^i_1]$, so A will store (a_i, f_i). By Σ_1-completeness of the provability predicate $\Box$, this argument can be carried out in PA.

Since A's action is determined by which action/outcome pairs have been stored, we are done. □

Definition 4.2 (The aboveness and domination relations)**.** Let $i \neq k$. Say that a_k is **above** a_i if $u_k \succ u_i$, and otherwise say that a_k is **below** a_i. The action a_k **dominates** a_i if both $u_k \succ f_i$ and $f_k \succ f_i$.

Definition 4.3 (Safety of actions)**.** The action a_k is **safe from above** if it dominates a_i for all a_i above a_k. The action a_k is **safe from below** if it is not dominated by a_i for any a_i below a_k. The action a_k is **safe** if it is both safe from above and safe from below.

Lemma 4.4. *If $A(U) = a_k$, then a_k is safe.*

Proof. By soundness, $\neg\Box[A(U) = a_k \to U(A) = y^k_s]$ for any $y^k_s \neq u_k$, so A stores (a_k, u_k).

Suppose first that a_k is not safe from above, so a_k fails to dominate some action a_i above a_k. It can't be that A stored (a_i, u_i), since by aboveness we have $u_i \succ u_k$; otherwise $A(U) \neq a_k$, contrary to the hypothesis. So in fact A stores (a_i, f_i), and we have $f_i \prec u_k$ for the same reason.

This happens only if $\Box[A(U) = a_i \to U(A) = f_i]$, so by Σ_1-completeness, $\Box[\Box[A(U) = a_i \to U(A) = f_i]]$. Then $\Box[A(U)$ stores $(a_i, f_i)]$. Since a_k fails to

dominate a_i, but $u_k \succ f_i$, it must be that $f_k \prec f_i$. Then we have the Löbian argument:

$$\mathsf{PA} \vdash \Box[A(U) \neq a_k] \to A(U) \text{ stores } (a_i, f_i) \text{ and } (a_k, f_k)$$
$$\mathsf{PA} \vdash \Box[A(U) \neq a_k] \to A(U) \neq a_k$$
$$\mathsf{PA} \vdash A(U) \neq a_k,$$

contradicting soundness. Therefore, a_k is safe from above.

Suppose now that a_k is not safe from below, so a_k is dominated by some action a_i below a_k. That is, $u_i \succ f_k$ and $f_i \succ f_k$. But then, by Lemma 4.1 in PA for i,

$$\mathsf{PA} \vdash \Box[A(U) \neq a_k] \to A(U) \text{ stores } (a_k, f_k) \text{ and either } (a_i, f_i) \text{ or } (a_i, u_i)$$
$$\mathsf{PA} \vdash \Box[A(U) \neq a_k] \to A(U) \neq a_k$$
$$\mathsf{PA} \vdash A(U) \neq a_k,$$

again contradicting soundness. Therefore a_k is safe from below, and so a_k is safe. 🕮

Theorem 4.5. *$A(U) = a_k$ for the unique safe a_k.*

Proof. The agent is well-defined, so by Lemma 4.4, $A(U) = a_k$ for some safe a_k. Each a_i above a_k is dominated by a_k, and so is not safe from below. Likewise, each a_i below a_k does not dominate a_k, and so is not safe from above. Done.

Let us see directly why there always exists a safe action a_k. Say that a_k **defeats** a_i if either a_k dominates a_i or a_k is above a_i and a_i does not dominate a_k; so a_k is safe if and only if a_k defeats all $a_i, i \neq k$.

Defeat is transitive: suppose we have a_i above a_j above a_k. If a_k defeats a_j defeats a_i, then by transitivity of $\prec$ via f_j we have that a_k defeats a_i. If a_k defeats a_i defeats a_j, then either $f_i \succ f_j$ or $f_i \succ a_j$. If $f_i \succ a_j$, then $a_k \succ a_j$, contradicting that a_j is above a_k. So in fact $f_i \succ f_j$, and by transitivity of $\prec$ via f_i we have that a_k defeats a_j. The other cases are similar.

Therefore, the maximum action under the defeats ordering is safe. We can compute the unique safe action a_k given the f_i and u_i as follows: do a single pass through the actions, maintaining the largest action so far, under the defeats ordering; the largest action overall is the action $A(U)$. 🕮

Suppose we take all the f_i to be ∞, i.e. a formal symbol representing something that is $\prec$-greater than any other number. Then no action a_k dominates any a_i, since $u_k \prec f_i$. Hence the defeats relation is just the aboveness relation, and so A will take the action above all other actions, i.e. the action with the greatest actual utility. (Note that the f_i may have some order among themselves, but this will not affect the behavior of A. If there is a maximal possible outcome y, which happens e.g. if there are only finitely many outcomes, then we can set $f_i = y$.)

This strategy is analogous to Nesov's strategy of playing chicken with the universe. Choosing $f_i = \infty$ means that if $\Box[A(U) \neq a_i]$, then A stores (a_i, ∞). If this ever happens, A will take some action a_k with (a_k, ∞) stored, and this will contradict soundness. Since this won't happen, A will actually prove only the "correct" counterfactuals $A(U) = a_i \to U(A) = u_i$.

In fact, this is the only way to ensure victory:

Corollary 4.6 (Playing chicken with the universe is the only way to win)**.** *Given just the set of actions and the set of possible outcomes y, the assignment $\forall i(f_i = \infty)$*

or $\forall i(f_i = \sup y)$ *is the only search strategy which guarantees that* $U(A)$ *is maximal for available* a_k*.*

Proof. Take the $\prec$-least f_i, and suppose that $f_i < y$ for some possible outcome y. Take any other a_k; then it may be that $u_k = y$. By choice of f_i we have $f_k \succ f_i$, and by hypothesis we have $u_k \succ f_i$. Hence we have that a_k dominates a_i. But then u_i may be arbitrarily large, while $A(U)$ will never be a_i.

So we must have $f_i \geq y$ for all possible outcomes y, i.e. we have $f_i = \infty$, or at least $f_i = \sup y$ if the supremum exists. Since f_i is $\prec$-minimal, the same equations hold for all the f_j. □

5. Discussion

For ease of analysis our model uses logical deduction as the only mode of reasoning, but the goal is to understand agents reasoning naturalistically under uncertainty. Updateless decision theory can be cast in terms of probabilistic beliefs and probabilistic reasoning [Fallenstein, 2014]. In that setting, the agent has a probability distribution over possible complete first order theories of the world. Instead of searching for proofs, the agent conditions its beliefs on the statement "My algorithm always follows policy π," and computes the expected value of its utility function according to that conditional distribution. It follows the policy with the highest expected utility.

The probabilistic setting has its own complications; for one thing, the agent is now an algorithm that makes calls to an entire probability distribution over logical theories. This can be represented coherently using the framework developed in [Christiano et al., 2013]. But if the agent has too much information about its probabilistic beliefs, it may be able to deduce facts about the expected utilities of some actions, and so deduce that it will not take certain actions.

The problem of spurious counterfactuals then becomes the problem of conditioning on statements of probability 0, as the agent will attempt to do when evaluating its available actions. The most straightforward analog of playing chicken with the universe—immediately taking any action that has probability 0—may only, in Christiano's framework, force those actions to have infinitesimal probability.

The present paper offers some small evidence that the unappealing strategy of crippling one's own self-knowledge may be unavoidable if we allow agents to be logically omniscient, and this inevitability is further supported by the appearance of these issues even in the probabilistic setting. This gives one motivation for the development of a theory of logical uncertainty—how to reason under uncertainty *without* the assumption that deductive consequences of an agents beliefs are necessarily available to that agent. See [Christiano, 2014] for current work on logical uncertainty. This may offer a way to coherently reason counterfactually about a computation doing something other than what it actually ends up doing.

References

Alex Altair. A comparison of decision algorithms on newcomblike problems. `http://intelligence.org/files/Comparison.pdf`, 2013. Machine Intelligence Research Institute.

Paul Christiano. Non-omniscience, probabilistic inference, and metamathematics. `http://intelligence.org/files/Non-Omniscience.pdf`, 2014. Machine Intelligence Research Institute.

Paul Christiano, Eliezer Yudkowsky, Marcello Herreshoff, and Mihaly Barasz. Definability of truth in probabilistic logic (early draft). http://intelligence.org/files/DefinabilityTruthDraft.pdf, 2013. Machine Intelligence Research Institute.

Wei Dai. Towards a new decision theory. http://lesswrong.com/lw/15m/towards_a_new_decision_theory/, 2009.

Wei Dai. Explicit optimization of global strategy (fixing a bug in UDT1). http://lesswrong.com/lw/1s5/explicit_optimization_of_global_strategy_fixing_a/, 2010.

Benja Fallenstein. SUDT: A toy decision theory for updateless anthropics. http://lesswrong.com/lw/jpr/sudt_a_toy_decision_theory_for_updateless/, 2014.

Daniel Hintze. Problem class dominance in predictive dilemmas. https://www.dropbox.com/s/n262sxmsbzx305a/Hintze_Problem%20Class%20Dominance_Approved.pdf, 2014. Barrett, The Honors College at Arizona State University.

Vladimir Slepnev. A model of UDT with a halting oracle. http://lesswrong.com/lw/8wc/a_model_of_udt_with_a_halting_oracle/, 2011.