Machine Learning Elicitation of Risk Preferences

Aurélien Baillon[1], **Ewen Gallic**[2,3], **and Olivier L'Haridon**[4,5]

[1]emlyon business school
[2]Aix Marseille Univ, AMSE, CNRS, Marseille, France
[3]CNRS – Université de Montréal CRM - CNRS
[4]Université de Rennes, CREM
[5]Institut Universitaire de France

Journée d'Econometrie appliquée en l'honneur de Michel Terraza
Université de Montpellier
May 23, 2025

Applied Economics

- **Suppose you want:**
  - a predictive model, m,
  - for a variable of interest, y,
  - using explanatory variables, $\boldsymbol{X}$.

With Good Old-fashioned Applied Economics

- **Well, we know how to do that!**
  - use **economic theory** for the model,
  - assume a **parametric form** with parameters $\boldsymbol{\theta}$,
  - take a **dataset** of observations $\{y_i, \boldsymbol{X_i}\}_{i=1}^n$,
  - use **statistical inference methods** (MLE, GMM)
    - to estimate $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}}$,
  - Charpentier et al. (2018): key element here is **asymptotic theory** (Taylor, LLN, CLT).

Applied Decision Analysis

- **Suppose that I want:**
  - a predictive **decision model**, m,
  - for a variable of interest, y (*e.g.*, behavior),
  - using explanatory variables, $\mathbf{X}$ (*e.g.*, environment).

With Good Old-fashioned Applied Decision Analysis

- **Well, I know how to do that too!**
  - use **economic theory** for the model
    - *e.g.*, Expected Utility, EUT, $\sum p_i u(x_i)$ (Bernoulli, 1738; Von Neumann and Morgenstern, 1947),
  - assume a **parametric form**
    - *e.g.*, CARA: $u(x) = \begin{cases} (1 - e^{-\theta x})/\theta & \theta \neq 0 \\ x & \theta = 0 \end{cases}$ (Pratt, 1964),
  - take a **dataset** of observations $\{y_i, \boldsymbol{X_i}\}_{i=1}^n$
    - *e.g.*, an experiment à la Holt and Laury (2002),
  - use **statistical inference methods**.
    - MLE to estimate θ.

Good Old-fashioned Way

- **Well, sure, you know how to do that too!**
  - there exists now plenty of other ways to do it,
  - take the **dataset** $\{y_i, \boldsymbol{X_i}\}_{i=1}^n$,
  - find the **optimal** $m^\star(\mathbf{x})$,
  - with a **learning algorithm**:
    - ChatGPT-4o takes 2 seconds to advice `library(nnet)`, `library(glmtnet)`, `library(randomForest)`, `library(e1071)`, ...

Advantages of ML Techniques (1/2)

- **Less restrictive** than parameteric models,
- Easier to estimate than non-parametric models with **large amounts of data**
  - especially when the **number of parameters is higher than the number of data points**,
    - e.g.,Peysakhovich and Naecker (2017): 55,000 parameters for 2,100 data points (7 choices by 300 participants),
  - allows to (easily) manipulate **different types of data** (reaction times, psychological scales, outcomes, . . .).

Advantages of ML Techniques (2/2)

- Charpentier et al. (2018):
  - for **correctly specified linear models**, both are **substitutes**,
  - or **incorrectly specified** nonlinear models, **ML performs better**.
- Plus, ML can be also used to **renew experimental methods**
  - *e.g.*, adaptative alorigithms using Support Vector Machines (Bertani et al., 2025) or active learning (Benabbou et al., 2017),
- and for **generating novel hypotheses** about human behavior (Ludwig and Mullainathan, 2024).

ML for Decision Under Risk and Uncertainty

- Peysakhovich and Naecker (2017) use **ML to elicit a decision model** with:
  - a basis expansion of all potential decision-relevant variables (p_i's and x_i's, and their squares),
  - as well as their interactions,
  - and interactions with subject-level dummies.
- They **compare** (on test-set):
  - MLE predictions in (i) a EUT (Von Neumann and Morgenstern, 1947) or (ii) Expected Utility with non-linear probability weighting (EUP; with parameter constraints) (Tversky and Kahneman, 1992)
  - with ML predictions obtained using **regularization methods** (Lasso, Ridge)
  - for 10 lottery choices and 300 participants.

Peysakhovich and Naecker (2017): Results

- Use 7 lotteries for in-sample predictions, 3 for out-of-sample predictions,
- **Representative agent model**: ML outperforms EUT,
- **Individual-level predictions**: EUT is an in-between for lasso and ridge regression,
- ML model (rigde) seems to "learn" the EUP (behavioral model).

The ML methods we have used appear to have the same predictive power as the EUP model. However, it is not clear what functional form the ridge regression has actually learned. ***There are two hypotheses****:* ***the first is that the ML simply "re-discovers" EUP****;* ***the second is that the ML learns some function which is very different from EUP*** *but yields the same prediction error (because it does better on certain regions of the parameter space and worse on others).* (Peysakhovich and Naecker, 2017)

Psychological Random Forests

- Plonsky et al. (2017): "psychological" random forest as a synthetic ML model:
  - includes sensitivity to expected values; minimization of immediate regret; outcomes equally likely; max. prob. of gain and min. prob. of losing; pessimism; dominance,
  - outperforms standard behavioral models, e.g. BEAST (*Best Estimate and Sampling Tools*, Erev et al., 2017).
- Same conclusion on "**loss aversion**" by Saltık et al. (2023).

Is ML the New Econometrics? (1/2)

- Peterson et al. (2021): *ML offers tools to automate theory search*
  - in particular, function approximation with **deep neural networks**,
  - main difficulty: most of the experimental literature has "**small**" data.
- Using large dataset with binary choices, they found that complex decision theories exhibit better predictive performance than simpler ones:
  - best model is a mixture of EUT and Cumulative Prospect Theory, found S-shape utility, pseudo inverse S-shape probability weighting,
  - structural models perform better with small data.

Is ML the New Econometrics? (2/2)

- Zhu et al. (2025) prediction of decisions for **behavioral game-theory models** (level-k reasoning, quantal-response, risk-aversion):
  - Human choices in strategic settings: 2×2 matrix games.
  - Comparison between:
    - **a random model** (choices predicted uniformly at random),
    - **a Multi-Layer Perceptron** (benchmark),
    - **behavioral models**.
  - Results: the predictions made by the **deep neural network** are more accurate than those made by traditional **behavioral models**,
  - Modified network: **structural parameters** estimated with a **deep neural network** → results closer to benchmark.

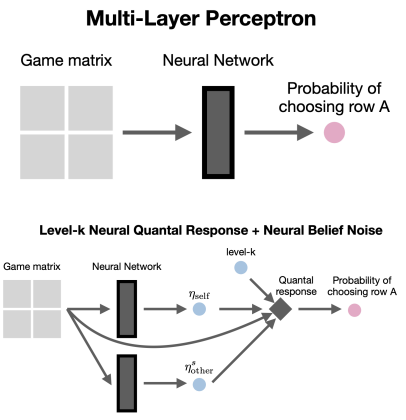


Figure 1: Models in Zhu et al. (2025)

Objectives of Our Paper

- **Main objective:** determine whether a deep learning model in which some **constraints imposed by economic theory** allows for more accurate estimations of choices made by groups of individuals compared to an **unconstrained neural network**.
- **Additionally:** explore how these different models perform when used to predict **individual choices** rather than **aggregated ones**.
- This research seeks to expand our understanding of the potential applications of integrating behavioral economics into machine learning models.

Contributions

Here, we:

1. investigate the differences between **aggregate** and **individual** data found in Peysakhovich and Naecker (2017) with a **structural model**,
2. extend Peterson et al. (2021) model to **different datatype**, e.g., **certainty equivalents** [**not presented here**],
3. extend the analysis to CPT with different datatypes [**ongoing project**].

Key Findings

Introducing a **structural layer** into a deep learning model to force it to account aspects imposed by economic theory to represent rational choice gives mixed results:

- **on aggregated Data:** a **Standard Neural Network** is slightly better than the **Neural Expected Utility Network** (the model with a structural layer) to predict choices,
- **on individual Data:** the **Neural Expected Utility Network** becomes much more competitive,
- the **Neural Expected Utility Network** exhibits **less overfitting** than the **Standard (unconstrained) Neural Network**.

Outline

2. Elicitation of Risk Preferences: Experimental Data

Setup

- How can we model and quantify the **preferences** and **decision-making process** of an individual?
- **Decision theory** routinely creates a mapping from outcomes or options to a numerical representation (the preference functional) that reflects an individual's preferences.
- When facing **uncertainty** in the outcomes, decision theory is useful: by assigning utility values to outcomes, it becomes possible to compare different choices without knowing the exact outcome.

Choice Experiments: Lotteries

- Participants in a prototypical experiment are faced with lottery choices (Savage, 1972; Kahneman and Tversky, 1979, 1984).
- They are presented with a series of choices between different lotteries.
- Each lottery consists of:
  - a set of **possible outcomes** x,
  - **associated probabilities** p.

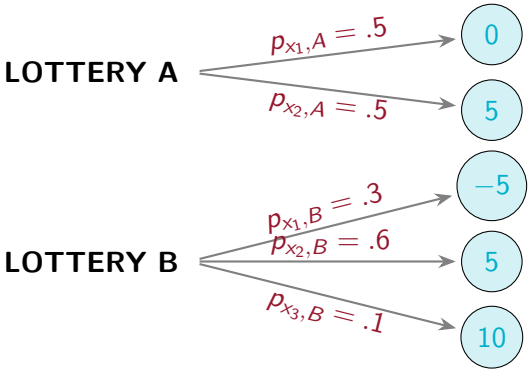


Figure 2: Example of a choice

Choice Experiments: Value

Participant choice	**Implication on $V(\cdot)$**
Selected A over B	$V(A) > V(B)$
Selected B over A	$V(A) < V(B)$
Indifferent between A and B	$V(A) = V(B)$

Table 1: Relation between participant choice and attributed values

Problem

We observe the choices made by participants, not explicit values $V(A)$ or $V(B)$.
How can we measure $V(A)$ or $V(B)$?

Choice Experiments: Value

Participant choice	**Implication on $V(\cdot)$**
Selected A over B	$V(A) > V(B)$
Selected B over A	$V(A) < V(B)$
Indifferent between A and B	$V(A) = V(B)$

Table 1: Relation between participant choice and attributed values

Problem

We observe the choices made by participants, not explicit values $V(A)$ or $V(B)$. How can we measure $V(A)$ or $V(B)$?

Random Utility Model

$$Pr[V(A) > V(B)] = \frac{e^{\eta V(A)}}{e^{\eta V(A)} + e^{\eta V(B)}}$$

- As the utility of alternat. A increases, the proba. increases and approaches 1.
- The proba. depends on the utility of alternative A according to an S-shaped curve:
  - typically, if $V(A)$ is very low (or very high), compared to $V(B)$ a small increase in utility has little impact on the probability of choice,
  - with identical utilities, probability of choice is equal to $1/2 \rightarrow$ indifferent,
  - η: low values $\rightarrow$ more random choice behaviour (i.e., high "noise"),
  - around $(1/2, 1/2)$ point: small changes in utility can strongly influence the choice,
  - any variation in value taking the decision-maker out of the indifference zone has a significant impact on the chances of choosing this option.

Choice Experiments: Functional Form

- Some economic theories provide **functional forms** to model $V(A)$ and $V(B)$.
- For example, the **Expected Utility theory** (Von Neumann and Morgenstern, 1947) assumes that:

$$V(A) = \sum_{j=1}^{N_A} u(x_{j,A}) \times p_{j,A},$$

where N_A is the number of outcomes in lottery A and $u(\cdot)$ is a utility function (of unknown form).

Choice Experiments: the Utility Function

Another Problem

We do not know the explicit form of the utility function.

- It is then possible to assume a specific parametric form:
  - e.g., exponential (CARA) utility: $u(x) = \begin{cases} (1 - e^{-\theta x})/\theta & \theta \neq 0 \\ x & \theta = 0 \end{cases}$,
  - or, power (CRRA) utility $u(x) = \begin{cases} \frac{x^{1-\theta}}{1-\theta} & \theta \neq 1 \\ ln(x) & \theta = 1 \end{cases}$,
  - where the unknown parameter(s) of the function needs to be estimated (e.g., with maximum likelihood techniques).

Choice Experiments: the Utility Function

Another Problem

We do not know the explicit form of the utility function.

- It is then possible to assume a specific parametric form:
  - e.g., exponential (CARA) utility: $u(x) = \begin{cases} (1 - e^{-\theta x})/\theta & \theta \neq 0 \\ x & \theta = 0 \end{cases}$,
  - or, power (CRRA) utility $u(x) = \begin{cases} \frac{x^{1-\theta}}{1-\theta} & \theta \neq 1 \\ ln(x) & \theta = 1 \end{cases}$,
  - where the unknown parameter(s) of the function needs to be estimated (e.g., with maximum likelihood techniques).

A popular parametric method: Holt and Laury (2002) (1/2)

- A series of binary choices between lotteries

TABLE 1—THE TEN PAIRED LOTTERY-CHOICE DECISIONS WITH LOW PAYOFFS

Option A	Option B	Expected payoff difference
1/10 of \$2.00, 9/10 of \$1.60	1/10 of \$3.85, 9/10 of \$0.10	\$1.17
2/10 of \$2.00, 8/10 of \$1.60	2/10 of \$3.85, 8/10 of \$0.10	\$0.83
3/10 of \$2.00, 7/10 of \$1.60	3/10 of \$3.85, 7/10 of \$0.10	\$0.50
4/10 of \$2.00, 6/10 of \$1.60	4/10 of \$3.85, 6/10 of \$0.10	\$0.16
5/10 of \$2.00, 5/10 of \$1.60	5/10 of \$3.85, 5/10 of \$0.10	−\$0.18
6/10 of \$2.00, 4/10 of \$1.60	6/10 of \$3.85, 4/10 of \$0.10	−\$0.51
7/10 of \$2.00, 3/10 of \$1.60	7/10 of \$3.85, 3/10 of \$0.10	−\$0.85
8/10 of \$2.00, 2/10 of \$1.60	8/10 of \$3.85, 2/10 of \$0.10	−\$1.18
9/10 of \$2.00, 1/10 of \$1.60	9/10 of \$3.85, 1/10 of \$0.10	−\$1.52
10/10 of \$2.00, 0/10 of \$1.60	10/10 of \$3.85, 0/10 of \$0.10	−\$1.85

A popular parametric method: Holt and Laury (2002) (2/2)

- The number of safe choices can be translated into bounds on utility parameter θ,
- or estimated by MLE, on the basis of, e.g., $Pr[\text{choose } A] = \frac{e^{\eta V_A}}{e^{\eta V_A} + e^{\eta V_B}}$

TABLE 3—RISK-AVERSION CLASSIFICATIONS BASED ON LOTTERY CHOICES

Number of safe choices	Range of relative risk aversion for $U(x) = x^{1-r}/(1-r)$	Risk preference classification	Proportion of choices: Low real[a]	Proportion of choices: 20x hypothetical	Proportion of choices: 20x real
0–1	$r < -0.95$	highly risk loving	0.01	0.03	0.01
2	$-0.95 < r < -0.49$	very risk loving	0.01	0.04	0.01
3	$-0.49 < r < -0.15$	risk loving	0.06	0.08	0.04
4	$-0.15 < r < 0.15$	risk neutral	0.26	0.29	0.13
5	$0.15 < r < 0.41$	slightly risk averse	0.26	0.16	0.19
6	$0.41 < r < 0.68$	risk averse	0.23	0.25	0.23
7	$0.68 < r < 0.97$	very risk averse	0.13	0.09	0.22
8	$0.97 < r < 1.37$	highly risk averse	0.03	0.03	0.11
9–10	$1.37 < r$	stay in bed	0.01	0.03	0.06

Problem

Elicitation is subject to **parametric misspecification** (Goeree and Garcia-Pola, 2023).

A popular parametric method: Holt and Laury (2002) (2/2)

- The number of safe choices can be translated into bounds on utility parameter θ,
- or estimated by MLE, on the basis of, e.g., $Pr[\text{choose } A] = \frac{e^{\eta V_A}}{e^{\eta V_A} + e^{\eta V_B}}$

TABLE 3—RISK-AVERSION CLASSIFICATIONS BASED ON LOTTERY CHOICES

Number of safe choices	Range of relative risk aversion for $U(x) = x^{1-r}/(1-r)$	Risk preference classification	Proportion of choices		
			Low real[a]	20x hypothetical	20x real
0–1	$r < -0.95$	highly risk loving	0.01	0.03	0.01
2	$-0.95 < r < -0.49$	very risk loving	0.01	0.04	0.01
3	$-0.49 < r < -0.15$	risk loving	0.06	0.08	0.04
4	$-0.15 < r < 0.15$	risk neutral	0.26	0.29	0.13
5	$0.15 < r < 0.41$	slightly risk averse	0.26	0.16	0.19
6	$0.41 < r < 0.68$	risk averse	0.23	0.25	0.23
7	$0.68 < r < 0.97$	very risk averse	0.13	0.09	0.22
8	$0.97 < r < 1.37$	highly risk averse	0.03	0.03	0.11
9–10	$1.37 < r$	stay in bed	0.01	0.03	0.06

Problem

Elicitation is subject to **parametric misspecification** (Goeree and Garcia-Pola, 2023).

Estimation Methods With Non-parametric Utility

- An alternative to parametric forms is to **estimate a number attached to each utility index**
  - e.g., Hey and Orme (1994): $u(0) = 0, u(10) = \theta, u(20) = 1$,
  - then, estimate θ by MLE with $Pr[\text{choose } A] = \frac{e^{\eta V_A}}{e^{\eta V_A} + e^{\eta V_B}}$.

Problem

Only one utility point per individual.

Estimation Methods With Non-parametric Utility

- An alternative to parametric forms is to **estimate a number attached to each utility index**
  - e.g., Hey and Orme (1994): $u(0) = 0, u(10) = \theta, u(20) = 1$,
  - then, estimate θ by MLE with $Pr[\text{choose } A] = \frac{e^{\eta V_A}}{e^{\eta V_A} + e^{\eta V_B}}$.

Problem

Only one utility point per individual.

Non-parametric Elicitation Methods

- Build the set of lotteries such as a series of indifferences provides a non-parametric elicitation of the utility function,
- Wakker and Deneffe (1996) "tradeoff" method:
  - robust to deviations from Expected Utility.

Problem

Cannot be used for any dataset: the structure of the tradeoff method needs to be respected.

Non-parametric Elicitation Methods

- Build the set of lotteries such as a series of indifferences provides a non-parametric elicitation of the utility function,
- Wakker and Deneffe (1996) "tradeoff" method:
  - robust to deviations from Expected Utility.

Problem

Cannot be used for any dataset: the structure of the tradeoff method needs to be respected.

An Elicitation Dilemma

1. We want to elicit a functional form,
2. without specifying an ad-hoc parametric form,
3. but handling many datatypes necessitates to assume a utility function.

Machine Learning may help us with that.

Choice Experiments: the Utility Function

- Here, we will use a **Neural Network** as in Peterson et al. (2021) to **encode** the utility function.
- Neural networks also have a parametric form.
- But their structure—made of **hidden layers** populated with **units** that apply **non-linear transformations** of the input data—add (a lot of) **complexity** in the parametric form.

3. Connecting a Neural Network to a Structural Model

By integrating a **structural layer** into a standard neural network that will **encode the utility**, we force the latter to take into account aspects imposed by economic theory to represent rational choice.

The parameters of the model are then estimated on observed data obtained from lab experiments thanks to a **random utility model**.

Utility Encoder **Structural Model** **Random Utility Model**

First Bloc: Utility Encoder

Encoding the Utility Function

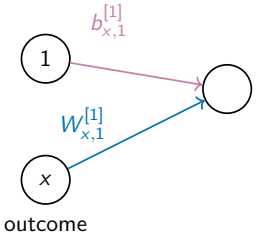
An outcome for a lottery

outcome

Encoding the Utility Function

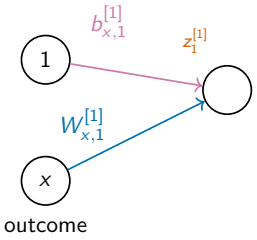
An outcome for a lottery, and a bias

Encoding the Utility Function



Linear combination, with **weights** and **bias**, both need to be estimated

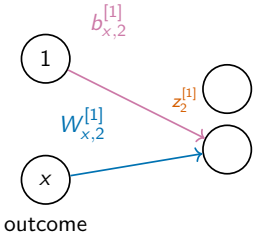
Encoding the Utility Function



The **received value** from the first hidden layer for the 1st unit writes:

$$z_1^{[1]} = x_{j,\mathcal{A}} \cdot w_{x,1}^{[1]} + b_{x,1}^{[1]} \in \mathbb{R}$$

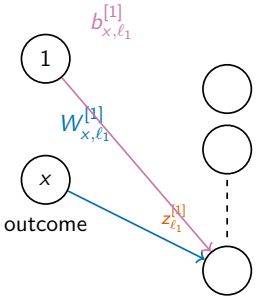
Encoding the Utility Function



A second unit receives a **linear combination** with different **weights**:

$$z_1^{[1]} = x_{j,\mathcal{A}} \cdot w_{x,1}^{[1]} + b_{x,1}^{[1]} \in \mathbb{R}$$

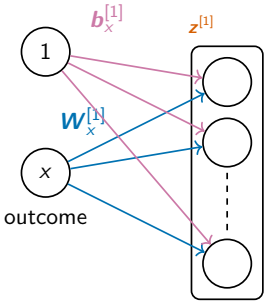
Encoding the Utility Function



The last unit of the layer receives a **linear combination** with different **weights**:

$$z^{[1]}_{\ell_1} = x_{j,\mathcal{A}} \cdot w^{[1]}_{x,\ell_1} + b^{[1]}_{x,\ell_1} \in \mathbb{R}$$

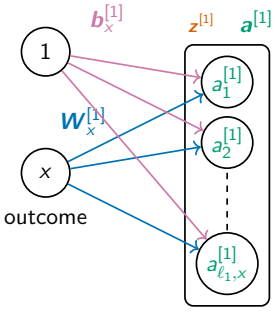
Encoding the Utility Function



The **received values** from the first hidden layer for the ℓ_1 units write:

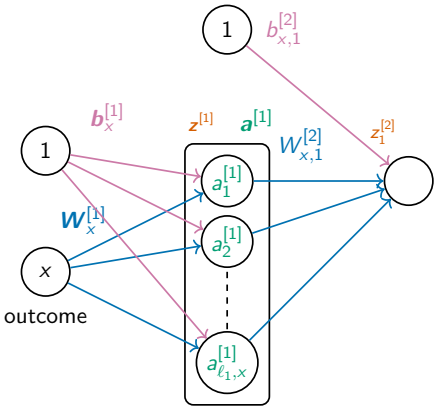
$$z^{[1]} = x_{j,\mathcal{A}} \cdot w_x^{[1]} + b_x^{[1]} \in \mathbb{R}^{1 \times \ell_1}$$

Encoding the Utility Function



An activation function $g(\cdot)$ is applied to each unit of the first layer: $\boldsymbol{a}^{[1]} = g(\boldsymbol{z}^{[1]}) \in \mathbb{R}^{1 \times \ell_1}$

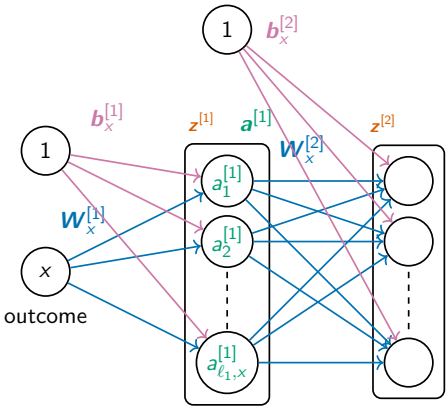
Encoding the Utility Function



These values are then the input from the second hidden layer, such that the **received values** for the 1st unit in layer 2 writes:

$$z^{[2]}_1 = a^{[1]} \cdot w^{[2]}_{x,1} + b^{[2]}_{x,1} \in \mathbb{R}$$

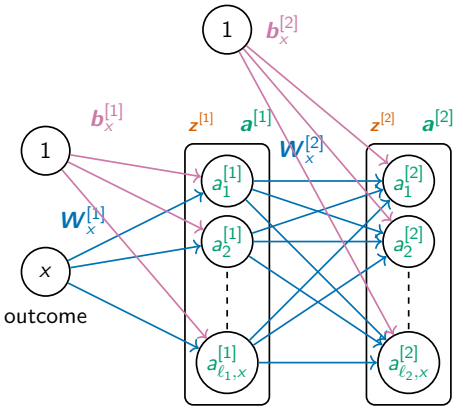
Encoding the Utility Function



The received values in the second hidden layer for the ℓ_2 units write:

$$\boldsymbol{z}^{[2]} = \boldsymbol{a}^{[1]} \cdot \boldsymbol{w}_x^{[2]} + \boldsymbol{b}_x^{[2]} \in \mathbb{R}^{1 \times \ell_2}$$

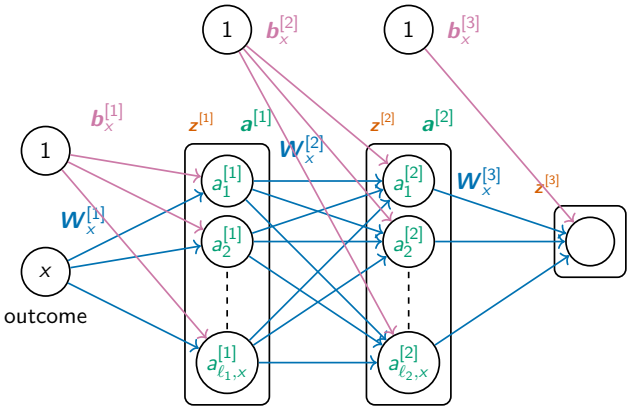
Encoding the Utility Function



The activation function $g(\cdot)$ is then applied to each unit from that second hidden layer:

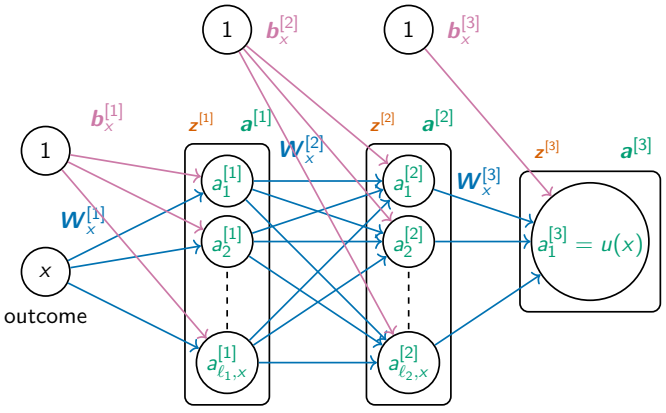
$$\boldsymbol{z}^{[2]} = \boldsymbol{a}^{[1]} \cdot \boldsymbol{w}_x^{[2]} + \boldsymbol{b}_x^{[2]} \in \mathbb{R}^{1 \times \ell_2}$$

Encoding the Utility Function



These activated values are then the input of the output layer of the first part of the neural network, the one that returns the utility $u(x_{j,\mathcal{G}})$ (thus made of a single unit) such that: $z^{[3]} = a^{[2]} w_x^{[3]} + b_x^{[3]} \in \mathbb{R}^{1\times 1}$

Encoding the Utility Function



The activation function $g(\cdot)$ is then applied to each unit from that second hidden layer.

This activation function is the identity function:

$a^{[3]} = I(z^{[3]}) := u(x) \in \mathbb{R}^{1\times 1}$

Structure of the Utility Encoder Part

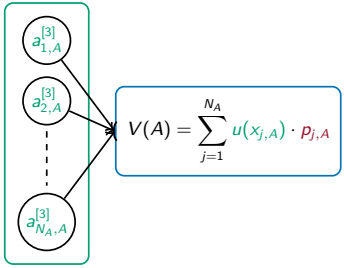
- We only consider 1 hidden layer, with 10 units.
- The activation function in the hidden layer is a sigmoid.
- The activation function that outputs $u(x)$ is the identity function.

Second Bloc: Structural Model

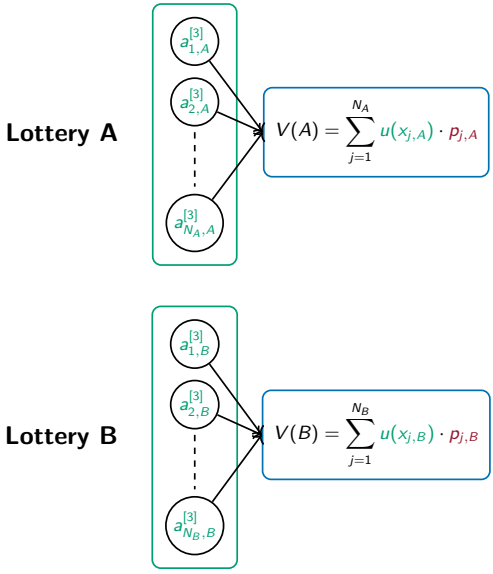
Utility Encoder | **Structural Model** | Random Utility Model

Connecting with the Structural Model

Lottery A



Connecting with the Structural Model



Third Bloc: Random Utility Model

Utility Encoder

Structural Model

Random Utility Model

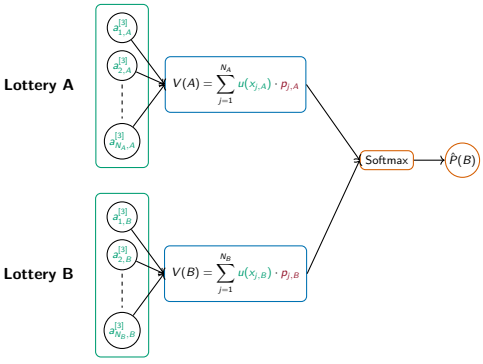
Comparing Predictions and Observed Values

Lastly, a sigmoid function is applied to $\begin{bmatrix} V_A & V_B \end{bmatrix}$ to estimate the probability of selecting prospect B:

$$\hat{Pr}[\text{Choose } B] = \frac{1}{1 + e^{-\eta(V(B)-V(A))}},$$

where η is the noise parameter (also called temperature) that needs to be estimated.

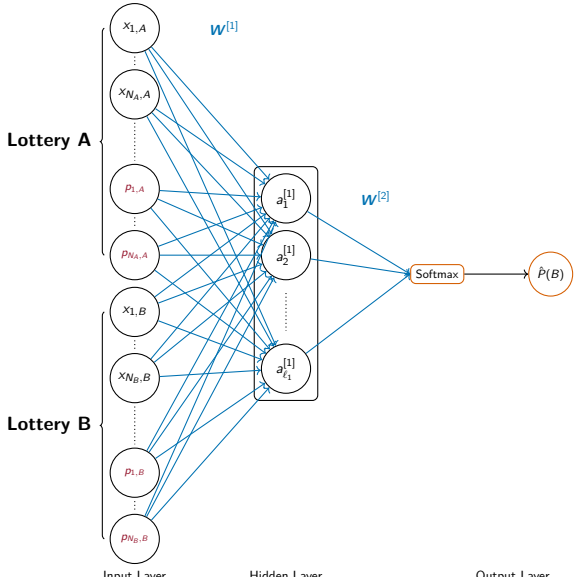
Comparing Predictions and Observed Values



Simple Fully Connected Neural Network

What about using a **Simple Neural Network**, **without constraints**, to predict $Pr[\text{choose } B]$?

Simple Fully Connected Neural Network



Simple Fully Connected Neural Network: Input

We consider 6 different configurations for the Input layer:

Config.	$x_{A,B}$	$p_{A,B}$	N_A	N_B	N	$\mathbb{E}_{A,B}$	$\mathrm{Var}_{A,B}$	$\tilde{\mu}^3_{A,B}$	**Ranks**
1	X	X							
2	X	X	X	X	X				
3		X	X	X	X				X
4	X	X	X	X	X	X			
5	X	X	X	X	X	X	X		
6	X	X	X	X	X	X	X	X	

4. Data

Three Samples

The economic experiment data come from three sources:

- Peterson et al. (2021),
- Hey and Orme (1994),
- Baillon et al. (2020).

The participants are asked to state which prospect they prefer between A or B. The **target variable** depends on the type of data:

- for **individual data**: $y \in \{0, 1\}$ ($y = 1$ if the participant chose lottery B),
- for **aggregated data**: $y \in [0, 1]$ ($y = 1$ if all the participants chose lottery B).

Example from Peterson et al. (2021)

Each participant is subjected to a **lottery choice** of the following form:

	Prospect A (2 options)		Prospect B (3 options)		
Outcomes	37	8	87.5	86.5	-31
Probabilities	0.05	0.95	0.125	0.125	0.75
Rank	0.05	1	0.125	0.25	1

For every binary lottery decision, the percentage of participants opting for lottery B is shown (no individual choices are accessible). Here:

Frequency gamble B selected: 0.22.

Data: Peterson et al. (2021)

- 9,831 lottery choices (complete information, no ambiguity),
- only access to aggregated data,
- 1 or 2 outcomes in lottery A; 1 or up to 9 outcomes in lottery B.

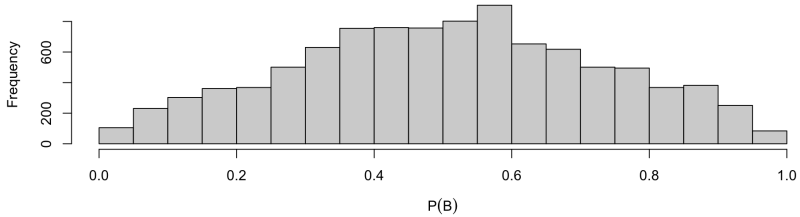


Figure 3: Distribution of the proportions for lottery B choices, Peterson et al. (2021) data.

Data: Hey and Orme (1994)

- 100 lottery choices,
- 80 participants,
- 4 outcomes in each lottery.

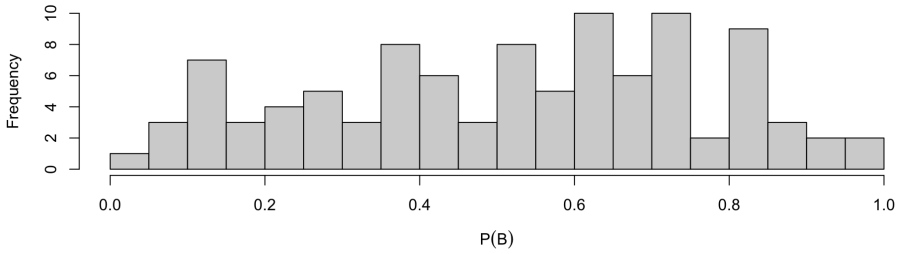


Figure 4: Distribution of the proportions for lottery B choices, Hey and Orme (1994) data.

Data: Hey and Orme (1994)

- 100 lottery choices,
- 80 participants,
- 4 outcomes in each lottery.

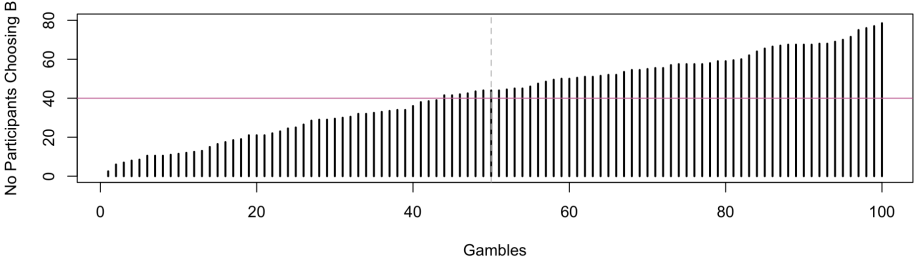


Figure 4: No. of participants who chose B in each lottery choice, Hey and Orme (1994) data.

Data: Baillon et al. (2020)

- 70 lottery choices,
- 139 participants,
- 1 to 4 outcomes in lottery A, 2 to 4 outcomes in lottery B.

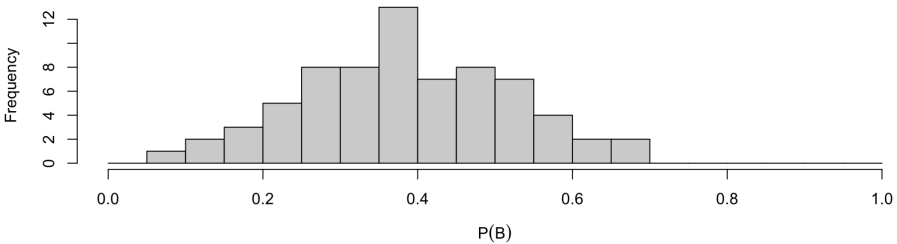


Figure 5: Distribution of the proportions for lottery B choices, Baillon et al. (2020) data.

Data: Baillon et al. (2020)

- 70 lottery choices,
- 139 participants,
- 1 to 4 outcomes in lottery A, 2 to 4 outcomes in lottery B.

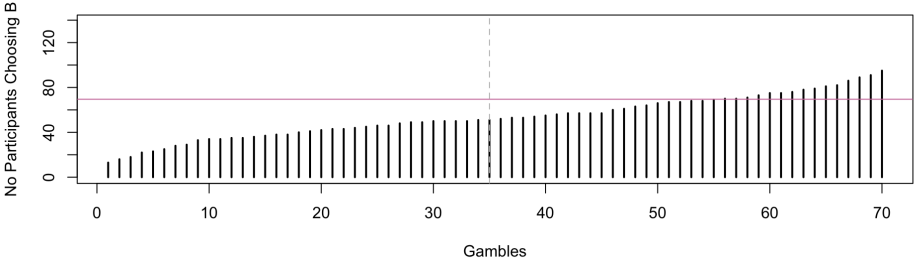


Figure 5: No. of participants who chose B in each lottery choice, Hey and Orme (1994) data.

5. Simulations

Setup

- We estimate the following **models**:
  - **Benchmark** (Mean choice rate observed in the data),
  - **Neural Expected Utility Network (NN-EU)**
  - **Simple Neural Network** (with 6 configurations).
- We use data from:
  - Peterson et al. (2021) (aggregated only),
  - Hey and Orme (1994) (aggregated, and individual),
  - Baillon et al. (2020) (aggregated, and individual).
- Each model is estimated **200 times** on random splits of the data into a **train set** (80%) and a **test set** (20%).

Creation of the Train and Test Data

- For **aggregated data**:
  - 80% of the **gambles** are randomly drawn from the dataset and used to **train the models**.

Creation of the Train and Test Data

- For **individual date**:
  - We first draw 80% of the **participants** to be in the **train set**,
  - We also draw 80% of the **gambles** to be in the **train set**.

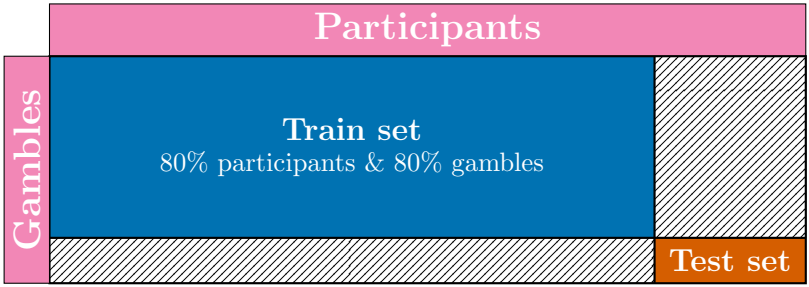


Figure 6: Train/Test split for individual data

Goodness of Fit

- To assess the goodness of fit of each model, we compute the Mean Squared Error (MSE).
- Fo each model, MSE are computed on the **train set** and on the **test set**.
- The MSE can be compared to a **benchmark**, which simply consists in using the **mean choice rate observed on the train set** to make predictions.

6. Results

Aggregated Data: Goodness of Fit (Peterson et al., 2021)

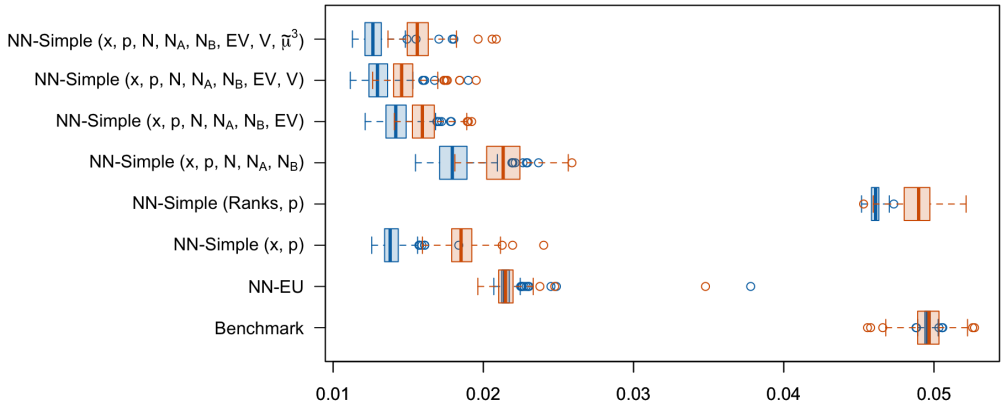


Figure 7: MSE, 200 replications, Peterson et al. (2021) aggregated data. [**train set**, **test set**].

Aggregated Data: Goodness of Fit (Hey and Orme, 1994)

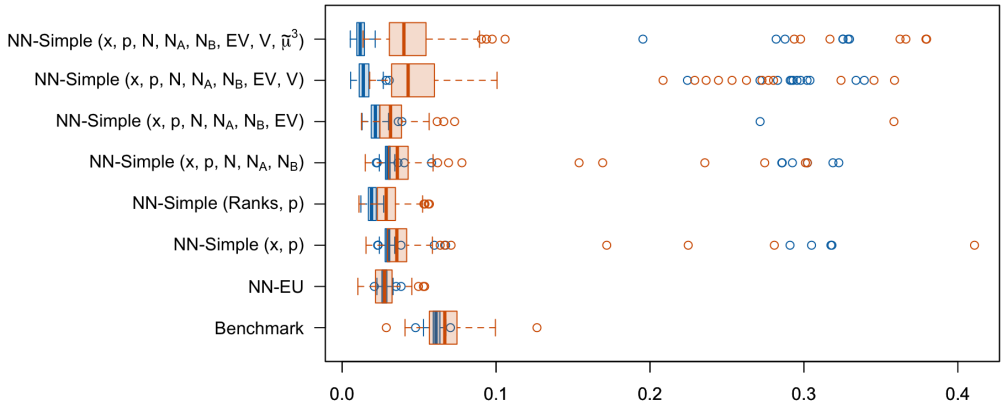


Figure 8: MSE, 200 replications, Hey and Orme (1994) aggregated data. [**train set**, **test set**].

Aggregated Data: Goodness of Fit (Baillon et al., 2020)

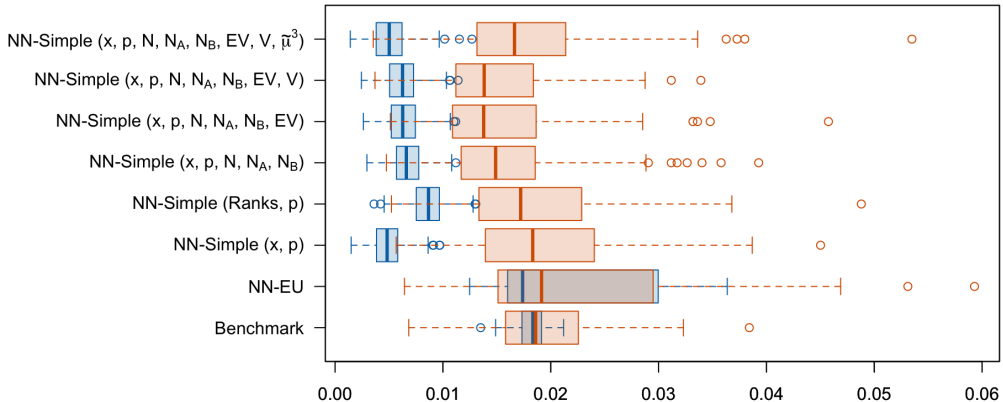


Figure 9: MSE, 200 replications, Baillon et al. (2020) aggregated data. [**train set**, **test set**].

Aggregated Data: Utility Function (Peterson et al., 2021)

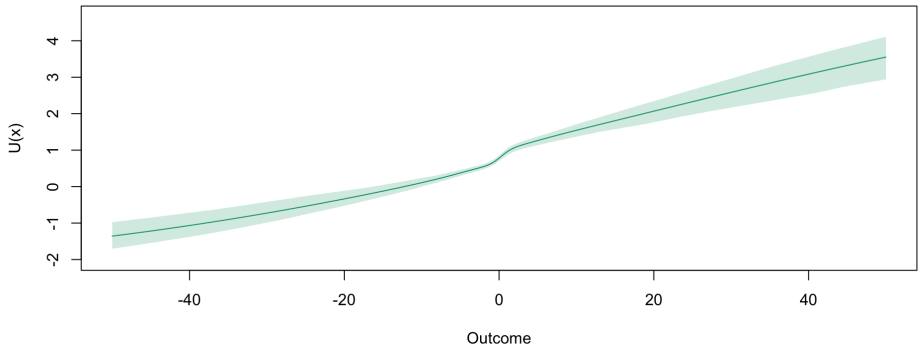


Figure 10: Utility function from the 200 replications of the estimation of the **Neural Expected Utility Model**, using Peterson et al. (2021), aggregated data. **Green bands:** 90% interval defined by quantiles. The values of the utility are centered around 0.

Aggregated Data: Utility Function (Hey and Orme, 1994)

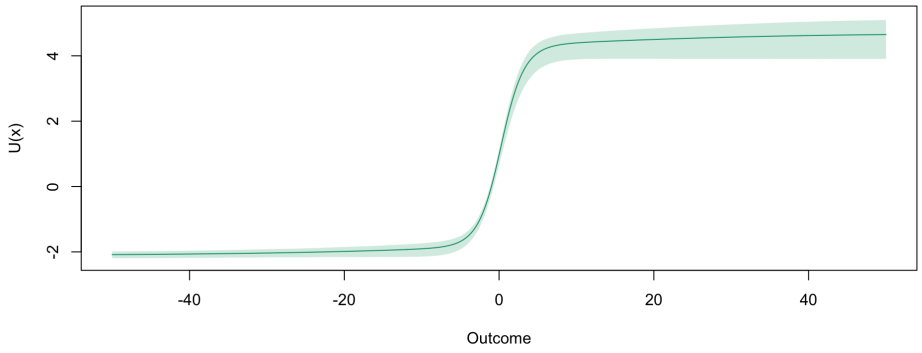


Figure 11: Utility function from the 200 replications of the estimation of the **Neural Expected Utility Model**, using Hey and Orme (1994), aggregated data. **Green bands:** 90% interval defined by quantiles. The values of the utility are centered around 0.

Aggregated Data: Utility Function (Baillon et al., 2020)

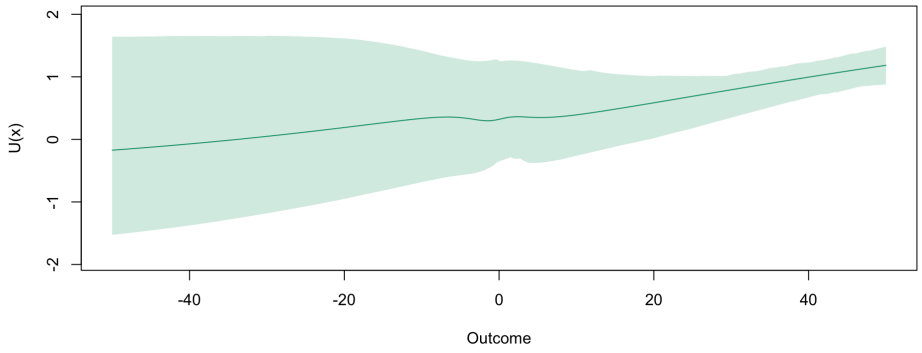


Figure 12: Utility function from the 200 replications of the estimation of the **Neural Expected Utility Model**, using Baillon et al. (2020), aggregated data. **Green bands:** 90% interval defined by quantiles. The values of the utility are centered around 0.

Individual Data: Goodness of Fit (Hey and Orme, 1994)

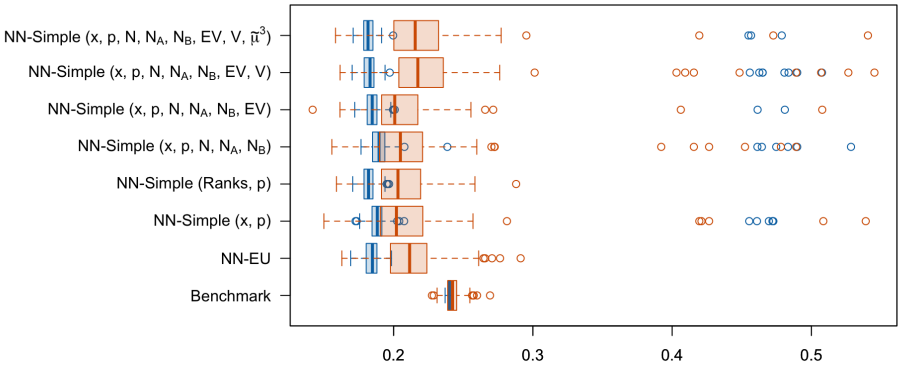


Figure 13: MSE, 200 replications, Hey and Orme (1994) individual data. [**train set**, **test set**].

Individual Data: Goodness of Fit (Baillon et al., 2020)

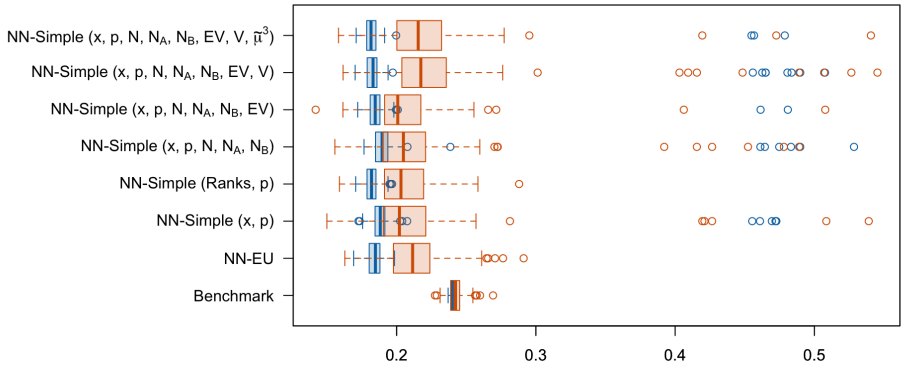


Figure 14: MSE, 200 replications, Baillon et al. (2020) individual data. [**train set**, **test set**].

Individual Data: Utility Functions (Hey and Orme, 1994)

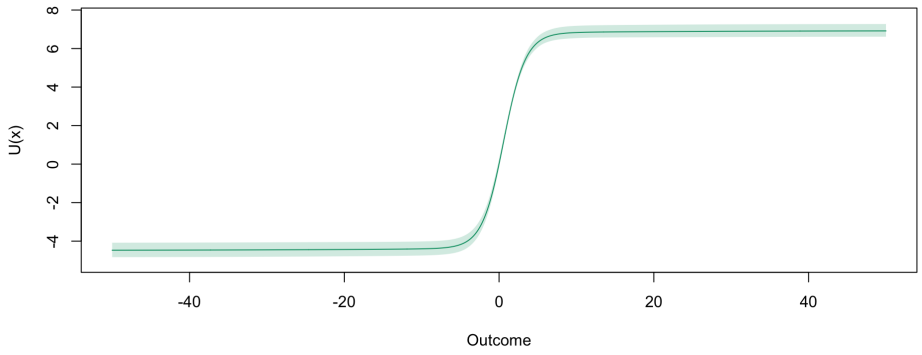


Figure 15: Utility function from the 200 replications of the estimation of the **Neural Expected Utility Model**, using Hey and Orme (1994), aggregated data. **Green bands:** 90% interval defined by quantiles. The values of the utility are centered around 0.

Individual Data: Utility Functions (Baillon et al., 2020)

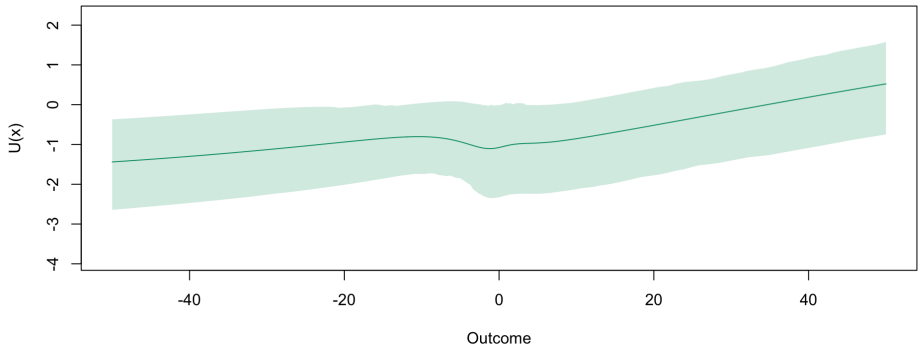


Figure 16: Utility function from the 200 replications of the estimation of the **Neural Expected Utility Model**, using Baillon et al. (2020), aggregated data. **Green bands:** 90% interval defined by quantiles. The values of the utility are centered around 0.

7. Conclusion

Wrap-up

- We investigate the differences between **aggregate** and **individual** data with **structural model**.
- **On aggregated data:** a **Standard Neural Network** is slighlty better than the **Neural Expected Utility Network** to predict choices
- **On individual data:** the **Neural Expected Utility Network** becomes much more competitive.
- The **Neural Expected Utility Network** exhibits **less overfitting** than the **Standard Neural Network**.

Agenda

- Solving the problem with the **inversion of the utility function** obtained with the **constrained Neural Network** to extend our work to **certainty equivalents**.
- Consider **nonlinearities** in the probabilities in the EU: Expected Utility with probability weighting à la Kahneman and Tversky (1979).
- Creation of a `R` package to use the developed models.

Thank you!

Comments are welcome!

Aurélien Baillon
baillon@em-lyon.com

Ewen Gallic
ewen.gallic@univ-amu.fr

Olivier L'Haridon
olivier.lharidon@univ-rennes.fr

8. Appendix

References I

Baillon, A., Bleichrodt, H., and Spinu, V. (2020). Searching for the reference point. *Management Science*, 66(1):93–112.

Benabbou, N., Perny, P., and Viappiani, P. (2017). Incremental elicitation of choquet capacities for multicriteria choice, ranking and sorting problems. *Artificial Intelligence*, 246:152–180.

Bernoulli, D. (1738). Specimen theoriae novae de mensura sortis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, pages 175–192.

Bertani, N., Boukhatem, A., Diecidue, E., Perny, P., and Viappiani, P. (2025). Fast and simple adaptive elicitations. Available at SSRN: https://ssrn.com/abstract=3569625.

Charpentier, A., Flachaire, E., and Ly, A. (2018). Econometrics and machine learning. *Economie et Statistique*, 505(1):147–169.

Erev, I., Ert, E., Plonsky, O., Cohen, D., and Cohen, O. (2017). From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience. *Psychological Review*, 124(4):369–409.

Goeree, J. K. and Garcia-Pola, B. (2023). A non-parametric test of risk aversion. *arXiv preprint arXiv:2308.02083*.

Hey, J. D. and Orme, C. (1994). Investigating generalizations of expected utility theory using experimental data. *Econometrica: Journal of the Econometric Society*, pages 1291–1326.

References II

Holt, C. A. and Laury, S. K. (2002). Risk aversion and incentive effects. *American economic review*, 92(5):1644–1655.

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

Kahneman, D. and Tversky, A. (1984). Choices, values, and frames. *American psychologist*, 39(4):341.

Ludwig, J. and Mullainathan, S. (2024). Machine learning as a tool for hypothesis generation. *The Quarterly Journal of Economics*, 139(2):751–827.

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

Peysakhovich, A. and Naecker, J. (2017). Using methods from machine learning to evaluate behavioral models of choice under risk and ambiguity. *Journal of Economic Behavior & Organization*, 133:373–384.

Plonsky, O., Erev, I., Hazan, T., and Tennenholtz, M. (2017). Psychological forest: Predicting human behavior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Pratt, J. W. (1964). Risk aversion in the small and in the large. *Econometrica*, 32(1/2):122.

References III

Saltık, Ö., Söyü, R., Değirmen, S., Şengönül, A., et al. (2023). Predicting loss aversion behavior with machine-learning methods. *Humanities and Social Sciences Communications*, 10(1):1–14.

Savage, L. J. (1972). *The foundations of statistics*. Courier Corporation.

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

Von Neumann, J. and Morgenstern, O. (1947). Theory of games and economic behavior, 2nd rev.

Wakker, P. and Deneffe, D. (1996). Eliciting von neumann-morgenstern utilities when probabilities are distorted or unknown. *Management science*, 42(8):1131–1150.

Zhu, J.-Q., Peterson, J. C., Enke, B., and Griffiths, T. L. (2025). Capturing the complexity of human strategic decision-making with machine learning. *Nature Human Behaviour*.

10. Inverting the Utility Function

Two lotteries

Consider two lotteries:

Lottery A

- 50% chance of obtaining 0,
- 50% chance of obtaining 10.

Lottery B

- 10% chance of obtaining -5,
- 90% chance of obtaining 20.

```
x_A <- c(0, 10)
x_B <- c(-5, 20)
p_A <- c(.5, .5)
p_B <- c(.1, .9)
```

Utility with the NN-EU

The NN-EU gives us a utility function which can be applied:

```
u_A <- utility_function(x_A)
u_B <- utility_function(x_B)
rbind(u_A, u_B)
```

```
          [,1]      [,2]
u_A   0.0000000  9.563202
u_B  -0.4889169 18.303310
```

Contribution of each lottery to the EU

Each lottery's contribution to the expected utility:

```
z_values_A <- u_A * p_A
z_values_B <- u_B * p_B
rbind(z_values_A, z_values_B)
```

```
                  [,1]      [,2]
z_values_A  0.00000000  4.781601
z_values_B -0.04889169 16.472979
```

The expected utility for each lottery, EU:

```
z_val_A <- sum(z_values_A)
z_val_B <- sum(z_values_B)
c(z_val_A, z_val_B)
```

```
[1]  4.781601 16.424087
```

Inversion of the utility function

- The **certainty equivalent** writes:

$$CE = u^{-1}(EU)$$

- Our approach: using the `uniroot()` function, which calculates the inverse value of another function within a monotonous range.
- This approach is not suitable for non-bijective functions. We therefore partition the sequences of values within the domain of the function to obtain splits of **monotonous subsequences**.

Inversion of the utility function

```r
#' Splits a sequence of numerical values into sequences of monotonic sequences
#'
#' @param seq_x vector of numerical values to use to create the splits
#' @param seq_y corresponding values to put in the sequences
#' @export
#' @importFrom zoo na.locf
split_monotonic <- function(seq_x, seq_y) {
  seq_x_diff <- diff(seq_x)
  seq_x_diff[seq_x_diff == 0] <- NA
  seq_x_diff_replaced <- zoo::na.locf(seq_x_diff, fromLast = TRUE)

  n_obs_diff <- length(seq_x_diff)
  n_obs_diff_replaced <- length(seq_x_diff_replaced)
  # Last obs might have been removed if equal to 0
  if(n_obs_diff_replaced < n_obs_diff) {
    n_to_replace <- n_obs_diff - n_obs_diff_replaced
    seq_x_diff_replaced <- c(
      seq_x_diff_replaced,
      rep(seq_x_diff_replaced[n_obs_diff_replaced], n_to_replace)
    )
  }

  seq_x_rle <- rle(sign(seq_x_diff_replaced))
  if (length(seq_x_rle$lengths) > 1) {
    groups <- rep(seq_along(seq_x_rle$values), seq_x_rle$lengths)
    groups <- c(1, groups)
    splits <- split(seq_y, groups)
  } else {
    splits <- list(`1` = seq_y)
  }
  increasing_val <- seq_x_rle$values > 0
  list(
    splits = splits,
    increasing_val = increasing_val
  )
}
```

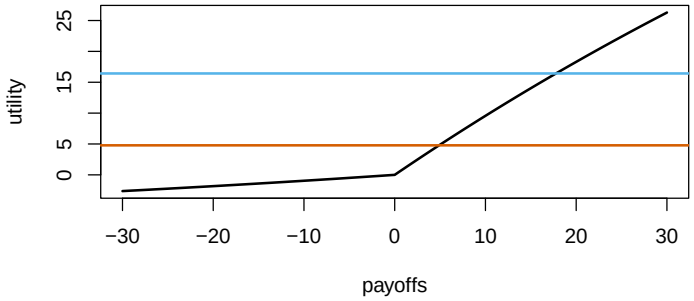
Inversion of the utility function

Sequences of outcomes and corresponding utilities:

```
outcomes_range <- c(-30, 30)
xs <- seq(outcomes_range[1], outcomes_range[2], by = .5)
ys <- utility_function(xs)
```

Inversion of the utility function

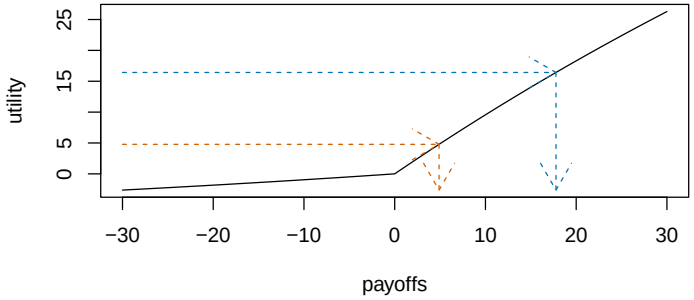
We aim to determine the antecedent(s) of 4.7816008, representing the **expected utility of lottery A**, as well as the antecedent(s) of 16.4240872, which corresponds to the **expected utility of lottery B**.



Utility function.

Inversion of the utility function

We aim to determine the antecedent(s) of 4.7816008, representing the **expected utility of lottery A**, as well as the antecedent(s) of 16.4240872, which corresponds to the **expected utility of lottery B**.



Utility Function: Certainty Equivalents of Lottery A and B.