Tractable Probabilistic Models for Moral Responsibility

Lewis Hammond*
University of Oxford
Oxford, OX1 3QD
lewis.hammond@cs.ox.ac.uk

Vaishak Belle
University of Edinburgh Alan Turing Institute
Edinburgh, EH8 9AB London, NW1 2DB
vaishak@ed.ac.uk

Abstract

Moral responsibility is a major concern in autonomous systems, with applications ranging from self-driving cars to kidney exchanges. Although there have been recent attempts to formalise responsibility and blame, among similar notions, the problem of learning within these formalisms has been unaddressed. From the viewpoint of such systems, the urgent questions are: (a) How can models of moral scenarios and blameworthiness be extracted and learnt automatically from data? (b) How can judgements be computed effectively and efficiently, given the split-second decision points faced by some systems? By building on constrained tractable probabilistic learning, we propose a learning framework for inducing models of such scenarios automatically from data and reasoning tractably from them. We report on experiments that compare our system with human judgement in three domains: lung cancer staging, teamwork management, and trolley problems.

1 Introduction

Moral responsibility is a major concern in autonomous systems. In applications ranging from self-driving cars to kidney exchanges [12], contextualising and enabling judgements of morality and blame is becoming a difficult challenge, owing in part to the philosophically vexing nature of these notions. Within the context of interactions between humans and autonomous systems, the concept of blameworthiness has been argued as being critical to effective collaboration, decision-making, and to our thoughts about morality in general [25, 18]. Whilst there have been many formal definitions of blame and moral responsibility put forward by philosophers, lawyers, and psychologists over the decades, there have been relatively few that admit concrete computational implementations.

The limited number of implementations that do exist are typically based on hand-crafted or deterministic rules that encode ethical principles, which can result in such systems being brittle, lacking the flexibility and scalability offered by tractable probabilistic models and machine learning more generally [3, 14]. With that said, many moral decision-making scenarios can be concisely captured using a logical representation, and it also seems intuitively plausible that logical constraints may arise in many such situations (e.g. it is forbidden to kill any human being). Thus, a framework that incorporates *both* of these traditionally separate paradigms would seem most appropriate and desirable in helping to facilitate "provably moral AI" [40]. This corresponds precisely to the hybrid between the top-down (symbolic) and bottom-up (sub-symbolic) approaches discussed by Allen et al. [2], of which (as far as we are aware) our implementation represents the first concrete instance.

We propose a *learning framework for inducing models of moral scenarios and blameworthiness automatically from data, and reasoning tractably from them*. The framework leverages the tractable learning paradigm [39, 11, 26], which can induce both high- and low- tree width graphical models

*Work conducted while at the University of Edinburgh.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

with latent variables, and thus realises a deep probabilistic architecture. We remark that *we do not motivate any new definitions for moral responsibility, but show how an existing model can be embedded in our learning framework*. We suspect it should be possible to analogously embed other definitions from the literature too. We then study the computational features of this framework. Finally, we report on experiments regarding the alignment between automated and human judgements of moral decision-making in three illustrative domains: lung cancer staging, teamwork management, and trolley problems.

2 Preliminaries

2.1 Blameworthiness

We use the word *blameworthiness* to capture an important part of what can more broadly be described as moral responsibility, and consider a set of definitions (taken directly from the original work, with slight changes in notation for the sake of clarity and conciseness) put forward by Halpern and Kleiman-Weiner [19] (henceforth HK). In HK, environments are modelled in terms of variables and structural equations relating their values [20]. More formally, the variables are partitioned into *exogenous* variables $\mathcal{X}$ external to the model in question, and *endogenous* variables $\mathcal{V}$ that are internal to the model and whose values are determined by those of the exogenous variables. A range function $\mathcal{R}$ maps every variable to the set of possible values it may take. In any model, there exists one structural equation $F_V : \times_{Y \in \mathcal{X} \cup \mathcal{V} \setminus \{V\}} \mathcal{R}(Y) \to \mathcal{R}(V)$ for each $V \in \mathcal{V}$.

Definition 1. A **causal model** M is a pair $(\mathcal{S}, \mathcal{F})$ where $\mathcal{S}$ is a **signature** $(\mathcal{X}, \mathcal{V}, \mathcal{R})$ and $\mathcal{F}$ is a set of **modifiable structural equations** $\{F_V : V \in \mathcal{V}\}$. A **causal setting** is a pair $(M, \mathbf{X})$ where $\mathbf{X} \in \times_{X \in \mathcal{X}} \mathcal{R}(X)$ is a **context**.

In general we denote an assignment of values to variables in a set $\mathcal{Y}$ as $\mathbf{Y}$. Following HK, we restrict our considerations to *recursive* models, in which, given a context $\mathbf{X}$, the values of all variables in $\mathcal{V}$ are uniquely determined.

Definition 2. A **primitive event** is an equation of the form $V = v$ for some $V \in \mathcal{V}$, $v \in \mathcal{R}(V)$. A **causal formula** is denoted $[\mathcal{Y} \leftarrow \mathbf{Y}]\varphi$ where $\mathcal{Y} \subseteq \mathcal{V}$ and φ is a Boolean formula of primitive events. This says that if the variables in $\mathcal{Y}$ were set to values $\mathbf{Y}$ (i.e. by **intervention**) then φ would hold. For a causal formula ψ we write $(M, \mathbf{X}) \vDash \psi$ if ψ is satisfied in causal setting $(M, \mathbf{X})$.

An agent's epistemic state is given by $(\Pr, \mathcal{K}, \mathrm{U})$ where $\mathcal{K}$ is a set of causal settings, Pr is a probability distribution over this set, and U is utility function $\mathrm{U} : \mathcal{W} \to \mathbb{R}_{\geq 0}$ on the set of worlds, where a world $w \in \mathcal{W}$ is defined as a setting of values to all variables in $\mathcal{V}$. $w_{M,\mathbf{X}}$ denotes the unique world determined by the causal setting $(M, \mathbf{X})$.

Definition 3. We define **how much more likely it is that φ will result from performing an action a than from performing action a'** using:

$$\delta_{a,a',\varphi} = max\left(\left[\sum_{(M,\mathbf{X}) \in [\![[A \leftarrow a]\varphi]\!]} \Pr(M, \mathbf{X}) - \sum_{(M,\mathbf{X}) \in [\![[A \leftarrow a']\varphi]\!]} \Pr(M, \mathbf{X})\right], 0\right)$$

where $A \in \mathcal{V}$ is a variable identified in order to capture an action of the agent and $[\![\psi]\!] = \{(M, \mathbf{X}) \in \mathcal{K} : (M, \mathbf{X}) \vDash \psi\}$ is the set of causal settings in which ψ (a causal formula) is satisfied.

The costs of actions are measured with respect to a set of outcome variables $O \subseteq \mathcal{V}$ whose values are determined by an assignment to all other variables. In a given causal setting $(M, \mathbf{X})$, $\mathbf{O}_{A \leftarrow a}$ denotes the setting of the outcome variables when action a is performed and $w_{M,O \leftarrow \mathbf{O}_{A \leftarrow a}, \mathbf{X}}$ denotes the corresponding world.

Definition 4. The (expected) **cost of a relative to O** is:

$$c(a) = \sum_{(M,\mathbf{X}) \in \mathcal{K}} \Pr(M, \mathbf{X})\left[\mathrm{U}(w_{M,\mathbf{X}}) - \mathrm{U}(w_{M,O \leftarrow \mathbf{O}_{A \leftarrow a}, \mathbf{X}})\right]$$

Finally, HK introduce one last quantity N to measure how important the costs of actions are when attributing blame (this varies according to the scenario). Specifically, as $N \to \infty$ then $db_N(a, a', \varphi) \to \delta_{a,a',\varphi}$ and thus the less we care about cost. Note that blame is assumed to be non-negative and so it is required that $N > max_{a \in A} c(a)$.

Definition 5. The **degree of blameworthiness of** a **for** φ **relative to** a' (given c and N) is:

$$db_N(a, a', \varphi) = \delta_{a,a',\varphi} \frac{N - max(c(a') - c(a), 0)}{N}$$

The overall **degree of blameworthiness of** a **for** φ is then:

$$db_N(a, \varphi) = max_{a' \in \mathcal{R}(A) \setminus \{a\}} db_N(a, a', \varphi))$$

For reasons of space we omit an example here, but include several when reporting the results of our experiments. For further examples and discussions, we refer the reader to HK [19].

2.2 PSDDs

Since, in general, probabilistic inference is intractable [6], tractable learning has emerged as a recent paradigm where one attempts to learn classes of Arithmetic Circuits (ACs), for which inference is tractable [16, 26]. In particular, we use Probabilistic Sentential Decision Diagrams (PSDDs) [26] which are tractable representations of a probability distribution over a propositional logic theory (a set of sentences in propositional logic) represented by a Sentential Decision Diagram (SDD) [13]. PSDDs represent a complete, canonical class with respect to distributional representation, but can also be naturally learnt with the inclusion of logical constraints or background knowledge.

Space precludes us from discussing SDDs and PSDDs in detail, but the main idea behind SDDs is to factor the theory recursively as a binary tree: *terminal nodes* are either 1 or 0, and the *decision nodes* are of the form $(p_1, s_1), \ldots, (p_k, s_k)$ where *primes* $p_1, \ldots, p_k$ are SDDs corresponding to the left branch, *subs* $s_1, \ldots, s_k$ are SDDs corresponding to the right branch, and $p_1, \ldots, p_k$ form a *partition* (the primes are consistent, mutually exclusive, and their disjunction $p_1 \vee ... \vee p_k$ is valid). In PSDDs, each prime p_i in a decision node $(p_1, s_1), \ldots, (p_k, s_k)$ is associated with a non-negative parameter θ_i such that $\sum_{i=1}^{k} \theta_i = 1$ and $\theta_i = 0$ if and only if $s_i = \bot$. Each terminal node also has a a parameter θ such that $0 < \theta < 1$, and together these parameters can be used to capture probability distributions.

Most significantly, probabilistic queries, such as conditionals and marginals, can be computed in time linear in the size of the model. PSDDs can be learnt from data [29], and the ability to encode logical constraints into the model directly enforces sparsity which in turn can lead to increased accuracy and decreased size. In our setting, we can draw parallels between these logical constraints and deontological ethical principles (e.g. it is forbidden to kill another human being), and between learnt distributions over decision-making scenarios (encoding preferences) and the utility functions used in consequentialist ethical theories (where the moral value of an action depends on its consequences).

3 Blameworthiness Via PSDDs

We aim to leverage the learning of PSDDs, their tractable query interface, and their ability to handle domain constraints for inducing models of moral scenarios.[2] This is made possible by means of an embedding that we sketch below, while also discussing assumptions and choices. At the outset, we reiterate that we do not introduce new definitions here, but show how an existing one, that of HK, can be embedded within a learning framework. Where there is any chance of ambiguity we denote the original definitions with a superscript ^{HK}.

3.1 Variables

We distinguish between scenarios in which we do and do not model *outcome variables*; in the latter case we have $\mathcal{V} = \mathcal{D} = \mathcal{O}$ (this does not affect the notation in our later definitions, however). This is because we do not assume that outcomes can always be recorded, and in some scenarios it makes sense to think of decisions as an end in themselves.

Our *range function* $\mathcal{R}$ is defined by the scenario we model, but in practice we one-hot encode the variables and so the range of each is simply $\{0, 1\}$. A subset (possibly empty) of the *structural equations*

[2] Our technical development can leverage both parameter and (possibly partial) structure learning for PSDDs. Of course, learning causal models is a challenging problem [1], and in this regard, probabilistic structure learning is not assumed to be a recipe for causal discovery in general [36]. Rather, under the assumptions discussed later, we are able to use our probabilistic model for causal reasoning.

in $\mathcal{F}$ is implicitly encoded within the structure of the SDD underlying the PSDD, corresponding to the logical constraints that remain true in every causal model M. The remaining equations are those that vary depending on the causal model. Each possible assignment $\mathbf{V}$ given $\mathbf{X}$ corresponds to a set of structural equations that combine with those encoded by the SDD to determine the values of the variables in $\mathcal{V}$ given $\mathbf{X}$, as we make the trivial assumption that all parentless variables are considered exogenous. The PSDD then corresponds to the probability distribution Pr over $\mathcal{K}$.

Our critical assumption here is that the *signature* $\mathcal{S} = (\mathcal{X}, \mathcal{V}, \mathcal{R})$ (the variables and the values they may take) remains the same in all models, although the structural equations $\mathcal{F}$ (the ways in which said variables are related) may vary. This is necessary both for our theoretical embedding and learning PSDDs from decision-making data (where data points measure the same variables each time).

3.2 Probabilities

Thus, our *distribution* $\Pr : \times_{Y \in \mathcal{X} \cup \mathcal{D} \cup \mathcal{O}} \mathcal{R}(Y) \rightarrow [0,1]$, represented as PSDD, ranges over assignments to variables instead of $\mathcal{K}$. As a slight abuse of notation we write $\Pr(\mathbf{X}, \mathbf{D}, \mathbf{O})$. The key observation needed to translate between these two distributions (we denote the original as $\Pr^{HK}$), which relies on our assumption above, is that each set of structural equations $\mathcal{F}$ together with a context $\mathbf{X}$ deterministically leads to a unique, complete assignment $\mathbf{V}$ of the endogenous variables, which we write (abusing notation again) as $(\mathcal{F}, \mathbf{X}) \models \mathbf{V}$, though there may be many such sets of equations that lead to the same assignment. Hence, for any context $\mathbf{X}$ and any assignment $\mathbf{Y}$ for $\mathcal{Y} \subseteq \mathcal{V}$ we have:

$$\Pr(\mathbf{X}, \mathbf{Y}) = \sum_{M:(M,\mathbf{X})\models\mathbf{Y}} \Pr^{HK}(M, \mathbf{X}) = \sum_{\mathcal{F}:((\mathcal{S},\mathcal{F}),\mathbf{X})\models\mathbf{Y}} \Pr^{HK}((\mathcal{S}, \mathcal{F}), \mathbf{X}) = \sum_{\mathcal{F}:(\mathcal{F},\mathbf{X})\models\mathbf{Y}} \Pr^{HK}(\mathcal{F}, \mathbf{X})$$

Given our assumptions and observations described above, the following proposition is immediate.

Proposition 1. *Let* $\Pr^{HK}$ *be a probability distribution over a set of causal settings* $\mathcal{K}$*, and assume that the signature* $\mathcal{S} = (\mathcal{X}, \mathcal{V}, \mathcal{R})$ *in each causal setting* $M = (\mathcal{S}, \mathcal{F})$ *remains fixed. Then there exists a PSDD* P *representing a distribution* Pr *over the variables in* $\mathcal{X}$ *and* $\mathcal{V}$ *such that for any context* $\mathbf{X}$*, the joint probability of* $\mathbf{Y}$ *also occurring (where* $\mathcal{Y} \subseteq \mathcal{V}$*) is the same under both* $\Pr^{HK}$ *and* Pr*.*

We view a Boolean formula of primitive events (possibly resulting from decision A) as a function $\varphi : \times_{Y \in \mathcal{O} \cup \mathcal{D} \setminus \{A\}} \mathcal{R}(Y) \rightarrow \{0,1\}$ that returns 1 if the original formula is satisfied by the assignment, or 0 otherwise. We write $\mathbf{D}_{\setminus a}$ for a general vector of values over $\mathcal{D} \setminus \{A\}$, and hence $\varphi(\mathbf{D}_{\setminus a}, \mathbf{O})$. Here, the probability of φ occurring given that action a is performed (i.e. conditioning on *intervention*) $\sum_{(M,\mathbf{X})\in[\![[A\leftarrow a]\varphi]\!]} \Pr(M, \mathbf{X})$ given by HK can also be written as $\Pr(\varphi | do(a))$. In general, it is not the case that $\Pr(\varphi | do(a)) = \Pr(\varphi | a)$, but by assuming that the direct causes of action a are captured by the context $\mathbf{X}$ and that the other decisions and outcomes $\mathbf{D}_{\setminus a}$ and $\mathbf{O}$ are in turn caused by $\mathbf{X}$ and a we may use the *back-door* criterion [37] with $\mathcal{X}$ as a *sufficient set* to write:

$$\Pr(\mathbf{D}_{\setminus a}, \mathbf{O} | do(a)) = \sum_{\mathbf{X}} \Pr(\mathbf{D}_{\setminus a}, \mathbf{O} | a, \mathbf{X}) \Pr(\mathbf{X})$$

and thus may use $\sum_{\mathbf{D}_{\setminus a}, \mathbf{O}, \mathbf{X}} \varphi(\mathbf{D}_{\setminus a}, \mathbf{O}) \Pr(\mathbf{D}_{\setminus a}, \mathbf{O} | a, \mathbf{X}) \Pr(\mathbf{X})$ for $\Pr(\varphi | do(a))$. In order not to re-learn a separate model for each scenario we also allow the user of our system the option of specifying a current, alternative distribution over contexts $\Pr'(\mathbf{X})$, which may replace $\Pr(\mathbf{X})$ in each summand.

3.3 Utilities

We now consider our *utility function* U, the output of which we assume is normalised to the range $[0,1]$.[3] We avoid unnecessary extra notation by defining the utility function in terms of $\mathbf{X}$, $\mathbf{D}$, and $\mathbf{O} = (O_1, ..., O_n)$ instead of worlds w. In our implementation we allow the user to input an existing utility function or to learn one from data. In the latter case the user further specifies whether or not the function should be context-relative, i.e. whether we have $\mathrm{U}(\mathbf{O})$ or $\mathrm{U}(\mathbf{O}; \mathbf{X})$ (our notation) as, in some cases, how good a certain outcome $\mathbf{O}$ is depends on the context $\mathbf{X}$. Similarly, the user also decides whether the function should be linear in the outcome variables, in which case the final utility is $\mathrm{U}(\mathbf{O}) = \sum_i \mathrm{U}_i(O_i)$ or $\mathrm{U}(\mathbf{O}; \mathbf{X}) = \sum_i \mathrm{U}_i(O_i; \mathbf{X})$ respectively (where we assume that each

[3] This has no effect on our calculations as we only use cardinal utility functions with bounded ranges, which are invariant to positive affine transformation.

$\mathrm{U}_i(O_i;\mathbf{X}), \mathrm{U}_i(O_i) \geq 0$). Here the utility function is simply a vector of weights and the total utility of an outcome is the dot product of this vector with the vector of outcome variables.

When learning utility functions, the key assumption we make (before normalisation) is that *the probability of a certain decision being made given a context is proportional to some function of the expected utility of that decision in the context*, i.e. $\Pr(\mathbf{D}|\mathbf{X}) \propto f(\sum_{\mathbf{O}} \mathrm{U}(\mathbf{O}) \Pr(\mathbf{O}|\mathbf{D},\mathbf{X}))$. Note that here a decision is a general assignment $\mathbf{D}$, not a single action a, and $\mathrm{U}(\mathbf{O})$ may be context-relative and/or linear in the outcome variables. In our implemented demonstration system we make the simplifying assumption that f is the identity function, however this is by no means necessary. In general we may choose any invertible function f (on the range $[0,1]$) and simply apply f^{-1} to each datum $\Pr(\mathbf{D}|\mathbf{X})$ before fitting our utility function. In general we should expect f to be a positive monotonic transformation with non-negative range so as to preserve the ordinality of utilities. For example, using $f(x) = \exp(x) - 1$ allows us to capture (a slightly modified version of) the commonly-used *Logistic Quantal Response* model of bounded rationality in which the likelihood of a certain decision is proportional to the exponential of the resulting expected utility [30].

This proportionality assumption is critical to the learning procedure in our implementation, however we believe it is in fact relatively uncontroversial, and can be restated as the simple principle that an agent is more likely to choose a decision that leads to a higher expected utility than one that leads to a lower expected utility. Of course decisions are not always representative of utility functions (consider a smoker who wishes to quit but cannot due to their addiction), and attempting to learn the preferences of fallible, inconsistent agents such as humans is an interesting, difficult problem. While outside the scope of our current work, we refer the reader to Evans et al. for a recent discussion [15].

3.4 Costs and Blameworthiness

We also adapt the *cost function* given in HK, denoted c^{HK}. As actions do not deterministically lead to outcomes in our work, we cannot use $\mathbf{O}_{A \leftarrow a}$ to represent the specific outcome when decision a is made (in some context). For our purposes it suffices to use $c(a) = -\sum_{\mathbf{O},\mathbf{X}} \mathrm{U}(\mathbf{O};\mathbf{X}) \Pr(\mathbf{O}|a,\mathbf{X}) \Pr(\mathbf{X})$ or $c(a) = -\sum_{\mathbf{O},\mathbf{X}} \mathrm{U}(\mathbf{O}) \Pr(\mathbf{O}|a,\mathbf{X}) \Pr(\mathbf{X})$, depending on whether U is context-relative or not. This is simply the negative expected utility over all contexts, conditioning by intervention on decision $A \leftarrow a$. Using our conversion between $\Pr^{HK}$ and Pr, the *back-door* criterion [37], and our assumption that action a is not caused by the other endogenous variables (i.e. $\mathcal{X}$ is a *sufficient set* for A), it is a straightforward exercise in algebraic manipulation to show the following proposition.

Proposition 2. *Let c^{HK} be a cost function determined using a distribution $\Pr^{HK}$ and utility function U. Then, given an equivalent distribution* Pr *(via the assumptions and result of Proposition 1) and the assumption that $\mathcal{X}$ forms a sufficient set for any action variable A, the cost function c determined using* Pr *and U is such that for any values a, a' of A: $c(a') - c(a) = c^{HK}(a') - c^{HK}(a)$.*

Again, $\Pr(\mathbf{X})$ can also be replaced by some other distribution $\Pr'(\mathbf{X})$ so that the current model can be re-used in different scenarios. Given $\delta_{a,a',\varphi}$ and c, both $db_N(a,a',\varphi)$ and $db_N(a,\varphi)$ are computed as in HK, although we instead require that $N > -min_{a \in A} c(a)$ (the equivalence of this condition to the one in HK is an easy exercise). With this the embedding is complete.

Proposition 3. *Let* Pr *and c be equivalents of $\Pr^{HK}$ and c^{HK} under the assumptions and results described in Propositions 1 and 2. Then for any values a, a' of any action variable $A \in \mathcal{V}$, for any Boolean formula φ, and any valid measure of cost importance N, the values of $\delta_{a,a',\varphi}$, $db_N(a,a',\varphi)$, and $db_N(a,\varphi)$ are the same in our embedding as in HK.*

4 Experiments and Results

Details of our implementation can be found in Appendix A, with associated complexity results in Appendix B. The packaged version (including full documentation) will be made available online as part of an extended technical report [21]. We learnt several models using a selection of datasets from varying domains in order to test our hypotheses. In particular we answer three questions in each case: (Q1) does our system learn the correct overall probability distribution? (Q2) does our system capture the correct utility function? (Q3) does our system produce reasonable blameworthiness scores? In this section we first summarise the results from our three experiments before providing a more in-depth analysis of our final experiment as an example. We direct the interested reader to Appendix C for results from the other two experiments. Appendix D contains summaries of our datasets.

4.1 Summary

We performed experiments on data from three different domains. In *Lung Cancer Staging* we used a synthetic dataset generated from the lung cancer staging influence diagram given in [32]. The data was generated assuming that the overall decision strategy recommended in the original paper is followed with some high probability at each decision point. The *Teamwork Management* experiment uses a recently collected dataset of human decision-making in teamwork management [44]. This data was recorded from over 1000 participants as they played a game that simulates task allocation processes in a management environment, and includes self-reported emotional responses from each participant based on their performance. Finally, in *Trolley Problems* we devised our own experimental setup with human participants, using a small-scale survey (documents and data are included in the package [21]) to gather data about hypothetical moral decision-making scenarios. These scenarios took the form of variants on the famous trolley problem [42].

For (Q1) we measure the overall log likelihood of the models learnt by our system on training, validation, and test datasets (see Table 1). A full comparison across a range of similar models and learning techniques is beyond the scope of our work here, although to provide some evidence of the competitiveness of PSDDs we include the log likelihood scores of a sum-product network (SPN), another tractable probabilistic model, created using Tachyon [23] as a benchmark. We also compare the sizes (measured by the number of nodes) and the log likelihoods of PSDDs learnt with and without logical constraints in order to demonstrate the effectiveness of the former approach.

Table 1: Log likelihoods and sizes of the constrained PSDDs (the models we use in our system: *), unconstrained PSDDs, and the SPNs learnt in our three experiments.

Model		Training	Validation	Test	Size
1	PSDD*	-2.047	-2.046	-2.063	134
	PSDD	-2.550	-2.549	-2.564	436
	SPN	-3.139	-3.143	-3.158	1430
2	PSDD*	-5.541	-5.507	-5.457	370
	PSDD	-5.637	-5.619	-5.556	931
	SPN	-7.734	-7.708	-7.658	3550
3	PSDD*	-4.440	-4.510	-4.785	368
	PSDD	-6.189	-6.014	-6.529	511
	SPN	-15.513	-16.043	-15.765	3207

Answering (Q2) is more difficult, as self-reported measures of utility (or other proxy metrics, such as life expectancy in *Lung Cancer Staging*, for example) may form an unreliable baseline. In general, our models are able to match preferences up to ordinality in most cases, but the cardinal representations of utilities depends greatly on the function f in the proportionality relationship between expected decision probabilities and expected utilities. The exact choice of f is highly domain-dependent and an area for further experimentation in future.

In attempting to answer (Q3) we divide our question into two parts: does the system attribute no blame in the correct cases, and does the system attribute more blame in the cases we would expect it to (and less in others)? Needless to say, it is very difficult (perhaps even impossible, at least without an extensive survey of human opinions) to produce an appropriate metric for how correct our attributions of blame are, but we suggest that these two criteria are the most fundamental and capture the core of what we want to evaluate. We successfully queried our models in a variety of settings corresponding to the two questions above and present representative examples below.

4.2 Trolley Problems

In this experiment we extend the well-known trolley problem, as is not uncommon in the literature [5], by introducing a series of different characters that might be on either track: one person, five people, 100 people, one's pet, one's best friend, and one's family. We also add two further decision options: pushing whoever is on the side track into the way of the train in order to save whoever is on the main track, and sacrificing oneself by jumping in front of the train, saving both characters in

the process. Our survey then took the form of asking each participant which of the four actions they would perform (the fourth being inaction) given each possible permutation of the six characters on the main and side tracks (we assume that a character could not appear on both tracks in the same scenario). The general setup can be seen in Figure 1, with locations *A* and *B* denoting the locations of people on the main track and side track respectively.

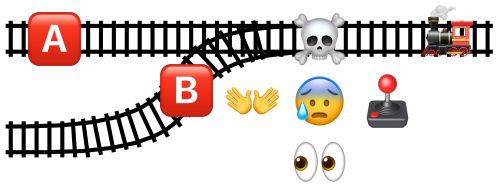

Figure 1: A cartoon given to participants showing the layout of the experimental scenario and the four possible options. Clockwise from top (surrounding the face symbol) these are: sacrificing oneself, flipping the switch, inaction, and pushing the character at *B* onto the main track. Locations *A* and *B* are instantiated by particular characters depending on the context.

Last of all, we added a probabilistic element to our scenarios whereby the switch only works with probability 0.6, and pushing the character at location *B* onto the main track in order to stop the train succeeds with probability 0.8. This was used to account for the fact that people are generally more averse to actively pushing someone than to flipping a switch [41], and people are certainly more averse to sacrificing themselves than doing either of the former. However, depending on how much one values the character on the main track's life, one might be prepared to perform a less desirable action in order to increase their chance of survival.

In answering (Q1) we investigate how well our model serves as a representation of the aggregated decision preferences of participants by calculating how likely the system would be to make particular decisions in each of the 30 contexts and comparing this with the average across participants in the survey. For reasons of space we focus here on a representative subset of these comparisons: namely, the five possible scenarios in which the best friend character is on the main track (see Figure 2). In general, the model's predictions are similar to the answers given in the survey, although the effect of smoothing our distribution during learning is noticeable, especially due to the fact that the model was learnt with relatively few data points. Despite this handicap, the most likely decision in any of the 30 contexts according to the model is in fact the majority decision in the survey, with the ranking of other decisions in each context also highly accurate.

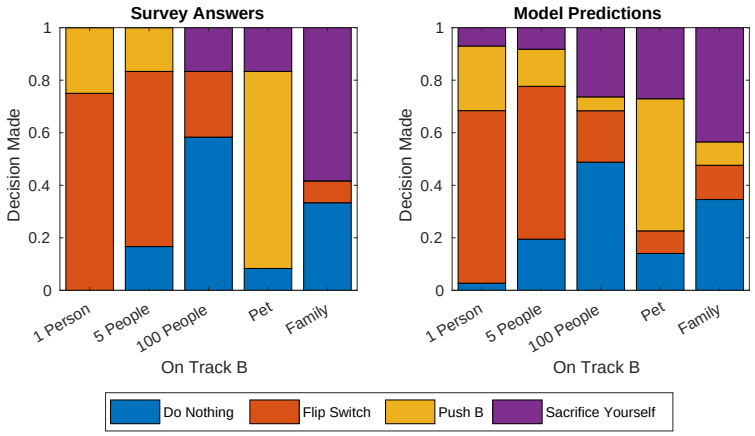

Figure 2: A comparison of the decisions made by participants and the predictions of our model in each of the five scenarios in which the best friend character is on the main track (*A*).

Unlike our other two experiments, the survey data does not explicitly contain any utility information, meaning our system was forced to learn a utility function by using the probability distribution encoded by the PSDD. Within the decision-making scenarios we presented, it is plausible that the decisions made by participants were guided by weights that they assigned to the lives of each of the six characters and to their own life. Given that each of these is captured by a particular outcome variable we chose to construct a utility function that was linear in said variables. We also chose to make the utility function insensitive to context, as we would not expect how much one values the life of a particular character to depend on which track that character was on, or whether they were on a track at all.

For (Q2), with no existing utility data to compare our learnt function, we interpreted the survival rates of each character as the approximate weight assigned to their lives by the participants. While the survival rate is a non-deterministic function of the decisions made in each context, we assume that over the experiment these rates average out enough for us to make a meaningful comparison with the weights learnt by our model. A visual representation of this comparison can be seen in Figure 3. It is immediately obvious that our system has captured the correct utility function to a high degree of accuracy. With that said, our assumption about using survival rates as a proxy for real utility weights does lend itself to favourable comparison with a utility function learnt from a probability distribution over contexts, decisions, and outcomes (which therefore includes survival rates). Given the setup of the experiment, however, this assumption seems justified and, furthermore, to be in line with how most of the participants answered the survey.

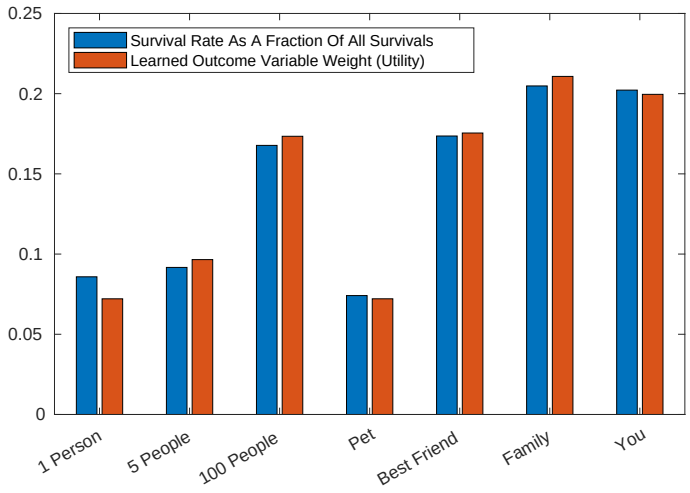

Figure 3: A comparison between the average survival rates of the seven characters (including the participants in the survey), normalised to sum to one, and the corresponding utility function weights learnt by our system.

Because of the symmetric nature of the set of contexts in our experiment, the probability of a particular character surviving as a result of a particular fixed action across all contexts is just the same as the probability of that character not surviving. Hence in answering (Q3) we use our system's feature of being able to accept particular distributions $\Pr'$ over the contexts in which we wish to attribute blame, allowing us to focus only on particular scenarios. Regarding the first part of (Q3), clearly in any of the possible contexts one should not be blamed at all for the the death of the character on the main track for flipping the switch (F) as opposed to inaction (I), because in the latter case they will die with certainty, but not in the former.[4] Choosing a scenario arbitrarily to illustrate this point, with one person on the side track and five people on the main track, we have $db_N(F, I, \neg L_5) = 0$ and $db_N(F, \neg L_5) = 0.307$ (with our measure of cost importance $N = 0.762$, 1.1 times the negative minimum cost of any action).

[4]Note that this is not to say one would not be blameworthy when compared to all other actions as one could, for example, have sacrificed oneself instead, saving all other lives with certainty.

For the second part of (Q3), consider the scenario in which there is a large crowd of a hundred or so people on the main track, but one is unable to tell from a distance if the five or so people on the side track are strangers or one's family. The more likely it is that the family is on the side track, the more responsible one is for their deaths ($\neg L_{Fa}$) if one, say, flips the switch (F) to divert the train. Conversely, we also expect there to be *less* blame for the deaths of the 100 people ($\neg L_{100}$) say, if one did nothing (I), the more likely it is that the family is on the side track (because the cost, for the participant at least, of diverting the train is higher). We compare cases where there is a 0.3 or 0.6 probability that the family is on the side track and for all calculations use the cost importance measure $N = 1$. Therefore, not only would we expect the blame for the death of the family to be higher when pulling the switch in the latter case, we would expect the value to be approximately twice as high as in the former case. Accordingly, we compute values $db_N(F, \neg L_{Fa}) = 0.264$ and $db_N(F, \neg L_{Fa}) = 0.554$ respectively. Similarly, when considering blame for the deaths of the 100 people due inaction, we find that $db_N(I, \neg L_{100}) = 0.153$ in the former case and that $db_N(I, \neg L_{100}) = 0.110$ in the latter case (when the cost of performing another action is higher).

5 Related Work

Our work here is differentiated from related efforts in two main ways: jointly addressing *the automated (constrained) learning of models of moral scenarios* and *tractable reasoning.* We discuss other efforts below. As mentioned before, we do not motivate new definitions for moral responsibility here but draw on HK which, in turn, is based upon earlier work on responsibility [10] and causality [20]. Their work is also related to the intentions model [27] which considers predictions about the moral permissibility of actions via influence diagrams, though there is no emphasis on learning or tractability. In fact, the use of tractable architectures for decision-making itself is recent [8, 31]. Choi et al. employ PSDDs to learn distributions over preference rankings in a work not dissimilar to our own [11]. The main distinction is that the variables in this distribution are the positions of items within preference lists, as opposed to the items themselves. An important part of learning a model of moral decision-making is in learning a utility function. This is often referred to as *inverse reinforcement learning* (IRL) [33, 7]. Our current implementation considers a simple approach for learning utilities (similar to Nielsen and Jensen [34]), but more involved paradigms could indeed have been used. One restriction we faced when performing our experiments was the relative lack of appropriate datasets. Recent work by Jentzsch et al. indicates that language corpora may form suitable resources from which data about ethical norms and moral decision-making may be extracted [22]. Our contributions here are related to the body of work surrounding MIT's Moral Machine experiment [5]. For example, Kim et al. [24] build on earlier theoretical work [28] by developing a computational model of moral decision-making whose predictions they test against Moral Machine data. Their focus is on learning abstract moral principles via hierarchical Bayesian inference, and although our framework can be used to these ends, it is also flexible with respect to different contexts, and allows constraints on learnt models. Noothigattu et al. develop a method of aggregating the preferences of all participants (again, a secondary feature of our system) in order to make a given decision [35]. However, due to the large numbers of such preference orderings, tractability issues arise and so sampling must be used. Recent work by Shaw et al. [40] has sought to address the tension between learnt models of moral decision-making and provable guarantees, and there are several other high-level overviews of strategies for creating moral decision-making frameworks in AI [12, 17]. We refer the reader to these works for more discussions.

6 Conclusion

Our system utilises the definition of decision-making scenarios in HK, and at the same time exploits many of the desirable properties of PSDDs (such as tractability, semantically meaningful parameters, and logically constrained learning). It is flexible in its usage, allowing various inputs and specifications. In general, the models in our experiments are accurate representations of the distributions over the moral scenarios that they are learnt from. Our learnt utility functions, while simple in nature, are still able to capture subtle details and in some scenarios are able to match human preferences with high accuracy using very little data. With these two elements we are able to generate blameworthiness scores that are, prima facie, in line with human intuitions. We hope that our work here goes some way towards bridging the gap between the existing philosophical work on moral responsibility and the existing technical work on decision-making in automated systems.

Acknowledgements

The authors wish to thank several anonymous reviewers for their helpful feedback. Vaishak Belle was supported by a Royal Society University Research Fellowship.

References

[1] Jayadev Acharya, Arnab Bhattacharyya, Constantinos Daskalakis, and Saravanan Kandasamy. Learning and testing causal models with interventions. *arXiv preprint arXiv:1805.09697*, 2018.

[2] Colin Allen, Iva Smit, and Wendell Wallach. Artificial morality: Top-down, bottom-up, and hybrid approaches. *Ethics and Information Technology*, 7(3):149–155, 2005.

[3] Michael Anderson and Susan Leigh Anderson. Geneth: A general ethical dilemma analyzer. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 253–261, 2014.

[4] Automated Reasoning Group (University Of California, Los Angeles). *The SDD Package 2.0*, 2018. *http://reasoning.cs.ucla.edu/sdd*, Accessed 2018-08-17.

[5] Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. The moral machine experiment. *Nature*, 563(7729):59, 2018.

[6] Fahiem Bacchus, Shannon Dalmao, and Toniann Pitassi. Solving #SAT and Bayesian inference with backtracking search. *Journal of Artificial Intelligence Research*, 34:391–442, 2009.

[7] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[8] Debarun Bhattacharjya and Ross D Shachter. Evaluating influence diagrams with decision circuits. *arXiv preprint arXiv:1206.5257*, 2012.

[9] Vicky Charisi, Louise Dennis, Michael Fisher, Robert Lieck, Andreas Matthias, Marija Slavkovik, Janina Sombetzki, Alan FT Winfield, and Roman Yampolskiy. Towards moral autonomous systems. *arXiv preprint arXiv:1703.04741*, 2017.

[10] Hana Chockler and Joseph Y Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

[11] Arthur Choi, Guy Van den Broeck, and Adnan Darwiche. Tractable learning for structured probability spaces: A case study in learning preference distributions. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 2861–2868, 2015.

[12] Vincent Conitzer, Walter Sinnott-Armstrong, Jana Schaich Borg, Yuan Deng, and Max Kramer. Moral decision making frameworks for artificial intelligence. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 4831–4835, 2017.

[13] Adnan Darwiche. SDD: A new canonical representation of propositional knowledge bases. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, page 819, 2011.

[14] Louise Dennis, Michael Fisher, Marija Slavkovik, and Matt Webster. Formal verification of ethical choices in autonomous systems. *Robotics and Autonomous Systems*, 77:1–14, 2016.

[15] Owain Evans, Andreas Stuhlmüller, and Noah D Goodman. Learning the preferences of ignorant, inconsistent agents. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 323–329, 2016.

[16] Robert Gens and Pedro Domingos. Learning the structure of sum-product networks. In *Proceedings of the 30th International Conference on Machine Learning*, pages 873–880, 2013.

[17] Joshua Greene, Francesca Rossi, John Tasioulas, Kristen Brent Venable, and Brian Charles Williams. Embedding ethical principles in collective decision support systems. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 4147–4151, 2016.

[18] Victoria Groom, Jimmy Chen, Theresa Johnson, F Arda Kara, and Clifford Nass. Critic, compatriot, or chump?: Responses to robot blame attribution. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction*, pages 211–218, 2010.

[19] Joseph Y Halpern and Max Kleiman-Weiner. Towards formal definitions of blameworthiness, intention, and moral responsibility. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 1853–1860, 2018.

[20] Joseph Y Halpern and Judea Pearl. Causes and explanations: A structural-model approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 56(4):843–887, 2005.

[21] Lewis Hammond. *TPM4MR Code Package*. University of Edinburgh, 2018. *https://github.com/lrhammond/tpm4mr*, Accessed 2018-10-05.

[22] Sophie Jentzsch, Patrick Schramowski, Constantin Rothkopf, and Kristian Kersting. Semantics derived automatically from language corpora contain human-like moral choices. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, pages 37–44, 2019.

[23] Agastya Kalra. *Tachyon*. University of Waterloo, 2017. *https://github.com/KalraA/Tachyon*, Accessed 2018-08-23.

[24] Richard Kim, Max Kleiman-Weiner, Andrés Abeliuk, Edmond Awad, Sohan Dsouza, Joshua B. Tenenbaum, and Iyad Rahwan. A computational model of commonsense moral decision making. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 197–203, 2018.

[25] Taemie Kim and Pamela Hinds. Who should I blame? Effects of autonomy and transparency on attributions in human-robot interaction. In *The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 80–85, 2006.

[26] Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*, pages 558–567, 2014.

[27] Max Kleiman-Weiner, Tobias Gerstenberg, Sydney Levine, and Joshua B Tenenbaum. Inference of intention and permissibility in moral decision making. In *Proceedings of the 37th Annual Conference of the Cognitive Science Society*, pages 1123–1128, 2015.

[28] Max Kleiman-Weiner, Rebecca Saxe, and Joshua B Tenenbaum. Learning a commonsense moral theory. *Cognition*, 167:107–123, 2017.

[29] Yitao Liang, Jessa Bekker, and Guy Van den Broeck. Learning the structure of probabilistic sentential decision diagrams. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, pages 134–145, 2017.

[30] Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995.

[31] Mazen A Melibari, Pascal Poupart, and Prashant Doshi. Sum-product-max networks for tractable decision making. In *Proceedings of the 15th International Conference on Autonomous Agents & Multiagent Systems*, pages 1419–1420, 2016.

[32] Robert F Nease Jr and Douglas K Owens. Use of influence diagrams to structure medical decisions. *Medical Decision Making*, 17(3):263–275, 1997.

[33] Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 663–670, 2000.

[34] Thomas D Nielsen and Finn V Jensen. Learning a decision maker's utility function from (possibly) inconsistent behavior. *Artificial Intelligence*, 160(1-2):53–78, 2004.

[35] Ritesh Noothigattu, Snehalkumar 'Neil' S Gaikwad, Edmond Awad, Sohan Dsouza, Iyad Rahwan, Pradeep Ravikumar, and Ariel D Procaccia. A voting-based system for ethical decision making. *arXiv preprint arXiv:1709.06692*, 2017.

[36] Judea Pearl. Graphical models for probabilistic and causal reasoning. In *Quantified Representation of Uncertainty and Imprecision*, pages 367–389. Springer, 1998.

[37] Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009.

[38] Robert Peharz, Robert Gens, Franz Pernkopf, and Pedro Domingos. On the latent variable interpretation in sum-product networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(10):2030–2044, 2017.

[39] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *IEEE International Conference on Computer Vision Workshops*, pages 689–690, 2011.

[40] Nolan P. Shaw, Andreas Stöckel, Ryan W. Orr, Thomas F. Lidbetter, and Robin Cohen. Towards provably moral ai agents in bottom-up learning frameworks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 271–277, 2018.

[41] Peter Singer. Ethics and intuitions. *The Journal of Ethics*, 9(3-4):331–352, 2005.

[42] Judith Jarvis Thomson. The trolley problem. *The Yale Law Journal*, 94(6):1395–1415, 1985.

[43] Ilya Volkovich. A guide to learning arithmetic circuits. In *Proceedings of the 29th Conference on Learning Theory*, pages 1540–1561, 2016.

[44] Han Yu, Zhiqi Shen, Chunyan Miao, Cyril Leung, Yiqiang Chen, Simon Fauvel, Jun Lin, Lizhen Cui, Zhengxiang Pan, and Qiang Yang. A dataset of human decision-making in teamwork management. *Scientific Data*, 4:160127, 2017.

Appendix A: Implementation

The importance of having *implementable* models of moral reasoning has been stressed by Charisi et al. [9] amongst others. Our demonstration system runs from the command line and prompts the user for a series of inputs including: data; existing PSDDs, SDDs, or vtrees; logical constraints; utility function specifications; variable descriptions; and finally the decisions, outcomes, and other details needed to compute a particular blameworthiness score. These inputs and any outputs from the system are saved and thus each model and its results can be easily accessed and re-used if needed. Note that we assume each datum is a sequence of fully observed values for binary (possibly as a result of one-hot encoding) variables that correspond to the context, the decisions made, and the resulting outcome, if recorded. Our implementation makes use of two existing resources: The SDD Package 2.0 [4], an open-source system for creating and managing SDDs, including compiling them from logical constraints; and LearnPSDD [29], a recently developed set of algorithms that can be used to learn the parameters and structure of PSDDs from data, learn vtrees from data, and to convert SDDs into PSDDs. The resulting functionalities of our system can then be broken down into four broad areas (a high-level overview of the complete structure of the demonstration system is provided in the package documentation [21]):

- Building and managing models, including converting logical constraints specified by the user in simple infix notation to restrictions upon the learnt model. For example, $(A \wedge B) \leftrightarrow C$ can be entered as a command line prompt using `=(&(A,B),C)`.
- Performing inference by evaluating the model or by calculating the most probable evidence (MPE), both possibly given partial evidence. Each of our inference algorithms are linear in the size of the model, and are based on pseudocode given in [26] and [38] respectively.
- Learning utility functions from data, whose properties (such as being linear or being context-relative) are specified by the user in advance. This learning is done by forming a matrix equation representing our assumed proportionality relationship across all decisions and contexts, then solving to find utilities using non-negative linear regression with L2 regularisation (equivalent to solving a quadratic program).
- Computing blameworthiness by efficiently calculating the key quantities from our embedding, using parameters from particular queries given by the user. Results are then displayed in natural language and automatically saved for future reference.

Appendix B: Complexity Results

Given our concerns over tractability we provide several computational complexity results for our embedding. Basic results were given in [19], but only in terms of the computations being polynomial in $|M|$, $|\mathcal{K}|$, and $|\mathcal{R}(A)|$. Here we provide more detailed results that are specific to our embedding and to the properties of PSDDs. The complexity of calculating blameworthiness scores depends on whether the user specifies an alternative distribution $\Pr'$, although in practice this is unlikely to have a major effect on tractability. Finally, note that we assume here that the PSDD and utility function are given in advance and so we do not consider the computational cost of learning. This parallels the results in HK, in which only the cost of reasoning is considered (there is no mention of how their models are obtained). In general, guarantees within the tractable learning paradigm are provided for *tractable inference within learnt models*, but not for the learning procedure itself, which is often approximate [43]. A summary of our results is given in Table 2.

Table 2: Time complexities for each of the key terms that we compute. If the user specifies an extra distribution $\Pr'$ over contexts, then the complexity is given by the expressions below with each occurrence of the term $|P|$ replaced by $|P| + Q$, where $O(Q)$ is the time taken to evaluate $\Pr'$.

Term	Time Complexity
$\delta_{a,a',\varphi}$	$O(2^{\lvert\mathcal{X}\rvert+\lvert\mathcal{D}\rvert+\lvert\mathcal{O}\rvert}(\lvert\varphi\rvert + \lvert P\rvert))$
$c(a)$	$O(2^{\lvert\mathcal{X}\rvert+\lvert\mathcal{O}\rvert}(U + \lvert P\rvert))$
$db_N(a,a',\varphi)$	$O(2^{\lvert\mathcal{X}\rvert+\lvert\mathcal{O}\rvert}(U + 2^{\lvert\mathcal{D}\rvert}(\lvert\varphi\rvert + \lvert P\rvert)))$
$db_N(a,\varphi)$	$O(\lvert\mathcal{R}(A)\rvert 2^{\lvert\mathcal{X}\rvert+\lvert\mathcal{O}\rvert}(U + 2^{\lvert\mathcal{D}\rvert}(\lvert\varphi\rvert + \lvert P\rvert)))$

Here, $O(|P|)$ is the time taken to evaluate the PSDD P where $|P|$ is the size of the PSDD, measured as the number of parameters; $O(U)$ is the time taken to evaluate the utility function; and $O(|\varphi|)$ is the time taken to evaluate the Boolean function φ, where $|\varphi|$ measures the number of Boolean connectives in φ. We observe that all of the final time complexities are exponential in the size of at least some subset of the variables. This is a result of the Boolean representation; our results are, in fact, more tightly bounded versions of those in HK, which are polynomial in the size of $|\mathcal{K}| = O(2^{|\mathcal{X}|+|\mathcal{D}|+|\mathcal{O}|})$. In practice, however, we only sum over worlds with non-zero probability of occurring. Using PSDDs allows us to exploit this fact in ways that other models cannot, as we can logically constrain the model to have zero probability on any impossible world. Thus, when calculating blameworthiness we can ignore a great many of the terms in each sum and speed up computation dramatically. To give some concrete examples, the model counts of the PSDDs in our experiments were 52, 4800, and 180 out of 2^{12}, 2^{21}, and 2^{23} possible variable assignments, respectively.

Appendix C: Further Experiments

Lung Cancer Staging

We use a synthetic dataset generated with the lung cancer staging influence diagram given in [32]. The data was generated assuming that the overall decision strategy recommended in the original paper is followed with some high probability at each decision point. In this strategy, a thoractomy is the usual treatment unless the patient has mediastinal metastases, in which case a thoractomy will not result in greater life expectancy than the lower risk option of radiation therapy, which is then the preferred treatment. The first decision made is whether a CT scan should be performed to test for mediastinal metastases, the second is whether to perform a mediastinoscopy. If the CT scan results are positive for mediastinal metastases then a mediastinoscopy is usually recommended in order to provide a second check, but if the CT scan result is negative then a mediastinoscopy is not seen as worth the extra risk involved in the operation. Possible outcomes are determined by variables that indicate whether the patient survives the diagnosis procedure and survives the treatment, and utility is measured by life expectancy.

For (Q1) we again measure the overall log likelihood of the models learnt by our system on training, validation, and test datasets. In particular, our model is able to recover the artificial decision-making

strategy well (see Figure 4); at most points of the staging procedure the model learns a very similar distribution over decisions, and in all cases the correct decision is made the majority of times.

Answering (Q2) here is more difficult as the given utilities are not necessarily such that our decisions are linearly proportional to the expected utility of that decision. However, our strategy was chosen so as to maximise expected utility in the majority of cases. Thus, when comparing the given life expectancies with the learnt utility function, we still expect the same ordinality of utility values, even if not the same cardinality. In particular, our function assigns maximal utility (1.000) to the successful performing of a thoractomy when the patient does not have mediastinal metastases (the optimal scenario), and any scenario in which the patient dies has markedly lower utility (mean value 0.134).

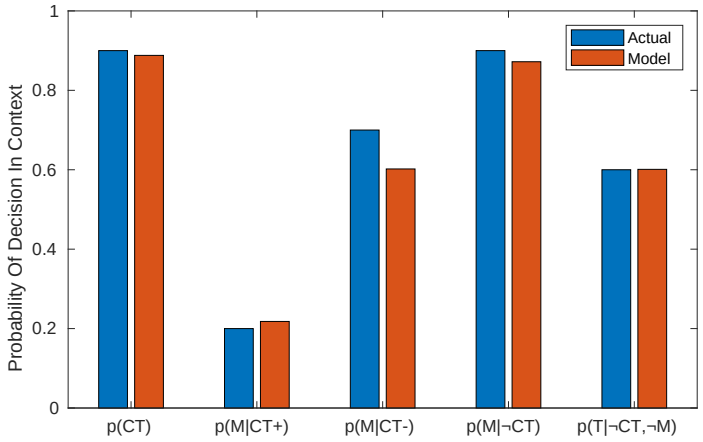

Figure 4: A comparison between the five probability values specified in our data generation process and the corresponding values learnt by our system from this data.

Regarding the first part of (Q3), one case in which we have blameworthiness scores of zero is when performing the action being judged is *less* likely to result in the outcome we are concerned with than the action(s) we are comparing it to. The chance of the patient dying in the diagnostic process ($\neg S_{DP}$) is increased if a mediastinoscopy (M) is performed, hence the blameworthiness for such a death due to *not* performing a mediastinoscopy should be zero. As expected, our model assigns $db_N(\neg M, M, \neg S_{DP}) = 0$. To answer the second part of (Q3), we show that the system produces higher blameworthiness scores when a negative outcome is more likely to occur (assuming the actions being compared have relatively similar costs). For example, in the case where the patient does not have mediastinal metastases then the best treatment is a thoractomy, but a thoractomy will not be performed if the result of the last diagnostic test performed is positive. The specificity of a mediastinoscopy is higher than that of a CT scan, hence a CT scan is more likely to produce a false positive and thus (assuming no mediastinoscopy is performed as a second check) lead to the wrong treatment.[5] In the case where only one diagnostic procedure is performed we therefore have a higher degree of blame attributed to the decision to conduct a CT scan (0.013) as opposed to a mediastinoscopy (0.000), where we use $N = 1$.

Teamwork Management

Our second experiment uses a recently collected dataset of human decision-making in teamwork management [44]. This data was recorded from over 1000 participants as they played a game that simulates task allocation processes in a management environment. In each level of the game the player has different tasks to allocate to a group of virtual workers that have different attributes and capabilities. The tasks vary in difficulty, value, and time requirements, and the player gains feedback from the virtual workers as tasks are completed. At the end of the level the player receives a score based on the quality and timeliness of their work. Finally, the player is asked to record their emotional

[5]Note that even though a mediastinoscopy has a higher cost (as the patient is more likely to die if it is performed), it should not be enough to outweigh the test's accuracy in this circumstance.

response to the result of the game in terms of scores corresponding to six basic emotions. We simplify matters slightly by considering only the self-declared management strategy of the player as our decisions. Within the game this is recorded by five check-boxes at the end of the level that are not mutually exclusive, giving 32 possible overall strategies. These strategy choices concern methods of task allocation such as load-balancing (keeping each worker's workload roughly even) and skill-based (assigning tasks by how likely the worker is to complete the task well and on time), amongst others. We also measure utility purely by the self-reported happiness of the player, rather than any other emotions.

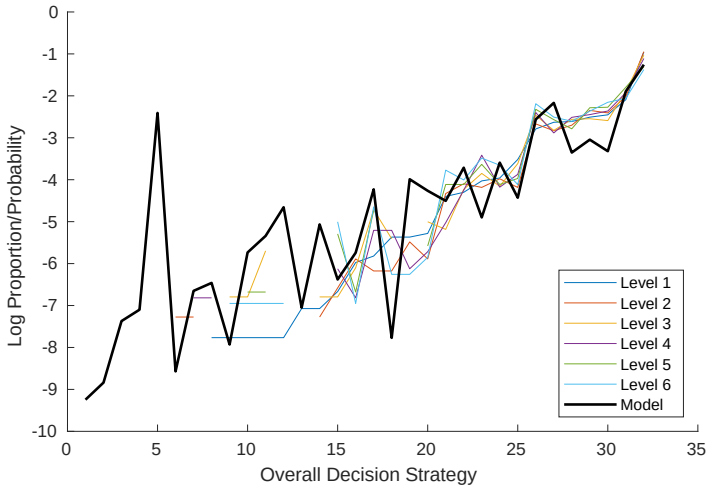

Figure 5: The log probability assigned to each possible decision strategy across all contexts by our model, compared to the log proportion of times each strategy was used in the six levels of the game by participants. Strategies are sorted in ascending order by their proportion of use in level 1 and gaps in each plot represent strategies never used in that game level.

As part of our answer to (Q1) we investigate how often the model would employ each of the 32 possible strategies (where a strategy is represented by an assignment of values to the binary indicator decision variables) compared to the average participant (across all contexts), which can be seen in Figure 5. In general the learnt probabilities are similar to the actual proportions in the data, though noisier. The discrepancies are more noticeable (though understandably so) for decisions that were made very rarely, perhaps only once or twice in the entire dataset. These differences are also partly due to smoothing (i.e. all strategies have a non-zero probability of being played).

For (Q2) we use the self-reported happiness scores to investigate our assumption that the number of times a decision is made is (linearly) proportional to the expected utility based on that decision. In order to do this we split the data up based on the context (game level) and produce a scatter plot (Figure 6) of the proportion of times a set of decisions is made against the average utility (happiness score) of that decision. Overall there is no obvious positive linear correlation as our original assumption would imply, although this could be because of any one or combination of the following reasons: players do not play enough rounds of the game to find out which strategies reliably lead to higher scores and thus (presumably) higher utilities; players do not accurately self-report their strategies; or players' strategies have relatively little impact on their overall utility based on the result of the game. We recall here that our assumption essentially comes down to supposing that people more often make decisions that result in greater utilities. The eminent plausibility of this statement, along with the relatively high likelihood of at least one of the factors in the list above means we do not have enough evidence here to refute the statement, although certainly further empirical work is required in order to demonstrate its truth.

Investigating this discrepancy further, we learnt a utility function (linear and context-relative) from the data and inspected the average weights given to the outcome variables (see right plot in Figure 7). A correct function should place higher weights on the outcome variables corresponding to higher ratings, which is true for timeliness, but not quite true for quality as the top rating is weighted only

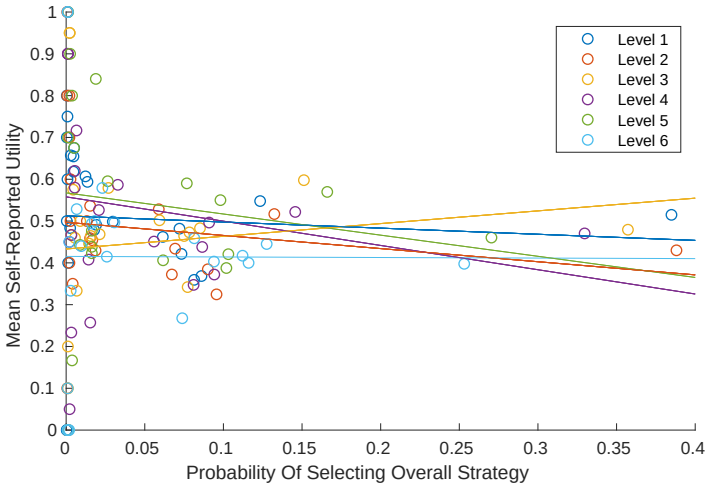

Figure 6: Each point is a decision strategy in a level of the game; we compare the proportion of times it is used against the average self-reported utility that results from it. Each line is a least-squares best fit to the points in that level.

third highest. We found that the learnt utility weights are in fact almost identical to the distribution of the outcomes in the data (see left plot in Figure 7). Because our utility weights were learnt on the assumption that players more often use strategies that will lead to better expected outcomes, the similarity between these two graphs adds further weight to our suggestion that, in fact, the self-reported strategies of players have very little to do with the final outcome.

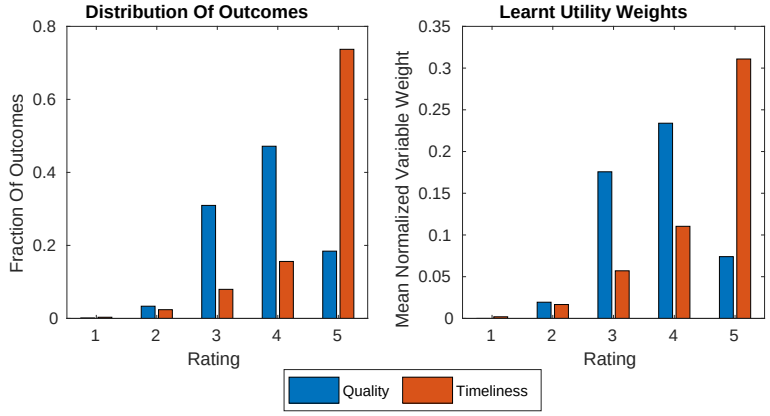

Figure 7: A comparison of the learnt utility weights for each of the outcome variables (to the right) and the proportion of times each outcome occurs in the data (to the left).

To answer (Q3) we examine cases in which the blameworthiness score should be zero, and then compare cases that should have lower or higher scores with respect to one another. Once again, comprehensive descriptions of each of our tested queries are omitted for reasons of space, but here we present some representative examples.[6] Firstly, we considered level 1 of the game by choosing an alternative distribution $\Pr'$ over contexts when generating our scores. Here a player is less likely to receive a low rating for quality (Q_1 or Q_2) if they employ a skill-based strategy where tasks are more frequently allocated to better workers (S). As expected, our system returns $db_N(S, \neg S, Q_1 \vee Q_2) = 0$. Secondly, we look at the timeliness outcomes. A player is less likely to obtain the top timeliness

[6] In all of the blameworthiness scores below we use the cost importance measure $N = 1$.

rating (T_5) if they do *not* use a strategy that uniformly allocates tasks (U) compared to their *not* using a random strategy of allocation (R). Accordingly, we find that $db_N(\neg U, \neg T_5) > db_N(\neg R, \neg T_5)$, and more specifically we have $db_N(\neg U, \neg T_5) = 0.002$ and $db_N(\neg R, \neg T_5) = 0$ (i.e. a player should avoid using a random strategy completely if they wish to obtain the top timeliness rating).

Appendix D: Dataset Summaries

The full set of data, source code, and other supplementary materials are included within a package which will be made available online upon publication of an extended version of this work. Here we provide brief summaries of the three datasets used in our experiments, including the variable encoding used for each domain and the underlying constraints.

Table 3: A summary of the lung cancer staging data used in our first experiment.

Number of data points	100000
Number of variables	12
Context variables ($\mathcal{X}$)	Mediastinal Metastases (MM), CT Positive (CT_+), CT Negative (CT_-), No CT ($CT_{N/A}$), Mediastinoscopy Positive (M_+), Mediastinoscopy Negative (M_-), No Mediastinoscopy ($M_{N/A}$)
Decision variables ($\mathcal{D}$)	Perform CT (CT), Perform Mediastinoscopy (M)
Outcome variables ($\mathcal{O}$)	Perform Thoractomy (T), Diagnosis Procedures Survived (S_{DP}), Treatment Survived (S_T)
Constraints	$(CT_+ \vee CT_-) \leftrightarrow CT$ $CT_{N/A} \leftrightarrow \neg CT$ $(M_+ \vee M_-) \leftrightarrow M$ $M_{N/A} \leftrightarrow \neg M$ $M_- \rightarrow T$ $M_+ \rightarrow \neg T$ $(CT_- \wedge \neg M) \rightarrow T$ $(CT_+ \wedge \neg M) \rightarrow \neg T$ $\neg S_{DP} \rightarrow M$ $\neg(CT_+ \wedge CT_-)$ $\neg(M_+ \wedge M_-)$ $\neg S_{DP} \rightarrow \neg S_T$
Model count	52
Utilities given?	Yes (life expectancy)

Table 4: A summary of the teamwork management data used in our second experiment.

Number of data points	7446
Number of variables	21
Context variables ($\mathcal{X}$)	Level 1 (L_1), ... , Level 6 (L_6)
Decision variables ($\mathcal{D}$)	Other (O), Load-balancing (L), Uniform (U), Skill-based (S), Random (R)
Outcome variables ($\mathcal{O}$)	Timeliness 1 (T_1), ... , Timeliness 5 (T_5), Quality 1 (Q_1), ... , Quality 5 (Q_5)
Constraints	$\bigvee_{i \in \{1,...,6\}} L_i$ $L_i \rightarrow \neg \bigvee_{j \in \{1,...,6\} \setminus i} L_j \forall i \in \{1, ..., 6\}$ $\bigvee_{i \in \{1,...,5\}} T_i$ $T_i \rightarrow \neg \bigvee_{j \in \{1,...,5\} \setminus i} T_j \forall i \in \{1, ..., 5\}$ $\bigvee_{i \in \{1,...,5\}} Q_i$ $Q_i \rightarrow \neg \bigvee_{j \in \{1,...,5\} \setminus i} Q_j \forall i \in \{1, ..., 5\}$
Model count	4800
Utilities given?	Yes (self-reported happiness score)

Table 5: A summary of the trolley problem data used in our third experiment.

Number of data points	360
Number of variables	23
Context variables ($\mathcal{X}$)	One Person On Track A (A_1), ... , Family On Track A (A_{Fa}), One Person On Track B (B_1), ... , Family On Track B (B_{Fa})
Decision variables ($\mathcal{D}$)	Inaction (I), Flip Switch (F), Push B (P), Sacrifice Oneself (S)
Outcome variables ($\mathcal{O}$)	One Person Lives (L_1), ... , Family Lives (L_{Fa}), You Live (L_Y)
Constraints	$\bigvee_{i\in\{1,...,Fa\}} A_i$ $\bigvee_{i\in\{1,...,Fa\}} B_i$ $\neg(A_i \wedge B_i) \forall i \in \{1, ..., Fa\}$ $A_i \rightarrow \neg \bigvee_{j\in\{1,...,Fa\}\setminus i} A_j \forall i \in \{1, ..., Fa\}$ $B_i \rightarrow \neg \bigvee_{j\in\{1,...,Fa\}\setminus i} B_j \forall i \in \{1, ..., Fa\}$ $\bigvee_{D\in\{N,F,P,S\}} D$ $D \rightarrow \neg \bigvee_{D'\in\{N,F,P,S\}\setminus D} D'$ $(A_i \wedge N) \rightarrow \neg L_i \forall i \in \{1, ..., Fa\}$ $(B_i \wedge N) \rightarrow L_i \forall i \in \{1, ..., Fa\}$ $L_i \rightarrow (A_i \vee B_i) \forall i \in \{1, ..., Fa\}$ $(S \wedge (A_i \vee B_i)) \rightarrow L_i \forall i \in \{1, ..., Fa\}$ $L_Y \leftrightarrow \neg S$ $(L_i \wedge (P \vee F)) \rightarrow \neg \bigvee_{j\in\{1,...,Fa\}\setminus i} L_j \forall i \in \{1, ..., Fa\}$ $(\neg L_i \wedge (P \vee F)) \rightarrow \bigvee_{j\in\{1,...,Fa\}\setminus i} L_j \forall i \in \{1, ..., Fa\}$
Model count	180
Utilities given?	No