Modeling Risky Choices in Unknown Environments

Ville Tanskanen VILLE.TANSKANEN@HELSINKI.FI
Chang Rajani CHANG.RAJANI@HELSINKI.FI
Department of Computer Science, University of Helsinki, Finland

Homayun Afrabandpey HOMAYUN.AFRABANDPEY@NOKIA.COM
Nokia, Finland

Aini Putkonen AINI.PUTKONEN@AALTO.FI
Aurélien Nioche NIOCHE.AURELIEN@GMAIL.COM
Department of Communications and Networking, Aalto University, Finland

Arto Klami ARTO.KLAMI@HELSINKI.FI
Department of Computer Science, University of Helsinki, Finland

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Decision-theoretic models explain human behavior in choice problems involving uncertainty, in terms of individual tendencies such as risk aversion. However, many classical models of risk require knowing the distribution of possible outcomes (rewards) for all options, limiting their applicability outside of controlled experiments. We study the task of learning such models in contexts where the modeler does not know the distributions but instead can only observe the choices and their outcomes for a user familiar with the decision problems, for example a skilled player playing a digital game. We propose a framework combining two separate components, one for modeling the unknown decision-making environment and another for the risk behavior. By using environment models capable of learning distributions we are able to infer classical models of decision-making under risk from observations of the user's choices and outcomes alone, and we also demonstrate alternative models for predictive purposes. We validate the approach on artificial data and demonstrate a practical use case in modeling risk attitudes of professional esports teams.

Keywords: risk models; prospect theory; inverse reinforcement learning; hybrid models

1. Introduction

Explaining human decision-making under risk has long been studied by behavioral economists and cognitive psychologists, in form of white–box theoretical models that explain individuals' deviations from maximizing the expected value when facing risky options. Prospect Theory (PT) (Kahneman and Tversky, 1979) and its descendant Cumulative Prospect Theory (CPT) (Tversky and Kahneman, 1992) are prime examples of such models. They explain risk behavior – how people make choices when the outcomes are stochastic – via interpretable parameters and can predict future choices. Estimating the parameters is typically done on data collected in controlled experiments, where the outcome probabilities are known and the choice problems are carefully constructed to expose specific types of risk behavior (Glöckner and Pachur, 2012; Stott, 2006).

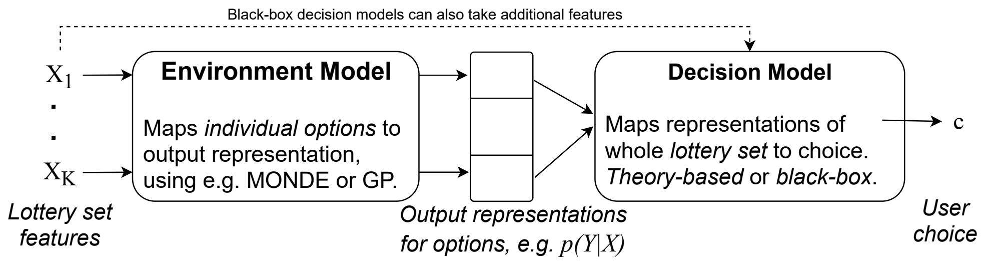
Black-box decision models can also take additional features
X_1
X_K
Lottery set features
Environment Model
Maps *individual options* to output representation, using e.g. MONDE or GP.
Output representations for options, e.g. $p(Y|X)$
Decision Model
Maps representations of whole *lottery set* to choice. *Theory-based* or *black-box*.
c
User choice

Figure 1: Two-component model for risk behavior in contexts where the user is familiar with the environment but the modeler is unaware of the outcome distributions of individual options. The **environment model** maps observable features of individual options into a latent *output representation*, and the **decision model** models the *risk behavior* of a user based on the representations of all options.

For predictive modeling of risky choices, we can also use machine learning (ML) (Chajewska et al., 2001; Hartford et al., 2016; Kolumbus and Noti, 2019; Phan et al., 2016) or hybrid methods that also incorporate elements of cognitive theory (Bourgin et al., 2019; Noti et al., 2016; Plonsky et al., 2017; Rosenfeld et al., 2016; Peterson et al., 2021). Hybrid models are typically ML models that incorporate theory-based elements, such as possible loss aversion, as additional inputs, and have been demonstrated to outperform both purely theoretical models and pure ML models in prediction tasks (Erev et al., 2017). A focus in many hybrid approaches thus far has been in addressing incomplete information – ambiguity – the decision maker has about the environment, identified as an important challenge in decision-making literature (Camerer and Weber, 1992; Trautmann and Van De Kuilen, 2015). These models have extended the scope of risk modeling beyond the simplest settings, but they are still trained on experiments specifically designed for learning the risk behavior in limited cases, considering e.g. only binary lotteries where exactly one outcome is assumed ambiguous (Bourgin et al., 2019; Erev et al., 2017; Plonsky et al., 2017) or a specific type of partial knowledge of the outcome in binary lotteries (Peysakhovich and Naecker, 2017).

The examples above consider problems where the *user* making the decisions has only partial knowledge of the environment, but the modeler knows and even has control of it. We consider partial information in risk modeling from the opposite perspective: We want to learn risk models by observing the choices of users in environments that are unknown for the *modeler*. That is, we assume that the modeler does not know the outcome distributions for the possible options and has no control over the experiment. Previous work on risk attitudes in less controlled cases has focused on specific domains (Krčál et al., 2016; Sydnor, 2010), whereas we want a general approach that does not require domain knowledge. To facilitate this, we need to assume more from the user and make the simplifying assumption that they have no ambiguity about the environment. This holds, for instance, for expert players of digital games with transparent user interface, as well as for our example case of competitive esports teams that employ professional coaches and analysts to provide information about the possible outcomes of all decisions.

To learn risk models in such conditions, we propose a novel hybrid approach combining cognitive theory with data-driven models. We first construct a flexible data driven representation of the environment using e.g. Gaussian processes (GP) or Monotonic Neural Density Estimator (MONDE; Chilinski and Silva, 2020), and then provide its output for a separate risk model, often a theory-based model such as CPT. This is in contrast to the existing hybrid approaches (Bourgin et al., 2019; Erev et al., 2017; Plonsky et al., 2017) that feed theory-based features to a black box ML model. Our approach, illustrated in Figure 1, allows both training traditional risk models outside controlled experiments designed for inferring the risk attitudes as well as purely predictive decision models.

The approach is loosely related to model-based reinforcement learning (RL) that also separates the environment model from a policy (Sutton and Barto, 2018), but departs from the *goal* of RL. Our aim is not to find a reward-maximizing policy, but to understand the policy of an agent operating in the environment and we do this based on observed data of independent decisions. The work also relates to *imitation learning* and *behavioral cloning* (Kroemer et al., 2021), where the goal is to mimic the user behavior. This is typically done with arbitrary policies, and our approach can be interpreted as one way of conducting imitation learning with interpretable policies.

We demonstrate the approach in learning white box models for risk behavior from observed data, by constraining the environment model to output probability distributions using MONDE. We also introduce a decision module based on recurrent neural networks for predicting risk behavior and for evaluation of how well the individual users follow specific risk theories. The proposed architecture can efficiently model choice problems, by supporting varying number of inputs with the same number of outputs. Furthermore, we introduce a new pseudo-labeling technique for addressing the selection bias in observed data. We empirically evaluate the approach on synthetic data to best characterise the effect of the model alternatives, and then demonstrate it in a concrete application case of modeling risk attitudes of professional esports teams.

2. The Learning Problem

We start with a motivational example to help understand the setting, but note that the approach is general. The task is to model risk behavior of players of a turn-based computer game where on every turn they select one discrete action and observe a stochastic outcome (e.g. a reward) for it. The outcome may be provided by the system (e.g. the amount of damage dealt in a role-playing game) or it may depend on the player's skill (e.g. result of a real-time combat sequence initiated by the action). In both cases the player makes the decision based on some features available for each option (e.g. attack and defence skills of the characters involved) and we assume players are familiar with the game, knowing the possible outcomes and their probabilities with sufficient accuracy. The modeler, however, does not know the probabilities for the outcomes. We will later consider a concrete case of esports teams making the decision of how much of their resources to invest for a particular round. The competitive teams have very accurate understanding of their winning chances, but we as the modelers need to learn them from historical data. Outside of gaming, any competition where competitors have spent significant effort training for it, serves as an example. For instance, for play calls in American Football or opening moves in chess the decision maker knows which actions are risky, but the modeler does not.

Concepts and Notation: We model the risk behavior of a *user* that faces a number of independent choice problems (called *lottery sets*), each of them characterized by K vector-valued *features* $\{x_1, \ldots, x_K\}$ for the K different options (with varying K for each set), and always selects exactly one of these stochastic *options* $Y|x$ according to some unknown policy $\pi(\{(Y_1|x_1), \ldots, (Y_K|x_K)\})$. We assume continuous outcomes $y \in \mathbb{R}$ with larger y corresponding for more favorable outcomes for notational simplicity. The outcomes follow some probability distribution $p(y|x)$, which is assumed known (with sufficient accuracy) for the user but unknown for the modeler. We assume $p(y|x)$ is static and somehow smooth, to facilitate learning, but do not make assumptions on how the lottery sets are formed.

We observe a user facing N_l lottery sets with varying number of options K_i, and we observe the features x_{ik} for all options $i \in \{1, \ldots, N_l\}$. The outcome y_{c_i}, however, is only observed for the choice $c_i \in \{1, \ldots, K_i\}$ the user made. This is because in most use cases the choice triggers a change in the environment and we can only observe the outcome for that particular choice. The full data is hence $\{\{x_{i1}, \ldots, x_{iK_i}\}, y_{c_i}, c_i\}_{i=1}^{N_l}$.

Approach: We want to learn a model for the risk attitude of the user by observing their choices, by learning both how the environment operates (a mapping $x \mapsto y$) and how the user behaves (the policy π). For this, we propose a two-component design shown in Figure 1. The first component models the environment, learning the stochastic outcomes $Y|x$. The second component models the risk behavior of the user, based on output of the first component and potentially other information. This architecture is influenced by the general principle of separately modeling the environment and the user, used e.g. in machine teaching (Zhu et al., 2018) and model-based RL (Sutton and Barto, 2018), but not previously used for modeling risk attitudes. Similar model decomposition has been used also as basis for machine theory of mind (Rabinowitz et al., 2018) and as intermediate between model-based and data-driven learning in robotics (Karkus et al., 2019).

The architecture is entirely modular, supporting alternative models for both components. However, they are tied to each other so that white box models for explaining risk behavior require specific outputs for the environment model. For predicting the user's behavior, however, we can use almost arbitrary models for both components. We will first discuss the environment models in Section 3 and then the decision models in Section 4.

3. Environment Models

The environment model maps the features x of individual options into representations of $Y|x$ useful for making decisions. In principle it can be any learnable function $g(x, \phi)$, such as a neural network or Gaussian process. We are not interested in the specific architecture, but instead focus on (a) the form of the output representations, and (b) how to best train the model on the observed data. We consider three alternatives for the output: a point prediction $\hat{y} = g(f(x, \phi))$, a latent representation $f(x, \phi)$ of a model trained for point predictions, and cumulative distribution function (CDF) $P(Y|x)$. The last one is motivated by the classical risk models, such as CPT, requiring distributional information as input.

3.1. Output Representation

For the first two representations we can use arbitrary black box models, typically feedforward neural networks with $f(x, \phi)$ denoting the last hidden layer and $g(\cdot)$ mapping it to the prediction. The latent representation $f(x, \phi)$ is a more informative, and we will later demonstrate that it helps for learning the decision model.

For $P(Y|x)$, we need to use specific models designed to provide distributions, and to emphasize this we use the notation $P_\phi(Y < y|x)$ for a model that outputs CDFs. For any such model, we form an outcome representation as fixed-dimensional vector $q_{\boldsymbol{p}} = q(\boldsymbol{p})$ storing the quantiles for given range of probabilities $\boldsymbol{p}$. This is sufficient information for learning many white-box decision models. To avoid making additional assumptions, we use MONDE (Chilinski and Silva, 2020) as a flexible model for CDFs. It directly optimizes the likelihood of the observed outcomes as the derivative of the CDF, $p_\phi(y|x) = \frac{d}{dy} P_\phi(Y < y|x)$. By forming the likelihood using the derivative, the model sidesteps the need to compute the normalization constant of the distribution, and the added computational overhead of double automatic differentiation remains manageable.

For small training data it may be difficult to learn a fully non-parametric estimate. In such cases the learning problem can be regularized by explicitly assuming a specific functional form of the distribution $Y|X$ and computing the CDF using the distribution's quantile function. For instance, we can use Gaussian processes (Rasmussen and Williams, 2005) and neural networks outputting the parameters of the selected distribution.

3.2. Learning

The parameters ϕ of the environment model are trained in standard fashion; we used PyTorch with Adam optimizer. Here we focus on how to address the semi-supervised nature of the problem: For the ith lottery set we observe a collection of features $\{x_{i1}, \dots, x_{iK_i}\}$, but only one y_{c_i}, the outcome of the option the user selected. Importantly, the labeling process is not uniformly random, but instead the labels are selectively provided based on the policy π_u of the user u. In most cases we observe outcomes for a highly biased selection of features, obtaining very few – if any – labels for options with low probability under π_u. Figure 2 shows an example where for $x \in [0.5, 1]$ we do not observe any outcomes.

We can learn environment models with the supervised labels alone, but also provide ways for learning better models. The first relies on the insight that we do not need to accurately model the bad options. Instead, it is enough that the outcome representation for them is such that the decision model will ignore it. For instance, for CPT, the softmax normalization will imply it is sufficient to output low cumulative prospect value (see Section 4.1 for details). The second approach utilises data collected for other users under different policies $\pi_{u'}$.

Pseudo-labeling: Pseudo-labeling (Lee, 2013) refers to methods determining pseudo-labels y' for instances without a label, to complement the supervised data with $\{x_{ik}, y'_{ik}\}_{k \neq c_i}$. We propose a heuristic encouraging models to identify the unselected options as poor, corresponding to low values of y for typical realizations. For models that output $P(Y|x)$ we have an internal definition for 'low': Since the model defines – at any stage of optimization – $P(Y|x)$ for all possible x, we can use the current estimate to define the scale by using pseudo-label y' corresponding to some low quantile of the current estimate.

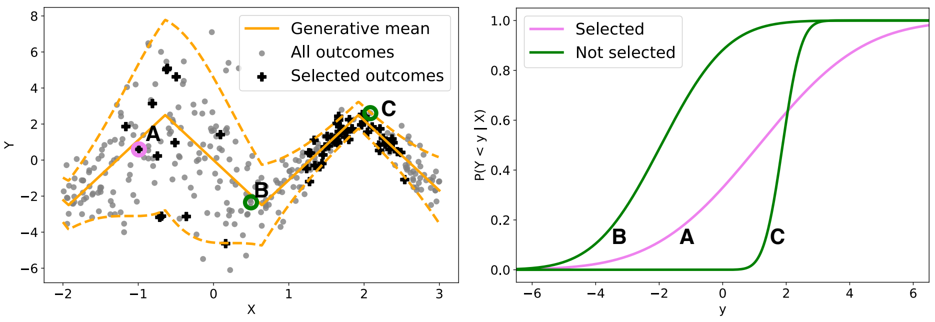

Figure 2: **Left** Example environment with orange lines showing the true dynamics (mean: solid; 0.05 and 0.95 quantiles: dashed) and black '+' indicating the outcomes for the choices of the user. The gray dots show hypothetical outcomes for the other options to illustrate how outcomes available for training the model are clearly non-uniform. The circles indicate a single lottery set of three options, denoted by capital letters, with A corresponding to the choice of the user. **Right** The user makes decisions according to distribution of possible values, here shown for the lottery set indicated by the circles. The choice of A corresponds to risk-seeking behavior; the expected value is lower than option C's but the user still prefers it.

Given a current estimate of the environment model $\hat{P}(Y|x)$ and a lottery set $\{x_1, \ldots, x_K\}$, we compute estimates of CDFs $\{\hat{P}(Y_1|x_1), \ldots, \hat{P}(Y_K|x_K)\}$, then select a number of covariates to sample the pseudo-label for from a low quantile range $[p_{\text{low}}, p_{\text{high}}]$ using inverse transform sampling $y' \sim \hat{P}^{-1}_{Y|x'}(U(p_{\text{low}}, p_{\text{high}}))$. Both the number of the pseudo-labels and the quantile range are hyperparameters; we assign pseudo-labels for random options so that the total fraction per minibatch increases linearly from 3% to 10% during training.

Shared Environment Models: Sometimes the environment is mechanistic and independent of the user, e.g. when investing moderate (not market-changing) amounts in stocks or for turn based games with no skill component. In contrast, a game where the outcome depends on the motor skills of the player would be an example of a non-shared environment.

For shared environments we can (and should) train the environment model on data of all users $\{\{x^{(u)}_{ic_i}, y^{(u)}_{ic_i}\}_{i=1}^{N_l^{(u)}}\}_{u=1}^{N_u}$, but not just because of the added total volume. By using multiple users we can observe outcomes also for features that a particular user would never consider. Other users with slightly different policies help modeling borderline cases better, whereas even irrational users selecting randomly help learning $P(Y|x)$ for generally poor options. We will later show how different compositions of policies influences the learning.

4. Decision Models

The second component models the behavior of an individual user, either for interpreting or predicting their risk behavior. Next we describe one theory-based model and one black box predictive model based on recurrent neural networks as concrete examples.

4.1. Cumulative Prospect Theory

A core motivation for our work was in enabling training of theory-based risk models from observational data, and as a specific example we describe the popular cumulative prospect theory (CPT) model (Tversky and Kahneman, 1992). It assumes humans have non-linear utility functions and a distorted perception of probabilities (over/underestimating the small/large probabilities). More formally, the continuous version of CPT (Wakker and Tversky, 1993) assumes decisions are based on *cumulative prospect values* (CPV)

$$\mathrm{CPV}(Y|x) = \int_{-\infty}^{0} v(a)\frac{d}{da}(w^{-}(F(a)))da + \int_{0}^{\infty} v(a)\frac{d}{da}(-w^{+}(1-F(a)))da, \tag{1}$$

where $F(a)$ is the CDF of the random variable $Y|x$. It immediately follows that CPT can only be trained based on environment models that output the CDF. The choice itself is modeled with softmax as $P(C=c) = \frac{\exp(\mathrm{CPV}_c)}{\sum_{j=1}^{K}\exp(\mathrm{CPV}_j)}$.

The CPV depends on weight function $w(\cdot)$ and utility function $v(\cdot)$ that transform the densities and the outcomes of a random variable. We use

$$w^{+}(p) = \frac{p^{\gamma^{+}}}{\left(p^{\gamma^{+}} + (1-p)^{\gamma^{+}}\right)^{\frac{1}{\gamma^{+}}}},\ w^{-}(p) = \frac{p^{\gamma^{-}}}{\left(p^{\gamma^{-}} + (1-p)^{\gamma^{-}}\right)^{\frac{1}{\gamma^{-}}}},\ v(a) = \begin{cases} a^{\alpha}, \text{ if } a \geq 0 \\ -\lambda(-a)^{\beta}, \text{ if } a < 0 \end{cases}$$

following Tversky and Kahneman (1992). A specific risk behavior is hence defined by five parameters: α, β, λ, γ^{+}, and γ^{-}. We optimize the log-likelihood of the observed choices using stochastic gradient ascent w.r.t the unconstrained parameters and use softplus transformation for λ and sigmoid for the others to constrain them to their appropriate ranges.

4.2. Black-box Model for Risk Behavior

The other main use-case is predictive modeling of risk behavior, which can also be done using black-box models. As will be shown in Section 5.2, we still want to use outputs of the environment model as inputs rather than the original features. We next propose a concrete black box architecture for predicting choices specifically in contexts where the user faces lottery sets of varying numbers of options and is expected to make the decisions independently based on fixed risk tendency. In principle any neural network mapping inputs to probability distributions over the K options could work, but we want (a) support for general and varying K combined with (b) easy minibatching over multiple lotteries, (c) without padding, which would be inefficient in case some lottery sets have very large K.

Recurrent neural networks (RNNs) offer a natural solution for variable length inputs. They are often used in partially observable sequential decision making and reinforcement learning contexts (Bhattacharyya et al., 2018), but here we use them to model data with variable amounts of options. We propose a combination of two RNNs illustrated in Figure 3 and defined as $\mathrm{RNN}_1(\mathrm{RNN}_2(z))$, where $z \in \mathbb{R}^{K_i \times d}$ denotes all inputs for the decision model, for instance the quantiles $z = q_{\boldsymbol{p}}$ or concatenation of input features and a hidden state of point prediction $z = [x, f(x, \phi)]$. One RNN maps the variable length inputs into a joint hidden representation of size h, and the other takes the hidden representation and outputs a single value for each option, followed by softmax normalization to a probability distribution.

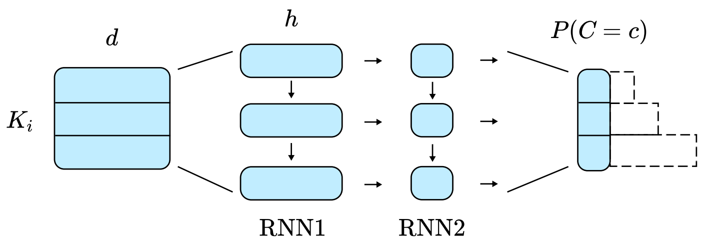

Figure 3: Black-box model for predicting choices. The output representations (here with $K_i = 3$, but in general variable) are fed to the first RNN that provides h features for the second RNN that outputs selection probabilities for all options.

This structure has the advantage of avoiding padding for both inputs and outputs, and the model does not output probabilities for options that cannot be selected.

The variable-length sequences are input as a packed tensor for efficient computation. More specifically, during training a minibatch is a $(b \times K_i \times d)$ tensor, where b is the batch size, K_i is the number of possible options for that particular lottery set and d is the number of features (e.g quantiles or task specific features). For each input we then output a $(b \times K_i)$ tensor with the probabilities for each option. Note that due to the sequential nature of RNNs the model inherently assumes an ordering over the options. To reduce the risk of this influencing the choices we randomize the order of the options within each lottery set during each training step, and in practice the model seems to result in order-invariant choices.

5. Experiments

We validate and demonstrate the method using three experiments. **Experiment 1** shows that we can infer risk behavior with decision-theoretic models without assuming known probabilities for outcomes. **Experiment 2** demonstrates the framework in prediction tasks. **Experiment 3** studies the question of training environment models when we can only observe the selected outcomes. Finally, we provide a **Case study** of modeling risk attitudes of esports teams. The technical details such as the exact model architectures and parameters of the synthetic environment are provided in the Supplement.

We experiment primarily on simulated data to enable comparison for the classical approach of learning risk behavior from known outcome distributions. Even though we use a relatively simple parametric environment, we emphasize that it is particularly challenging for the purpose of learning risk models. We do not control the selection of the lottery sets in any way, but rather sample x_k for individual options independently. This is in stark contrast with the classical approach using experiments specifically designed to maximally expose risk behavior (Glöckner and Pachur, 2012; Stott, 2006).

Simulated Data: For Experiments 1-3 we use a simulated environment with non-linear relationship between x and y, strongly heteroscedastic noise, and the possibility to control the complexity of the environment by a single parameter m, the dimensionality of the features $x \in \mathbb{R}^m$ – for larger m the environment is naturally more difficult to learn. For reproducibility we use the environment

$$y = \frac{5}{\pi} \arcsin\left(\sin(\frac{\pi}{2}\beta_2^T x)\right) + \exp(\cos(\beta_1^T x) + 0.2)z, \tag{2}$$

where $z \sim N(0, 1)$, but note that other environments with strong heteroscedasticity could also be used. The models do not know this form, except for the baseline of learning CPT from true outcome distributions. Figure 2 illustrates the environment for $m = 1$ and in the experiments we vary m to control the difficulty of the problem.

Each lottery set has $K_i \sim \text{Unif}(3, 10)$ options with features x_{ik} sampled independently from a continuous uniform distribution within the hypercube $[-2, 3]^m$, without controlling the richness of the lottery set in any way. The users face the lottery sets in sequence but make independent choices c_i for each following their policy π.

We simulate users from multiple policies π, with primary interest in *CPT* (Section 4.1). Each CPT policy is defined by the parameters $\theta^* \in \mathbb{R}^5$ indicating the true risk behavior, sampled randomly so that λ varies in $(1, \infty)$ while all other parameters vary in the range of $(0, 1)$. We also use alternative deterministic policies (that still only depend on $P(Y|x)$) for studying prediction of behavior not conforming to the specific theory: *risk neutral* selects the option with largest median, *loss averse* maximizes the 0.1 quantiles of the options, and *reckless* non-rationally selects the option with the 2nd lowest 0.3 quantile.

5.1. Experiment 1: Interpretable Risk Models

To show that we can learn decision-theoretic models from observational data, we compare the accuracy of inferring the CPT parameters of simulated users with MONDE as the environment model, against inferring them in known environment. We use the environment (2) with $m = 1$, simulate 18 users, each following a different CPT policy, and study our ability to infer their risk behavior for cases with varying number of choices from $N_l = 64$ to $N_l = 4096$. For each user and N_l, we repeat the experiment 10 times with newly generated lottery sets. We evaluate the parameter recovery using the median absolute error over the CPT parameter vector $MAE(\theta, \theta^*) = \text{Median}(|\theta - \theta^*|)$, so that for a good accuracy at least half of the parameters need to be estimated well.

Figure 4 (left) compares the approach with MONDE as the environment model against the theoretically optimal ground truth assuming known environment, averaging MAE over both users and lottery sets. It confirms we can learn CPT from observational data alone, even from purely stochastic lottery sets, but need more data. The accuracy improves naturally as a function of available data, and importantly at the same pace for both cases. For additional insight, we also plot the accuracy for a hypothetical model that has access to all outcomes y_i (also for the options the user did not select) when training the environment model. It confirms MONDE is capable of learning the environment model with ample data, and the comparison indicates the limiting factor here is learning the environment model.

Figure 4 (right) shows the main source of variation comes from the construction of the lottery sets, confirming a well-known property of risk models (Broomell and Bhatia, 2014). It plots the distribution of errors over 100 random data sets for two example users for $N_l = 512$. The difference between the users – and even between the learnt and known environment models – is small compared to the range of errors over different lottery sets.

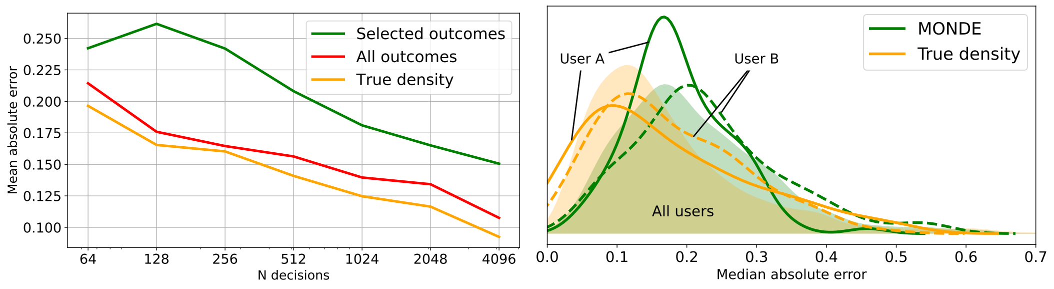

Figure 4: **Left:** If we can observe all outcomes (red), we can model the unknown environment so well that CPT parameter recovery is almost identical to the case of known environment (orange). If only observing the outcomes for selected choices we need more data but can still learn the true CPT parameters. **Right:** Almost all of the variation in parameter recovery comes from the lottery set construction, shown by wide distributions of errors for two users (solid and dashed lines), whereas the difference between users and even between estimated (green) and true (yellow) environments is small. The shaded area indicates the distribution of errors over all 25 users and 100 replicates, confirming the conclusion.

5.2. Experiment 2: Predicting Risk Behavior

For predictive tasks we consider a more challenging setup where the data is not necessarily sufficient for learning a perfect model for the environment and the policies may not match a given theory. We use the environment (2) with $m = 2$ and 4096 lottery sets for each user. We report the predictive accuracies over 1024 test lottery sets, averaging the results over 10 replications of the experiment.

We use the proposed RNN architecture with three environment models, each providing different representation for the outcomes: point prediction $\hat{y}$, a generic latent representation $f(x)$, and the quantiles $q_{\boldsymbol{p}}$. The last one uses MONDE as the architecture, whereas the other two use a generic feedforward network of similar complexity to focus on illustrating the effect of the output representation. We also matched the dimensions of the latent representations $f(x)$ and $q_{\boldsymbol{p}}$. We additionally experimented with decision models that include the original features, to show how including them can help correcting for suboptimal environment models. As baselines we consider the RNN without environment model at all, only taking x as input, and CPT using MONDE as the environment model.

Table 1 presents the prediction accuracies (median and std) for all policies and model variants. For the case of only $P(Y|X)$ as input and a true policy following a CPT the two methods are comparable in accuracy, confirming the black-box model can replicate the theory-based model when the assumptions of the latter hold. For all other true policies RNN always outperforms CPT that cannot describe them. The main observations regarding the different RNN variants are: (a) Using an environment model of any kind is highly beneficial as the baseline of $RNN(x, -)$ is clearly inferior compared to others; (b) a rich latent representation $f(x)$ is to be preferred compared to constraining the environment model to output densities $P(Y|x)$; (c) using also the input features helps.

	CPT	risk neutral	loss averse	reckless
RNN(X, -)	0.56 (0.028)	0.46 (0.067)	0.66 (0.02)	0.26 (0.021)
RNN(X, $\hat{y}$)	0.69 (0.02)	0.62 (0.051)	0.84 (0.025)	**0.42** (0.016)
RNN(X, $f(X)$)	**0.73** (0.025)	**0.76** (0.039)	**0.9** (0.022)	**0.42** (0.036)
RNN(X, $P(Y\|X)$)	0.68 (0.021)	0.4 (0.082)	0.86 (0.015)	0.3 (0.05)
RNN(-, $P(Y\|X)$)	0.65 (0.022)	0.41 (0.088)	0.83 (0.015)	0.3 (0.05)
CPT(-, $P(Y\|X)$)	0.62 (0.03)	0.26 (0.078)	0.79 (0.021)	0.22 (0.023)

Table 1: Predictive accuracy for two-dimensional environment, with true policies as columns and decision models as rows. The first term in parenthesis tells whether the decision model uses the features x as inputs and the second indicates the output representation of the environment model.

Model	Observed outcomes	Pseudo-labeling	All outcomes
RNN(-, $P(Y\|X)$)	0.65 (0.022)	0.69 (0.027)	0.75 (0.009)
RNN(x, $P(Y\|X)$)	0.68 (0.021)	0.71 (0.027)	0.76 (0.009)
CPT(-, $P(Y\|X)$)	0.62 (0.03)	0.65 (0.027)	0.75 (0.013)

Table 2: Effect of pseudo-labeling for predictive accuracy of a CPT policy. 'Pseudo-labeling' provides consistent improvement over the standard method using only the 'observed outcomes'. 'All outcomes' is a theoretical upper bound for pseudo-labeling accuracy, obtained by using also outcomes of unselected choices.

5.3. Experiment 3: Training of Environment Models

Experiment 1 revealed the inability to learn accurate environment model as the main bottleneck. Here we show how we can address it (a) by using data collected from other users for learning the environment model and (b) by pseudo-labeling of unselected options.

Sharing Data: Experiments 1 and 2 trained the environment model only on the data from the user of interest, but for shared environments we can use data from multiple users for learning the environment model. We consider the case of $N_l = 512$ choices for learning the decision model for π_u, but increasingly larger sample for training the environment model (with $m = 4$) from three alternative sources: (a) data for this user, following π_u; (b) data for *other* random CPT users π_{u_i}, each providing 512 choices; and (c) Data with half of observations from other CPT users π_{u_i} and half from users choosing uniform random. Figure 5 (plotting average error over 50 CPT profiles) shows that we learn π_u increasingly

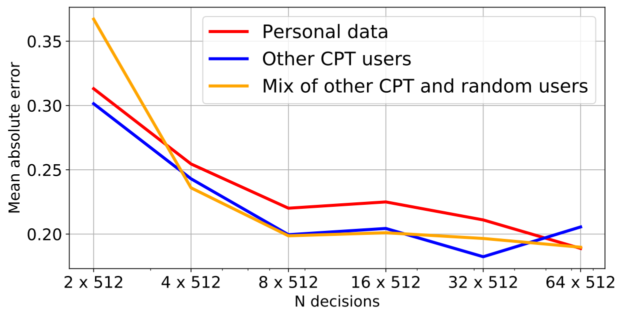

Figure 5: CPT parameter recovery for shared environment model. Data collected from users with different policies (blue), including even random behavior (yellow) is as good as data collected for this user (red).

more accurately for larger data for training the environment model also when the data was collected under other policies, until we can learn essentially perfect environment model.

Pseudo-labeling: To evaluate the value of pseudo-labeling for ensuring the environment model learns to indicate poor options as undesirable, we repeat Experiment 2 for the CPT policy and for the models using $P(Y|x)$ as outcome representation. Now, however, we train MONDE using also pseudo-labels for a subset of the unselected options, selected from $[0.3, 0.5]$ quantile range of the current model estimates.

Table 2 reveals consistent positive effect for both RNN and CPT decision models when using the proposed pseudo-labeling technique. To quantify the magnitude of the improvement, we also report the accuracy for a baseline that trains MONDE using the real outcomes for all possible actions that can here be obtained because of the artificial environment. It serves as an upper bound for what is in principle achievable (it corresponds to pseudo-labeling all unlabeled data points with perfect labels), and by using the pseudo-labels we can reduce the gap to this ideal accuracy by 23-40%. This was achieved without fine-tuning the parameters of the heuristic; e.g. cross-validation could be used to improve the accuracy.

5.4. Case Study: Risk Models in Esports

We demonstrate a typical way of using the approach by modeling the risk behavior of competitive *Counter-Strike: Global Offensive* (GS:GO) teams preparing for individual rounds. In CS:GO, players compete in teams of 5 against each other, using equipment and weapons such as guns, grenades and bullet-proof vests available from the in-game shop. A single game of CS:GO consists of 30 rounds (plus a possible overtime) and before each round the players can spend money earned during previous rounds to purchase new equipment. The purchase decisions play a significant role in the outcome of the round, especially in professional CS:GO games. The competitive teams make the decisions as a unit, and hence we consider each *team* as a user. We here present results for two top-level professional teams, *Astralis* and *Gambit Esports* to illustrate the approach, but leave more detailed analysis of risk behavior of CS:GO teams as future work. The teams competing at this level em-

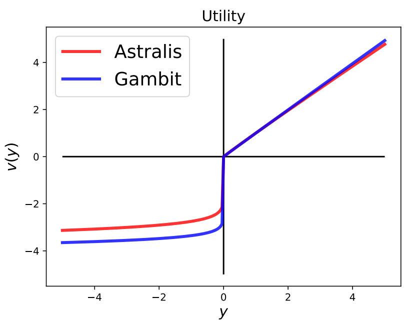

Figure 6: Utilities of two CS:GO teams, under the expected utility model.

	Astralis	Gambit
Train size	2238	2440
Test size	959	1045
EU acc.	0.387	0.548
RNN acc.	0.635	0.663
α	0.97	0.99
β	0.08	0.05
λ	2.75	3.35

Table 3: The setup and results of the CS:GO case study.

ploy professional coaches and are playing for substantial prize money, and hence we can safely assume they understand the environment well, but we need to learn the environment because the outcomes depend on the skills of the teams in a complex manner.

Setup: Each lottery set corresponds to one round where the team makes the decision of how much to invest. As individual choices we consider four discrete investment strategies ranging from very conservative saving strategy to aggressively investing all resources, corresponding to categories commonly used in CS:GO analytics. As the outcome $Y|X$ we use the sum of the money and the value of the team's equipment after the round, generally considered as strong proxy for winning probability of future rounds. This is influenced both by the result of the match as well as the investment strategy itself. As inputs for the environment model we use 12 features X describing the overall game situation. See Supplement for full description of the features and the setup.

We use MONDE as the environment model and Expected Utility (EU) as the decision model. EU is obtained as special case of CPT when $\gamma^+ = \gamma^- = 1$, and is here used for easier interpretation. Since the environment is not shared, we use team-specific environment models trained on 2,000+ lottery sets for each team, and we also evaluated the predictive accuracy for approximately $1,000$ rounds using both EU and RNN as the decision models.

Results: The results are shown in Figure 6 and Table 3. Both teams have similar risk behavior, with a small difference for losses. Both teams have low β, which indicates they do not pay significant attention to the degree of possible loss – already a small loss is considered clearly negative. We speculate that the parameters for the two teams' behaviors are similar because elite teams approach the game similarly, but also note that more detailed input features might be needed to reveal subtle differences.

For both teams EU predicts future behavior clearly better than the random baseline of 0.25 (the decisions are not balanced, but we predict shuffled indices), but not as well as RNN. This can be seen as evidence that their behavior does not match EU directly, and suggests we should not make strong conclusions based on this particular model. Note that this should not be considered as a negative result, but rather an illustration of how the approach provides information on whether the teams follow the given model or not.

6. Discussion

We considered maximally challenging setup for learning risk models, to emphasize deviation from controlled experiments. Figure 4 shows that the diversity of lottery sets greatly influences our ability to identify risk behavior, confirming previous results (Broomell and Bhatia, 2014), but also that with enough data we can infer risk behavior despite information-poor lottery sets. Since we passively observe the user, we can assume ability to collect much more data than in typical controlled experiments; in extreme cases we might be observing a user that has been playing a game daily for years. Furthermore, we do not expect most real applications to be this challenging. For instance, games are *designed* to engage the player by offering interesting choice problems. While they will not be ideal for identifying risk behavior, they still provide richer information than the random lottery sets used here.

One advantage of shared framework for theory-based and black box models is in recognizing the limitations of specific model based on differences in predictive performance. For instance, (a) if we have enough data and a user truly follows a CPT model, then $RNN(-, P(Y|x))$ and $CPT(-, P(Y|x))$ have similar predictive performance, and (b) if incorporating x as additional covariate improves predictive performance then either we have insufficient data for learning the environment model, or the user is not following any risk model but rather uses external factors (for instance, more often selecting their favorite character). Such comparisons can be used as part of a workflow for learning risk behavior from observational data, to determine whether the interpretations can be trusted.

Here learning the environment model was more challenging than learning the user model, even in low-dimensional environments. This strongly suggests using shared environment models when applicable, with pseudo-labeling as alternative for other cases. Finally, we note that the environment and decision models were trained in isolation, rather than jointly. This has the advantage of better modularity and conceptual separation, and we did not observe benefits from end-to-end training in preliminary experiments.

The case study on risk attitudes of esports teams was intended as demonstration of a prototypical use-case. For more detailed analysis, one would naturally need to consider more teams and pay more attention to careful selection of informative features for the models.

7. Conclusion

We proposed a general framework for understanding and predicting *risk behavior*, the way individuals select amongst stochastic outcomes, combining data-driven and decision-theoretic models in a new way. It enables training theory-based risk models from observational data alone, by leveraging data-driven models for the environment, while encompassing also black box predictive models. Compared to previous hybrid models it has the advantage of utilising standard decision-theoretic models as is. In summary, we provide the basis for interpreting risk behavior of individuals at scale for applications where we can passively observe a large number of choices for each individual.

Acknowledgments

This work was supported by Business Finland (project MINERAL) and the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence, FCAI). We thank the Finnish Grid and Cloud Infrastructure (FGCI) for computational resources.

References

R. P. Bhattacharyya, D. J. Phillips, B. Wulfe, J. Morton, A. Kuefler, and M. J. Kochenderfer. Multi-agent imitation learning for driving simulation. In *IROS*, pages 1534–1539, 2018.

D. D. Bourgin, J. C. Peterson, D. Reichman, S. J. Russell, and T. L. Griffiths. Cognitive model priors for predicting human decisions. In *ICML*, pages 5133–5141, 2019.

S. Broomell and S. Bhatia. Parameter recovery for decision modeling using choice data. *Decision*, 1(4):252–274, 2014.

C. Camerer and M. Weber. Recent developments in modeling preferences: Uncertainty and ambiguity. *J Risk Uncertain*, 5(4):325–370, 1992.

U. Chajewska, D. Koller, and D. Ormoneit. Learning an agent's utility function by observing behavior. In *ICML*, pages 35–42, 2001.

P. Chilinski and R. Silva. Neural likelihoods via cumulative distribution functions. In *UAI*, pages 420–429. PMLR, 2020.

I. Erev, E. Ert, O. Plonsky, D. Cohen, and O. Cohen. From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience. *Psychol. Rev.*, 124(4):369, 2017.

A. Glöckner and T. Pachur. Cognitive models of risky choice: Parameter stability and predictive accuracy of prospect theory. *Cognition*, 123(1):21–32, 2012.

J. S. Hartford, J. R. Wright, and K. Leyton-Brown. Deep learning for predicting human strategic behavior. In *NeurIPS*, pages 2424–2432, 2016.

D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.

P. Karkus, X. Ma, D. Hsu, L. Kaelbling, W. Sun Lee, and T. Lozano-Perez. Differentiable algorithm networks for composable robot learning. In *RSS*, 2019.

Y. Kolumbus and G. Noti. Neural networks for predicting human interactions in repeated games. In *IJCAI*, pages 392–399, 2019.

O. Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.*, 22:30–1, 2021.

O. Krčál, M. Kvasnička, and R. Staněk. External validity of prospect theory: The evidence from soccer betting. *J. Behav. Exp. Finance*, 65:121 – 127, 2016.

D.-H. Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML Workshop: Challenges in Representation Learning*, 2013.

G. Noti, E. Levi, Y. Kolumbus, and A. Daniely. Behavior-based machine-learning: A hybrid approach for predicting human decision making. *arXiv:1611.10228*, 2016.

J. C. Peterson, D. D. Bourgin, M. Agrawal, D. Reichman, and T. L. Griffiths. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547):1209–1214, 2021.

A. Peysakhovich and J. Naecker. Using methods from machine learning to evaluate behavioral models of choice under risk and ambiguity. *J. Econ. Behav. Organ.*, 133:373–384, 2017.

N. Phan, D. Dou, B. Piniewski, and D. Kil. A deep learning approach for human behavior prediction with explanations in health social networks: social restricted boltzmann machine SRBM+. *SNAM*, 6(1):79, 2016.

O. Plonsky, I. Erev, T. Hazan, and M. Tennenholtz. Psychological forest: predicting human behavior. In *AAAI*, pages 656–662, 2017.

N. Rabinowitz, F. Perbet, F. Song, C. Zhang, S. M. A. Eslami, and M. Botvinick. Machine theory of mind. In *ICML*, volume 80 of *PMLR*, pages 4218–4227, 2018.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

A. Rosenfeld, I. Zuckerman, E. Segal-Halevi, O. Drein, and S. Kraus. Negochat-a: a chat-based negotiation agent with bounded rationality. *Auton. Agents Multi-Agent Syst.*, 30 (1):60–81, 2016.

H. P. Stott. Cumulative prospect theory's functional menagerie. *J Risk Uncertain*, 32(2): 101–130, 2006.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2018.

J. Sydnor. (Over)insuring modest risks. *Am. Econ. J.: Appl. Econ.*, 2(4):177–199, 2010.

S. T. Trautmann and G. Van De Kuilen. Ambiguity attitudes. *The Wiley Blackwell handbook of judgment and decision making*, 1:89–116, 2015.

A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *J Risk Uncertain*, 5(4):297–323, 1992.

P. Wakker and A. Tversky. An axiomatization of cumulative prospect theory. *J Risk Uncertain*, 7(2):147–175, 1993.

X. Zhu, A. Singla, S. Zilles, and A. N. Rafferty. An overview of machine teaching. *arXiv:1801.05927*, 2018.