Problem Class Dominance in Predictive Dilemmas

Daniel Hintze

April 23, 2014

Abstract

One decision procedure dominates a given one if it performs well on the entire class of problems the given decision procedure performs well on, and then goes on to perform well on other problems that the given decision procedure does badly on. Performing well will be defined as generating higher expected utility before entering a problem. In this paper it will be argued that the timeless decision procedure dominates the causal and evidential decision procedures. It will also be argued in turn that the updateless decision procedure dominates the timeless decision procedure. The difficulties of formalizing a modern variant of the ”smoking gene” problem will then be briefly examined.

1 Introduction

Decision Theory can quickly get ideological, potentially raising questions about ethics, economics, causality, and even the nature of time. Yet we are faced with decisions every day. We need some principle to frame and decide upon choices we will make in the future. We want something that takes the model of the world we have formed (at least in part) from our sense data and uses this information to recommend an action that we should take. We will call something that does this a decision procedure.

Imagine that we are attempting to choose a decision procedure for a hypothetical agent (which could be ourselves). I argue there are two major topics we have to consider before endorsing a decision procedure this agent will use in the future:

1. What is our expectation of the kinds of problems the agent will face? What is the payoff structure of the problems? What is the mechanism by which the problem settles on those payoffs? Will the problems be of the same type or will there be a distribution of different types? Do we know the probability of different problem types or are we experiencing uncertainty about their distribution?

2. Given each decision procedure the agent can use, what outcomes will it obtain against various problem types?

I certainly do not intend to address all of this in one paper, but I give it as a useful background to consider as we focus in on our main topic. When we try to decide between various decision procedures, one very useful criterion to consider is whether one decision procedure dominates a given one.

The logic of dominant actions is one that I trust many readers are familiar with. It often makes great intuitive sense to choose an action that will leave you at least as well off as another action will, no matter what state of the world obtains. However, what if we apply that reasoning to our choice of decision procedures? We may find that the decision procedure we would choose is not one that recommends some traditionally dominant actions.

In this vein, I offer the idea of problem class dominance for decision procedures. We can then say that one decision procedure dominates another one. By this I mean that it performs well on all problems that a given decision procedure performs well on, and then goes on to perform well on a class of problems the given decision procedure does not perform well on.

I exclude from this analysis problems which are specifically and completely biased for or against a certain type of procedure. For example, imagine a problem that looks at an agent's decision procedure and grants (or removes) utility based solely upon whether it matches a predefined decision procedure.[1] If we admit this possibility, it appears impossible to talk of one decision procedure dominating another across the board. We could always design some problem which will disrupt the dominance. However, there still exists a broad range of problems where this is not happening, and this is where we will focus our attention.

Specifically, we are looking at problems where another agent (or nature itself) can predict what action an agent is going to take. Presumably they have some insight into the output of an agent's decision procedure in order to do so. In some situations, they might be able to actually run the agent's decision procedure and predict with perfect accuracy what the agent will do. This idea becomes particularly salient if you are thinking about agents who are computer programs, which will often have a well defined decision procedure that can then be simulated to predict an action with perfect fidelity.

In fact it is from such a background that two of the decision procedures we will discuss originate. However, they have potential applicability whenever another agent might have some degree of predictive ability regarding your actions, as we will soon see.

The course of this paper is a simple one. We will look at two possible problems an agent could face, and then take a look ahead at four possible decision procedures for handling those problems. We will then turn back to the problems and attempt to formalize them so that we can more clearly apply the decision procedures. We will then apply the decision procedures in some detail, which should also help in understanding how they process problems in general. Throughout this application, we will be considering the implications from a standpoint of problem class dominance. Finally, we will conclude with arguments for the dominance relationships between the various procedures. We will also call for either counterexamples to our dominance arguments or new procedures that are dominant over an even wider range of problems.

[1] That being said, I would like to strongly emphasize that calling a problem "biased" should be considered provisional and perhaps a call to further attention rather than definitive and an excuse (by itself) to ignore the problem or approaches that handle it well.

2 Some Problematic Situations

I'd like to start out by giving some examples of potential real world problems an actual person could face. These are highly selected problems, designed to make a specific point. For this reason they involve some relatively unusual situations. I will first present the problems, with a focus on presenting them realistically to build intuitions. Then we will attempt to formalize the problems.

2.1 Parfit's Hitchiker

You are stranded in the desert. You are out of supplies, and do not have long to live if you stay in the desert. Unfortunately you do not even have any money. You are to some degree transparent, meaning that you are not a perfect liar- it is sometimes possible for another human to tell when you are lying. Paul Ekman, an expert on reading facial microexpressions, is driving across the desert when he finds you. It is not worth his time or the gas to take you back to civilization unless you will give him $20 in gas money when you arrive in the city. It is not worth his effort to bother legally enforcing a promise of $20. Once you reach the city, it is therefore completely possible for you to run off and never see him again. There is not a commitment mechanism you can put in place to prevent yourself from doing this. You know all of this, and cannot remove that knowledge from your head.

You begin to consider future actions. You know that when you reach the city you will be strictly better off with the $20. Using some decision procedures you will run off and not pay your driver. If you are self-aware, you know that you will do this. So you will have to lie to the driver in the desert. Paul Ekman will detect your lie with 5% probability, and agents who would run off with the $20 will be more likely to starve in the desert. [2]

2.2 The Curious Benefactor

A wealthy woman decides to play a game with one of her associates. She flips a fair coin. If it comes up tails she will ask the associate to pay her $5. If it comes up heads she will give 1 million dollars to the associate, but only if she predicts that the associate would have given her the $5 if the coin had come up tails. She then flips the coin and it comes up tails. She explains the situation to her associate and asks them for the $5. Should the associate give her the $5?[3]

[2] Notice that Paul Ekman never accidentally thinks you are telling a lie when you are being truthful, yet he only has a mere 5% probability of detecting a lie. This lack of false positives is merely to simplify some calculations later so that they are easier to follow. The analysis that is performed on the problem applies with generality as long as Ekman has a higher probability of thinking that a lie is a lie than that a truth is a lie (and as long as the payout differences are substantial enough to make the detection matter).

[3] A version of this game was actually played by Anders Sandberg against the philosopher Nick Bostrom for 20£ vs. 1£, using an unknown decimal digit of pi instead of a coin. Bostrom paid the 1£, and it even turns out that Sandberg had correctly predicted that he would. See Yudkowsky, 2014 in the references section for details.

3 Decision Procedures

An agent's decision procedure takes sense data and outputs an action. These statements of various decision procedures combine a statement of the utility rule with the condition that the procedure selects the act with maximal utility.[4]

3.1 The "Evidential" Decision Procedure

$$EDP(s) = \arg\max_a \sum_i^n U(O_i) * P(\, O_i \,|\, a \cap s)$$

Given a string of sense data (s), finds the action (a) that generates the highest expected utility by weighting the utility of outcomes (U(O_i)) by their probability conditional on the action.

3.2 The "Causal" Decision Procedure

$$CDP(s) = \arg\max_a \sum_i^n U(O_i) * P(\, a \mathrel{\Box\!\!\rightarrow} O_i \,|\, s)$$

Finds the action that generates the highest expected utility where the outcomes are weighted by the probability that the action will cause them (Pearl's "do calculus").[5]

3.3 The "Timeless" Decision Procedure

$$TDP(s) = \arg\max_a \sum_i^n U(O_i) * P(\, \ulcorner TDP(s) := a \urcorner \mathrel{\Box\!\!\rightarrow} O_i \,|\, s)$$

Finds the action that will maximize the expected utility where the probabilities are the probability that if the procedure outputs the action[6], it will cause the outcome to occur.

3.4 The "Updateless" Decision Procedure

$$UDP(s) = \arg\max_f \sum_i^n U(O_i) * P(\, \ulcorner UDP := f : s \mapsto a \urcorner \mathrel{\Box\!\!\rightarrow} O_i \,)$$

Finds a function mapping sense data to actions that will maximize the utility of outcomes weighted by the probability that the outcome will be caused by the procedure returning that mapping.

[4] In the case of a tie, we will need a tiebreaking method. We can imagine many possible tie breaking criteria, so each statement is really a family of decision procedures utilizing various tie breaking criteria. On the particular problems we are using as examples there are no ties, so we are ignoring this complication for now.

[5] The symbol $\Box\!\!\rightarrow$ is commonly used to express conditionals that are "nonbacktracking" (and hence causal). For example $P(a \mathrel{\Box\!\!\rightarrow} b)$ indicates the probability that a will cause b (or that b is inevitable).

[6] The corner quotes $\ulcorner$*example*$\urcorner$ are used to allow the procedure to refer to its own output.

4 Formalizing The Problems

We will be using causal diagrams inspired by Judea Pearl's representation of bayes nets. These are a way of compactly representing a probability distribution using non-backtracking conditionals.

4.1 Formalizing Parfit's Hitchhiker

Let's start out by looking at a causal diagram of the problem. This diagram contains some superfluous nodes, but they are included to help make clear the assumptions going into various steps of the problem. The arrows represent the causal influence of one variable on another.

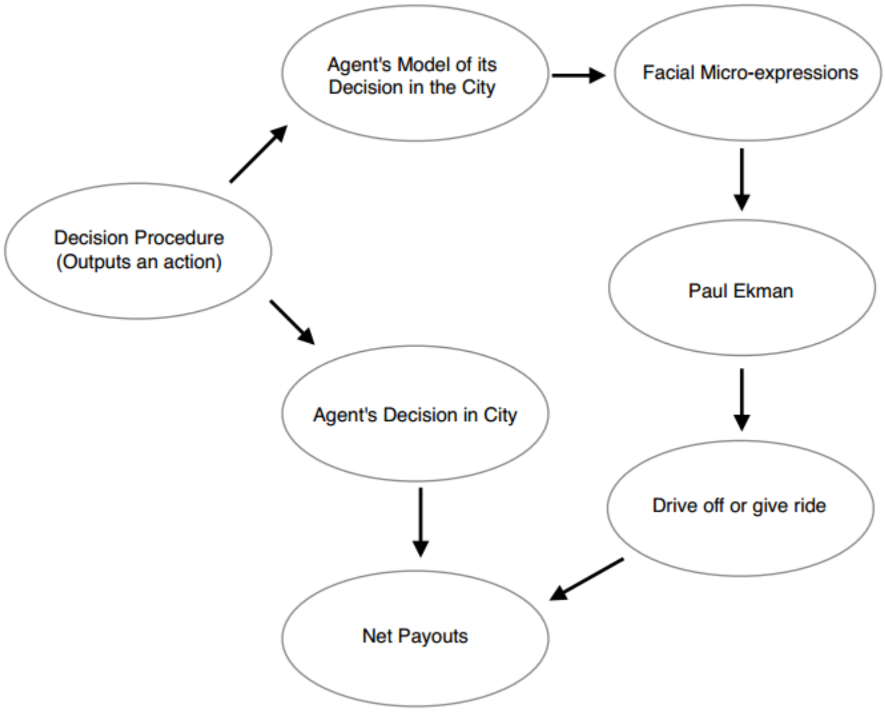

Figure 1: A diagram of the Hitchhiker Problem.

The agent has a model of their decision in the city that is created using their perfect knowledge of their decision procedure. This model leads to facial microexpressions that can be read by Paul Ekman. Based upon this reading, he decides whether or not to give the agent a ride. Meanwhile, the agent's decision procedure determines what the agent's decision will be if it arrives in the city. The net payoff is then determined by Paul Ekman's decision and the agent's decision if it does actually get a ride and reach the city.

It may be helpful to look at each node in some finer detail.

Decision Procedure Takes in sense data and outputs an action

Agent's Model of its Decision in the City The agent runs a hypothetical string of sense data through its own decision procedure to see the action the procedure would output if the agent found themselves in the city. This is how the agent knows what it will do if that situation obtains.

Facial Expressions Display the agent's actual model of "decision in city" to Ekman.

Paul Ekman Can read lies from facial expressions correctly 5% of the time. Returns "lie" or "truth".

Drive off or give ride Outputs "ride" if Ekman returns "truth". Outputs "no ride" if Ekman returns "lie".

Agent's Decision in City Output of agent's decision procedure determines what the agent will decide. Outputs "pay 20" or "don't pay 20"

The possible actions the agent can take are:

$$a_p = \text{"stay and pay \$20"}$$

$$a_l = \text{"leave and don't pay"}$$

The possible actions that Paul Ekman can take are:

$$r_y = \text{"give ride to city"}$$

$$r_n = \text{"don't give ride"}$$

Outcomes and corresponding utility levels[7]:

Condition	Outcome	Utility	Description
r_n	O_1	0	"Die in Desert"
$r_y \cap a_p$	O_2	1000	"Live and Give"
$r_y \cap a_l$	O_3	1020	"Live and Leave"

Table 1: The payouts for the Hitchiker Problem

4.2 Formalizing The Curious Benefactor

The benefactor has a model of what the agent will do, computed using the agent's decision procedure [8]. She flips a coin, which creates one of two counterfactual worlds (divided by the dotted line). If the coin is T, the agent is asked for $5, and its decision determines the payoffs (the coin flip already being fixed at "T"). If the coin comes up heads, then the benefactor querys her model of the agent, and using the agent's decision procedure, decides whether the agent

[7]Note that risk neutrality and a fairly direct mapping of money to utility are being imposed merely to make the arguments easier to follow for some readers. The arguments do not hinge on risk neutrality, although we do need sufficient utility differences for our choice between some of the procedures to matter.

[8]Note that the dashed arrow connecting the decision procedure to the curious benefactor is dotted merely to distinguish it from the other arrows, although it could also been seen as denoting the unique role it has in crossing the counterfactual dividing line.

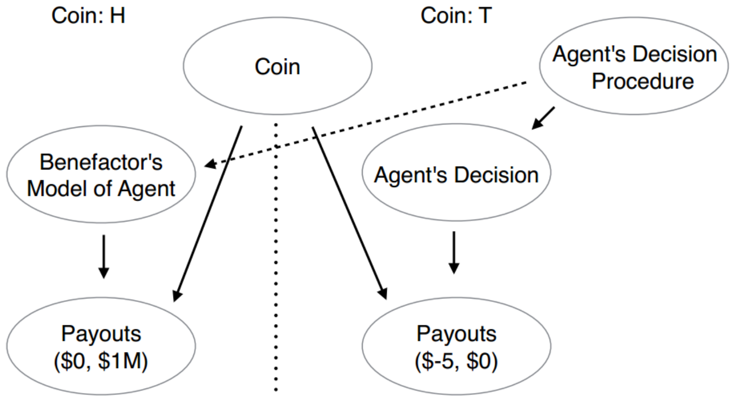

Figure 2: A diagram of the Curious Benefactor Problem.

would have given $5 in the situation where the coin came up T. If the agent would have given the $5, then she will give the agent $1,000,000.

The agent's possible actions if asked for $5 are:

$$a_1 = \text{"give \$5"}$$
$$a_2 = \text{"give nothing"}$$

The benefactor's possible actions after querying its model are:

$$b_1 = \text{"give 1M"}$$
$$b_2 = \text{"don't give 1M"}$$

The Benefactor outputs b_1 if her model of the agent returns a_1 when fed sensory input corresponding to world T.

The Benefactor outputs b_2 if her model of the agent returns a_2 when fed sensory input corresponding to world T.

The possible outcomes and associated utilities are:

Condition	Outcome	Utility	Description
$T \cap a_1$	O_1	-5	"Asked and Paid"
$T \cap a_2$	O_2	0	"Asked and didn't pay"
$H \cap b_2$	O_3	0	"Not Given Million"
$H \cap b_1$	O_4	1M	"Given Million"

Table 2: The payouts for the Curious Benefactor Problem

5 How do the decision procedures interact with these problems?

5.1 Defining Success and Failure

For this paper our definition of a successful decision procedure will be one that generates higher expected utility before entering a problem (ex-ante). In the models we are using the output of a decision procedure directly and completely determines the action an agent will take. So if we know what action the agent's decision procedure will output when faced with the problem, we can calculate the ex-ante expected utility leaving all chance variables free to vary according the problem setup.

5.2 EDP fails on the hitchiker problem

The Evidential Decision Procedure tells the agent to maximize the expected utility of outcomes using the probabilities of the outcomes conditional on the action taken. In mathematical terms, the evidental decision procedure is:

$$EDP(s) = \arg\max_{a} \sum_{i}^{n} U(O_i) * P(\, O_i \mid a \cap s)$$

This takes a string of sense data (s), and returns an action that maximizes the expected utility of outcomes weighted by the respective probabilities of the outcomes conditional on an action and the string of sense data the agent takes in (the intersection of the action and the sense data).

What action will EDP(s) output if the agent uses it upon arriving in the city?

If the agent models their arrival in the city, then their hypothetical sense data tells the agent that Paul Ekman's decision node is set to r_y. This means that as the agent considers potential actions, there will only be two possible outcomes.

To be explicit:

$$Pr(O_1|s) = 0$$

$$Pr(O_2|s) + Pr(O_3|s) = 1$$

The probabilities of O_2 and O_3 conditional on $a_p \cap s$ are:

$$Pr(O_2|a_p \cap s) = 1$$

$$Pr(O_3|a_p \cap s) = 0$$

This corresponds to our outcome mapping, where a_p and r_y yield O_2

The probabilities of O_2 and O_3 conditional on $a_l \cap s$ are:

$$Pr(O_2|a_l \cap s) = 0$$

$$Pr(O_3|a_l \cap s) = 1$$

This again corresponds to our outcome mapping, where a_l and r_y yield O_3

So the expected utility of a_p is:

$$U(O_1) * Pr(O_1|a_p \cap s) + U(O_2) * Pr(O_2|a_p \cap s) + U(O_3) * Pr(O_3|a_p \cap s)$$

and substituting,

$$0 * 0 + 1000 * 1 + 1020 * 0 = 1000$$

While the expected utility of a_l is:

$$U(O_1) * Pr(O_1|a_l \cap s) + U(O_2) * Pr(O_2|a_l \cap s) + U(O_3) * Pr(O_3|a_l \cap s)$$

and substituting,

$$0 * 0 + 1000 * 0 + 1020 * 1 = 1020$$

Since $EU(a_l) = 1020 > EU(a_p) = 1000$, EDP will recommend action a_l "live and leave"

This means that the agent's model of its actions upon arriving in the city are that it will choose a_l "live and leave" because it is an EDP agent. This will reflect in its facial expressions. Paul Ekman will take those facial expressions $FE(a)$ and decide if the agent is lying. He will give or not give a ride based upon that.

If $FE(a_p)$ are observed, he will give a ride with $Pr(r_y) = 1$

If $FE(a_l)$ are observed, he will give a ride with $Pr(r_y) = 0.95$

We know that the agent will choose a_l because it is following the EDP. We can then compute the expected outcomes from the causal graph as

$$Pr(O_1|a_l) = Pr(r_n|a_2) = 0.05$$
$$Pr(O_1|a_l) = Pr(r_y \cap a_p|a_l) = 0$$
$$Pr(O_3|a_l) = Pr(r_y \cap a_l|a_l) = 0.95$$

So the agent's expected utility is:

$$U(O_1) * 0.05 + U(O_2) * 0 + U(O_3) * 0.95$$
$$0 * 0.05 + 0 * 0 + 1020 * 0.95 = 969$$

On what basis can we say that EDP systematically fails on this problem?

Imagine the agent had a decision procedure (we'll call it "DP" as a stand in) that output action a_p (two of which we will examine), then:

$$Pr(O_1|a_p) = 0$$
$$Pr(O_2|a_p) = 1$$
$$Pr(O_3|a_p) = 0$$

and the expected utility would be:

$$U(O_1) * 0 + U(O_2) * 1 + U(O_3) * 0 = U(O_2) * 1$$

which in this case is just

$$1000 * 1 = 1000$$

Notice that:

$$EU(DP := a_p) = 1000 > 969 = EU(DP := a_l)$$

and this the basis for my claim that agents who implement EDP will systematically lose in expected utility on the Hitchiker Problem. In fact, any agent which implements a decision procedure that recommends action a_l upon arriving in the city will lose.

5.3 CDP systematically fails on the hitchiker problem

Given the above analysis, we simply have to show that CDP returns a_l if fed a sensory string corresponding to the ride variable being fixed to r_y.

Recall that CDP is expressed as:

$$CDP(s) = \arg\max_a \sum_i^n U(O_i) * P(\, a \;\Box\!\!\rightarrow O_i \,|\, s)$$

This chooses the action that maximizes expected utility where the utility levels of the various outcomes are weighted by the probability that the action will cause the outcome to obtain.

Upon inputting the hypothetical sensory string corresponding to being in the city, the agent will look at $P(a_l \;\Box\!\!\rightarrow O_i)$ and $P(a_p \;\Box\!\!\rightarrow O_i)$.

For a_l:

$$P(a_l \;\Box\!\!\rightarrow O_1) = 0$$
$$P(a_l \;\Box\!\!\rightarrow O_2) = 0$$
$$P(a_l \;\Box\!\!\rightarrow O_3) = 1$$

For a_p:

$$P(a_p \;\Box\!\!\rightarrow O_1) = 0$$
$$P(a_p \;\Box\!\!\rightarrow O_2) = 1$$
$$P(a_p \;\Box\!\!\rightarrow O_3) = 0$$

So expected utility for a_l is

$$U(O_1) * 0 + U(O_2) * 0 + U(O_3) * 1$$
$$EU(a_l) = 0 * 0 + 0 * 1000 + 1020 * 1 = 1020$$

Wheras expected utility of a_p is

$$U(O_1) * 0 + U(O_2) * 1 + U(O_3) * 0$$
$$EU(a_l) = 0 * 0 + 1000 * 1 + 1020 * 0 = 1000$$

Since $EU(a_l) = 1020 > 1000 = EU(a_p)$ when fed the sensory string corresponding to being in the city, CDP recommends the agent do a_l.

5.4 TDP succeeds on the hitchiker problem

Our discussion of the Timeless Decision Procedure is motivated by the observation that in many problems involving prediction, the prediction must be based upon knowledge about what the output of an agent's decision procedure will be. This idea becomes particularly salient if you are thinking about agents who are computer programs and therefore potentially have a decision procedure that can be run by another program with perfect fidelity to output an action given various inputs. Such a situation has been analyzed using modal agents in the context of a one shot prisoners' dilemma, and the agent which has so far enjoyed the greatest success was inspired by Timeless Decision Theory (Barasz et al, 2014).

TDP uses a variable in the decision diagram called a logical node that represents logical uncertainty about the outcome of the agent's decision procedure (See Altair, 2013; Yudkowsky, 2010). It considers the outcomes weighted by the probabilities that, if the decision procedure returns a certain action, this will cause the outcome to obtain. Mathematically, this is:

$$TDP(s) = \arg\max_a \sum_i^n U(O_i) * P(\ulcorner TDP(s) := a \urcorner \mathrel{\Box\!\!\to} O_i \mid s)$$

On the hitchiker problem, TDP has two actions to consider in the city, a_l and a_p. The key is that it gets to use essentially the full causal graph, because it can look from the node representing the decision procedure's output, instead of the action node itself. Therefore for a_l:

$$P(\ulcorner TDP(s) := a_l \urcorner \mathrel{\Box\!\!\to} O_1 \mid s) = 0.05$$

$$P(\ulcorner TDP(s) := a_l \urcorner \mathrel{\Box\!\!\to} O_3 \mid s) = 0.95$$

Wheras for a_p,

$$P(\ulcorner TDP(s) := a_p \urcorner \mathrel{\Box\!\!\to} O_2 \mid s) = 1$$

So

$$EU(TDP(s) := a_p) = U(O_2) * 1 = 1000$$

and

$$EU(TDP := a_l) = U(O_1) * 0.05 + U(O_3) * 0.95 = 0 * 0.05 + 1020 * 0.95 = 969$$

Since

$$EU(TDP(s) := a_p) = 1000 > 969 = EU(TDP(s) := a_l)$$

EDP will return a_p. Since it returns a_p it will have an expected utility of 1000 going into the problem instead of 969.

5.5 TDP fails on the Curious Benefactor

TDP's failure on the Curious Benefactor is straightforward. Upon seeing the coinflip has come up tails, it updates on the sensory data and realizes that it is in the causal branch where there is no possibility of getting a million.

Specifically, upon receiving the sense data that the coin has come up tails, $s(T)$, it will calculate the probability of the outcomes upon outputting the two different actions as:

$$Pr(\ulcorner TDP := a_1 \urcorner \mathrel{\Box\!\!\to} O_1 | s(T)) = 1$$

and

$$Pr(\ulcorner TDP := a_2 \urcorner \mathrel{\Box\!\!\to} O_2 | s(T)) = 1$$

This corresponds to our outcome mapping, where once the coin has come up tails, the only possible outcomes remaining are either giving the \$5 and receiving nothing (O_1), or not giving the \$5 and receiving nothing (O_2).

TDP will therefore evaluate the expected utility of giving the \$5 as:

$$U(O_1) * Pr(\ulcorner TDP := a_1 \urcorner \Box\!\!\rightarrow O_1 | s(T)) = -5 * 1 = -5$$

and the expected utility of not giving the \$5 as:

$$U(O_2) * Pr(\ulcorner TDP := a_1 \urcorner \Box\!\!\rightarrow O_2 | s(T)) = 0 * 1 = 0$$

Because $EU(a_1) = -5 < 0 = EU(a_2)$, TDP will take action a_2.

The problem is that an agent whose decision procedure outputs action a_2 upon receiving s(T) has an expected ex-ante utility of 0, while an agent whose procedure outputs action a_2 has an expected ex-ante utility of 500,000-2.5.

5.6 UDP succeeds on the Curious Benefactor

Our discussion of the "Updateless" Decision Procedure is motivated by a desire to have agents avoid losing out in ex-ante expected utility by updating on sensory evidence. This sensory updating appears to be the problem in the Curious Benefactor dilemma, and as we would therefore expect, UDP handles it well.

UDP tells us to set a mapping from sensory data to actions. It then computes the utility of the outcomes weighted by the probability that the outcome will obtain causally if the agent has that mapping.

Mathematically this is,

$$UDP(s) = \arg\max_f \sum_i^n U(O_i) * P(\ulcorner UDP := f : s \mapsto a \urcorner \Box\!\!\rightarrow O_i)$$

which finds a function f that maps sense data to actions in a way that maximizes the expected utility which is defined by weighting each outcome's utility level by the probability that the outcome is caused by the agent's use of that mapping

We will use $s(T)$ to denote the sense string corresponding to observation of world T (tails), and $s(H)$ to denote the sense string corresponding to observation of world H (heads).

The agent does not have to take any action for the situation where the coin has come up heads, so $s(H) \mapsto \emptyset$

There are then two options for UDP to consider. Recall that a_1 is giving the \$5 when asked while a_2 is not giving it.

$$f_1 : s(T) \mapsto a_1$$

and

$$f_2 : s(T) \mapsto a_2$$

Let's first look at $f_2 : s(T) \mapsto a_2$, else no action ($s(H) \mapsto \emptyset$). If the agent senses T it will take action a_2, and this will land it in outcome O_2. To be specific,

$$Pr(\ulcorner UDP := f_2 : s(T) \mapsto a_2 \urcorner \Box\!\!\rightarrow O_1 | T) = 0$$

because a_1 and a_2 are exclusive.

$$Pr(\ulcorner UDP := f_2 : s(T) \mapsto a_2 \urcorner \Box\!\!\rightarrow O_2 | T) = 1$$

because we are in situation tails and the action is a_2. So

$$Pr(\ulcorner UDP := f_2 : s(T) \mapsto a_2 \urcorner \mathrel{\Box\!\!\rightarrow} O_2) = Pr(\ulcorner f_2 : s(T) \mapsto a_2 \urcorner \mathrel{\Box\!\!\rightarrow} O_2|T) * Pr(T) = 1*0.5 = 0.5$$

Futhermore,

$$Pr(\ulcorner UDP := f_2 : s(T) \mapsto a_2 \urcorner \mathrel{\Box\!\!\rightarrow} O_3|H) = 1$$

$$Pr(\ulcorner UDP := f_2 : s(T) \mapsto a_2 \urcorner \mathrel{\Box\!\!\rightarrow} O_4|H) = 0$$

because when the coin comes up H (heads), the benefactor is simulating what the agent would do if situation T (tails) obtained.

Note that:

$$Pr(\ulcorner UDP := f_2 : s(T) \mapsto a_2 \urcorner \mathrel{\Box\!\!\rightarrow} O_3) = Pr(\ulcorner UDP := f : s(T) \mapsto a_2 \urcorner \mathrel{\Box\!\!\rightarrow} O_3|H) * Pr(H) = 1*0.5 = 0.5$$

The expected utility of f_2, $EU(f_2)$ is then:

$$EU(f_2) = U(O_2) * Pr(\ulcorner UDP := f_2 : s(T) \mapsto a_2 \urcorner \mathrel{\Box\!\!\rightarrow} O_2) + U(O_3) * Pr(\ulcorner UDP := f_2 : s(T) \mapsto a_2 \urcorner \mathrel{\Box\!\!\rightarrow} O_3)$$

which substituting, yields:

$$EU(f_2) = 0 * 0.5 + 0 * 0.5 = 0$$

Wheras if we look at $f_1 : s(T) \mapsto a_1$, else no action ($s(H) \mapsto \varnothing$) we will see that:

$$Pr(\ulcorner UDP := f_1 : s(T) \mapsto a_1 \urcorner \mathrel{\Box\!\!\rightarrow} O_1|T) = 1$$

because the agent takes a_1 and the coin flip is tails in the situation where the agent is called upon to decide while,

$$Pr(\ulcorner UDP := f_1 : s(T) \mapsto a_1 \urcorner \mathrel{\Box\!\!\rightarrow} O_2|T) = 0$$

because a_1 and a_2 are exclusive. Note that:

$$Pr(\ulcorner UDP := f_1 : s(T) \mapsto a_1 \urcorner \mathrel{\Box\!\!\rightarrow} O_4) = Pr(\ulcorner UDP := f_1 : s(T) \mapsto a_1 \urcorner \mathrel{\Box\!\!\rightarrow} O_4|H) * Pr(H) = 1*0.5 = 0.5$$

and this means that the expected utility of f_1 would be:

$$EU(f_1) = U(O_1) * Pr(\ulcorner UDP := f_1 : s(T) \mapsto a_1 \urcorner \mathrel{\Box\!\!\rightarrow} O_1) + U(O_4) * Pr(\ulcorner UDP := f_1 : s(T) \mapsto a_1 \urcorner \mathrel{\Box\!\!\rightarrow} O_4)$$

so substituting,

$$EU(f_1) = -5 * 0.5 + 1{,}000{,}000 * 0.5 = 500{,}000 - 2.5$$

Since $500{,}000 - 2.5 > 0$, UDP will recommend that the agent use f_1 and take action a_1, giving the \$5 if asked.

As noted above in the analysis of TDP, the agent will take a_1 and have an expected utility going into the problem of $500{,}000 - 2.5$ instead of 0.

6 Conclusion

6.1 TDP dominates CDP

We have seen that both the evidential and the causal decision procedures yield systematically lower utility on the hitchiker problem, while the timeless decision procedure succeeds. The fact that both the causal and evidential procedures obtain this result indicates that the problem is not specifically biased against a particular decision procedure. It then becomes necessary to consider whether timeless decision theory will properly handle any problems that CDP or EDP do well on in order to establish dominance.

Let us first consider CDP. CDP will perform well on any problems where the decision procedure is irrelevant for computing the outcome of the problem.[9] Only the action actually taken by the agent is needed. In these cases TDP works just as well as CDP, because it will simply choose to output whatever action CDP would have output.

Let us next consider EDP. EDP appears to perform well when there are no common (or confounding) causes of the action and the outcomes, other than the agent's decision procedure itself. TDP will perform well on all of these situations.

From this I conclude that TDP dominates CDP and EDP.

If it can be shown that there is a unbiased problem CDP performs well on that TDP does not, then I am wrong that TDP dominates CDP. Likewise, if it can be shown there is an unbiased problem EDP performs well on that TDP does not, I will be wrong that TDP dominates EDP.

If such a problem can be found, then we will be in a situation where there are some problems TDP does better on, and other problems which EDP or CDP do better on. The choice of which procedure to use would have to be made based on the expectation of the problem pool the agent will encounter.[10]

6.2 UDP dominates TDP

We have seen that UDP yields higher expected utility on the Curious Benefactor problem than TDP. While we did not show it, CDP and EDP would also fail the Curious Benefactor problem, indicating that it is unbiased.

We must then show that UDP succeeds in all problems where TDP will succeed. TDP appears to succeed in situations where there aren't counterfactual branches with different sensory inputs. In these cases, UDP will only have one piece of sense data to consider. It will then choose a mapping from that sense data to the same action that TDP would have chosen. So it will succeed in all the situations where TDP would succeed. I conclude that UDP dominates TDP, and by extension, CDP and EDP.

If it can be shown that there is an unbiased problem that TDP performs well on that UDP does not, then I am wrong that UDP dominates TDP.

[9] Or more accurately the combination of discrepancy in utility levels and probability of effect is sufficiently small.

[10] I believe that if we choose to limit our choice of procedure to one between CDP and EDP, then this is also the situation we find ourselves in. Neither has problem class dominance over the other by our ex-ante success criterion, and I believe this may help in part to explain why there has been so much debate about which is superior.

6.3 Can we do better?

It would be exciting to find a problem that UDP performs badly on, and then a decision procedure that handles both it and all the problems that UDP performs well on. The blackmailer problem [11] may well be a source of such an insight.

It would also be exciting to find a procedure that somehow even handles problems that are biased against it, allowing us to drop the qualification of excluding biased problems from our definition of dominance.

6.4 A different perspective

It is no small leap to model agents who can predict another agent's action by simulating its decision procedure. It is also no small leap to imagine choosing a decision procedure for an agent, particularly if we imagine the agent is choosing a decision procedure for itself. TDP and UDP are attempts to not only effectively model these predictive dilemmas, but to process them in a way that is preferable for an agent looking to do as well as possible in the future. I have presented arguments that they succeed on this count, and that agents who adopt UDP can expect to do at least as well, if not better, than agents who adopt TDP. Likewise, agents who adopt TDP can expect to do at least as well, if not better, as agents who adopt CDP or EDP. I also believe that the analysis of decision procedures with the perspective of problem class dominance opens up exciting new possibilites for thinking about decision theory not only in the context of predictive dilemmas but in any category of problems we may face in the future.

[11] Yudkowsky via Armstrong, 2010

7 Toxoplasmosis Gondii (Appendix)

7.1 The Problem

There is a single celled parasitic organism called toxoplasmosis gondii which is normally found in cats. When rats are infected with it, it has been known to cause attraction to cats and cat urine. Humans can also be infected, and it has been shown to be correlated with various psychiatric disorders in humans. There has been some speculation that it could potentially cause humans to exhibit increased attraction to cats. Suppose we live in a world where this speculation has been solidly verified, and the correlation is 80% between toxoplasmosis infection and liking cats. Suppose also that a particular cat has been verified to be free of the parasite and cannot possibly infect you. The action an agent has to consider is whether to enjoy petting the cat (Altair refers to this as ”adore cats”), keeping in mind that petting the cat is fun and gains utility, but 80% of the people who enjoy petting cats have toxoplasmosis. [12]

7.2 The Trouble with Formalizing The Toxoplasmosis problem

A tentative causal diagram (Diagram 1) of the problem is included. This diagram appears reasonable if we assume that Toxoplasmosis Gondii is stepping in to modify the agent’s action after the decision procedure has already outputted its decision (Think of how I might knock your hand out of the way after you’ve decided to reach for a glass of water). In this case, both CDP and TDP handle the problem easily while basic EDP does not.

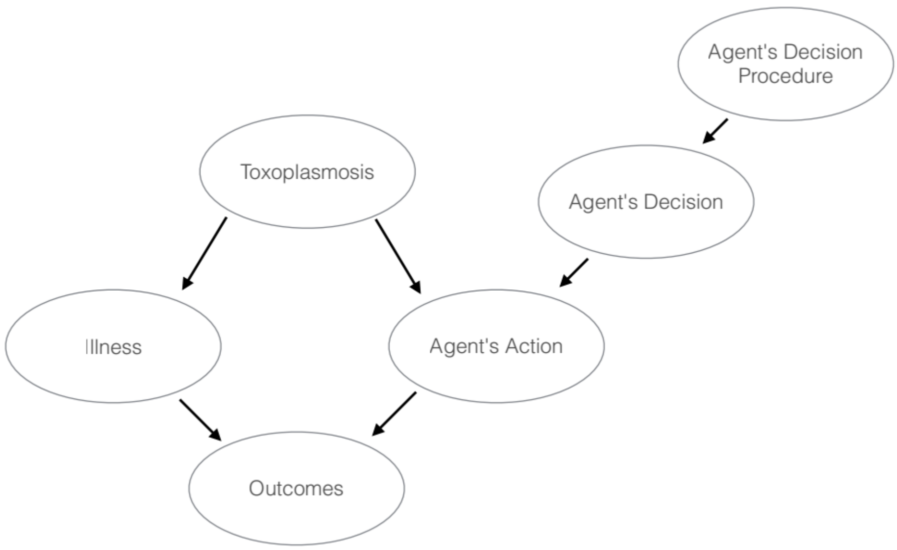

Figure 3: Diagram 1 of the Toxoplasmosis Problem.

[12]Note that this problem is designed to be a modern version of the smoking gene problem with assumptions that do not contradict current scientific knowledge.

The real issue comes when we consider that the problem setup seems to indicate that the toxoplasmosis is having an effect on the agent's decision itself, as seen in diagram 2. If this is the case and the decision is a direct output of the decision procedure, then this means that toxoplasmosis must be influencing the decision procedure itself. As we have been specifying decision procedures throughout this paper, this means that it likely either acts through the utility function or through the specification of outcome probabilities (or both). The problem setup seems to suggest that it is acting through the utility function. We could use a "tickle defense" to allow the procedures to know what their utility function is, but that seems to be missing the real meat of the issue.

How can toxoplasmosis uniformly create this supposed 80% correlation using only the utility function if it doesn't even get to know beforehand what decision procedures the agents are using? We might assume that there is a certain population implementing some mixture of decision procedures that has created the 80% correlation, but that does little to help us know what is actually going on within the decision procedures themselves.

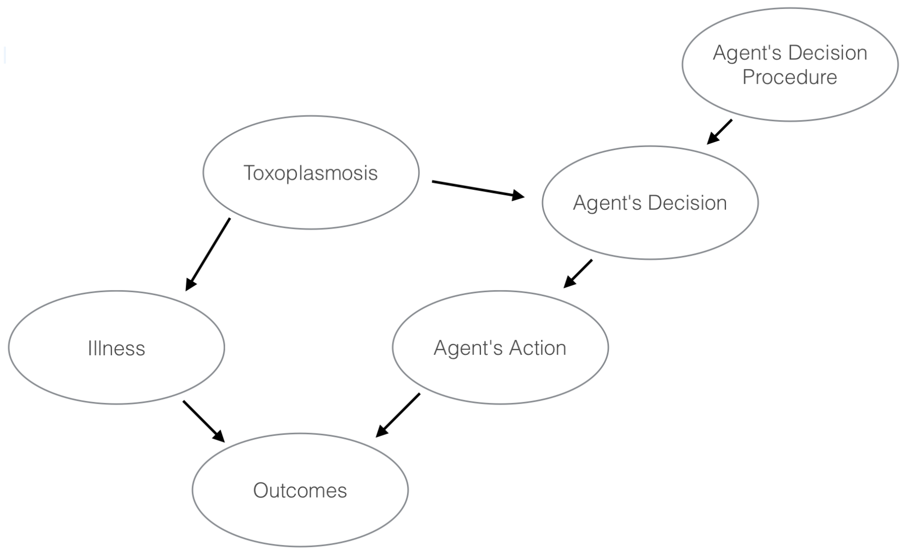

Figure 4: Diagram 2 of the Toxoplasmosis Problem.

References

[1] Altair, Alex. "A Comparison of Decision Algorithms on Newcomblike Problems." 2013.

[2] Armendt, Brad. 2011. "Causal Decision Theory." Handbook of Formal Philosophy.

[3] Armstrong, Stuart. March 10, 2010. "The Blackmail Equation." LessWrong. http://lesswrong.com/lw/1vv

[4] Barasz, Mihaly, Paul Christiano, Benja Fallenstein, Marcello Herreshoff, Patrick LaVictoire, and Eliezer Yudkowsky. ″Robust Cooperation in the Prisoner's Dilemma: Program Equilibrium via Provability Logic.″ arXiv preprint arXiv:1401.5577 (2014).

[5] Bostrom, Nick. 2001. "The Meta-Newcomb Problem." Analysis 61 (4): 309–310. doi:10.1093/analys/ 61.4.309

[6] Drescher, Gary L. 2006. Good and Real: Demystifying Paradoxes from Physics to Ethics. Bradford Books. Cambridge, MA: MIT Press.

[7] Egan, Andy. 2007. "Some Counterexamples to Causal Decision Theory." Philosophical Review 116 (1): 93–114. doi:10.1215/00318108-2006-023.

[8] Gibbard, Allan, and William L. Harper. 1978. "Counterfactuals and Two Kinds of Expected Utility: Theoretical Foundations." In Foundations and Applications of Decision Theory, edited by Clifford Alan Hooker, James J. Leach, and Edward F. McClennen. Vol. 1. The Western Ontario Series in Philosophy of Science 13. Boston. D Reidel.

[9] Hitchcock, Christopher (1996) "Causal Decision Theory and Decision-Theoretic Causation." Noûs 30 (4): 508-526.

[10] Jeffrey, Richard C. 1983. The Logic of Decision. 2nd ed. Chicago: Chicago University Press

[11] Lewis, David. 1981.'Why Ain'cha Rich?' Noûs 15 (3): 377-380

[12] Nozick, Robert. 1969. "Newcomb's Problem and Two Principles of Choice." In Essays in Honor of Carl G. Hempel: A Tribute on the Occasion of His Sixty-Fifth Birthday, edited by Nicholas Rescher, 114–146. Synthese Library 24. Dordrecht, The Netherlands: D. Reidel.

[13] Parfit, Derek. 1986. Reasons and Persons. New York: Oxford University Press. doi:10.1093/019824908X. 001.0001.

[14] Spohn, Wolfgang. 2003. "Dependency Equilibria and the Causal Structure of Decision and Game Situations." Homo Oeconomicus. 20:195-255.

[15] Stalnaker, Robert (1972) "Letter to David Lewis." In W. Harper, et al. (eds.), Ifs, 151-152, Dordrecht: Reidel.

[16] Wei Dai. 2009. ″Towards a New Decision Theory.″ LessWrong. Accessed 2014-4-14. http://lesswrong.com/lw/15m/

[17] Yudkowsky, Eliezer. 2010. Timeless Decision Theory. Machine Intelligence Research Institute, San Francisco, CA. http://intelligence.org/files/TDT.pdf

[18] Yudkowsky, Eliezer. March 9, 2014. Historical comment on ″You have just been Counterfactually Mugged.″ http://lesswrong.com/lw/162