Online Appendix

This Online Appendix contains related materials for the submission of the article "Search costs and adaptive consumers: short time delays do not affect choice quality".

Contents

A Utility calculation based on conjoint analysis

A.1 Linear utility model

For calculating utilities from a linear model, a very basic linear model is used. Let

$$u(\mathbf{x}) = \sum_{k=1}^{m} w_k x_k \tag{1}$$

be the utility derived from product $\mathbf{x}$ that is characterized by m attributes, where x_k denotes the value of attribute k and w_k (s.t. $\sum_k w_k = 1$) the weight the decision maker puts on attribute k. Whereas x_k can directly used from the product data used in the Mouselab study, w_k must be estimated from the choice-based conjoint experiment. First, I will explain the procedures used for estimating attribute weights. Second, I will illustrate how the utility values were calculated for the products in the Mouselab choice set.

A.1.1 Estimating attribute weights

Let $p_{\mathbf{xx}'}$ be the probability that alternative $\mathbf{x}$ is chosen over alternative $\mathbf{x}'$ and

$$p_{\mathbf{xx}'} = Pr[u_{\mathbf{x}} \geq u_{\mathbf{x}'}] \tag{2}$$

with

$$u_{\mathbf{x}} = v_{\mathbf{x}} + \epsilon_{\mathbf{x}} \tag{3}$$

$$u_{\mathbf{x}'} = v_{\mathbf{x}'} + \epsilon_{\mathbf{x}'} \tag{4}$$

where $v_{\mathbf{x}}$ and $v_{\mathbf{x}'}$ denote the deterministic valuation of alternative $\mathbf{x}$ and $\mathbf{x}'$, respectively; $\epsilon_{\mathbf{x}}$ and $\epsilon_{\mathbf{x}'}$ represent stochastic error terms. Since the error terms are assumed to be independently Gumbel distributed random variables (see McFadden, 1974) one can the rewrite (2)

$$Pr[u_{\mathbf{x}} \geq u_{\mathbf{x}}] = Pr[\epsilon_{\mathbf{x}'} - \epsilon_{\mathbf{x}} \leq v_{\mathbf{x}} - v_{\mathbf{x}'}] = \frac{1}{1 + e^{-(v_{\mathbf{x}} - v_{\mathbf{x}'})}} \tag{5}$$

This implies

$$v_{\mathbf{x}} - v_{\mathbf{x}'} = ln\left(\frac{p_{\mathbf{xx}'}}{1 - p_{\mathbf{xx}'}}\right). \tag{6}$$

which describes the standard logit model. Therefore I estimated the following econometric specification for all pairwise comparisons per product category obtaining $\boldsymbol{\beta}$, the vector of attribute-specific parameters.

$$ln\left(\frac{p_{\mathbf{xx}'}}{1 - p_{\mathbf{xx}'}}\right) = \sum_{k=1}^{m} \beta_k (x_k - x'_k) \tag{7}$$

As the scales of attribute values were different across attributes, the absolute values of the obtained regression coefficients $\boldsymbol{\beta}$ could not be interpreted as attribute weights directly, but needed to be adjusted in size. Let $\underline{x}_k$ and $\overline{x}_k$ denote the overall lowest and the overall highest value of attribute k in the set of all attribute values of all alternatives used for the pairwise comparisons, respectively. Then the vectors $\underline{\mathbf{x}}$ and $\overline{\mathbf{x}}$ describe the lowest value and the highest value that could be induced by the attribute values used in the pairwise comparisons.[1] Multiplying the lowest and the highest attribute values with the estimated $\boldsymbol{\beta}$ provides us with the maximum range of valuation changes that can be induced by the particular attribute values used in the conjoint choice task $\Delta\mathbf{v}$.

$$\underline{\mathbf{v}} = \sum_{k=1}^{m} \beta_k \underline{x}_k \tag{8}$$

$$\overline{\mathbf{v}} = \sum_{k=1}^{m} \beta_k \overline{x}_k \tag{9}$$

$$\Delta\mathbf{v} = |\overline{\mathbf{v}} - \underline{\mathbf{v}}| \tag{10}$$

Dividing the maximum variation in the valuation of a product induced by the maximum change of the value of attribute k, i.e. $\Delta\mathbf{v} = \Delta v_k|_{k\in\{1,\dots,m\}}$, by the sum of the maximum variation induced by all attributes $k \in \{1,\dots,m\}$ results in the measure of attribute weights that satisfies $\sum_k w_k = 1$.

$$w_k = \frac{\Delta v_k}{\sum_{k=1}^{m} \Delta v_k} \tag{11}$$

A.1.2 Calculating utility

Let

$$\widetilde{\mathbf{x}} = \frac{\mathbf{x} - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} \tag{12}$$

be a vector attribute values that is rescaled to values on the interval $[0,1]$. One can then rewrite (1)

$$u(\mathbf{x}) = \sum_{k=1}^{m} w_k \widetilde{x}_k \tag{13}$$

to obtain utility values for every alternative in the choice set.

A.2 Lexicographic utility model

In the following, the procedures for estimating the lexicographic utility model are explained in detail. First, I will outline how to infer attribute ranks, i.e. establish the lexicographic order of attributes. Second, I will describe how these attribute ranks were used to compute a lexicographic utility measure for the choice sets used in the Mouselab part of the study.

[1] Consider a case of three notebook computers, characterized by four attributes (CPU, RAM, display size and price) each. NB_1 ={2 GHz, 4 GB, 15.6 inch, €1199}, NB_2 ={2.1 GHz, 4 GB, 16.0 inch, €1099} and NB_3 ={2.4 GHz, 2 GB, 15.4 inch, €999}. Then the best possible combination of attribute values (equivalent to $\overline{\mathbf{x}}$) would be $\overline{NB}$ = {2.4 GHz, 4 GB, 16.0 inch, €999} and the worst possible combination (equivalent to $\underline{\mathbf{x}}$) would be $\underline{NB}$ ={2 GHz, 2 GB, 15.4 inch, €1199}. The actual choice set did not contain clearly dominating or dominated products, i.e. the products described by $\underline{\mathbf{x}}$ and $\overline{\mathbf{x}}$ were not actually used for the pairwise comparison.

A.2.1 Inferring attribute ranks

I used the algorithm developed by Kohli and Jedidi (2007) to infer lexicographic rules for each individual participant. The subsequent section closely resembles Kohli and Jedidi's (2007) description of the algorithm which I programmed in R.

Initialization step. Let

$$S_0 = \{1, \dots, m\} \tag{14}$$

denote the set of all attributes.[2] Further, let n_k denote the number of different levels attribute k has in its attribute values.[3] Arrange the levels of each attribute $k \in S_0$ in increasing preference order, and assign to them the sequence of integers $0, \dots, n_k - 1$. Each level of attribute $k \in S_0$ is thus identified by a unique value $x_k \in \{0, \dots, n_k - 1\}$.[4] Let $\mathbf{x} = (x_1, \dots, x_m)$ denote a product profile in which attribute $k \in S_0$ has the level associated with x_k. For each alternative $\mathbf{x}$, compute

$$u_{1k}(\mathbf{x}) = \frac{x_k}{n_k}, \forall k \in S_0. \tag{15}$$

Let Ω denote the set of all pairwise comparisons $(\mathbf{x}, \mathbf{x}')$ in which $\mathbf{x}$ is preferred to $\mathbf{x}'$. For each each $(\mathbf{x}, \mathbf{x}')$, for all $\forall k \in S_0$ compute

$$d_{1k}(\mathbf{x}, \mathbf{x}') = \begin{cases} 1 & \text{if } u_{1k}(\mathbf{x}) < u_{1k}(\mathbf{x}') \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Remember, it was defined above that a product $\mathbf{x}$ is preferred over $\mathbf{x}'$. Decision makers are therefore assumed to choose $\mathbf{x}$ from the choice set $(\mathbf{x}, \mathbf{x}')$. As d_{1k} (and later d_{tk}, in the recursive steps) indicates that a product's attribute value k of product $\mathbf{x}'$ was preferred over the value of the same attribute in $\mathbf{x}$ (d_{1k} is computed for all pairwise comparisons), non-zero values for d_{1k} can be seen as *penalizing attributes*. In other words, alternative $\mathbf{x}$ was chosen despite the fact that the decision maker preferred the value of attribute $k \in S_0$ from alternative $\mathbf{x}'$ over the corresponding attribute value of alternative $\mathbf{x}$. Let

$$Z_{1k} = \sum_{(\mathbf{x}, \mathbf{x}') \in \Omega} d_{1k}(\mathbf{x}, \mathbf{x}'), \forall k \in S_0 \tag{17}$$

count how many times alternative $\mathbf{x}$ was chosen despite the fact that the decision maker preferred the value of attribute $k \in S_0$ from alternative $\mathbf{x}'$ over the corresponding attribute value of alternative $\mathbf{x}$. In that sense, Z_{1k} is a measure of *how badly* the attribute $k \in S_0$ performs in predicting choices ($\mathbf{x}$ over $\mathbf{x}'$).

[2] For notebook computers, these are: Processor (CPU), graphics card, weight, screen size, retail price, number of connectors, capacity of the hard disk drive and working memory (RAM). For TV-sets: screen resolution, retail price, contrast, brightness, screen size, number of connectors, power consumption, refresh rate

[3] For example, consider Notebooks are offered at the following prices: £999, £1089 and £1199. Then the number of different price levels is three and thus if the kth attribute is price, $n_k = 3$.

[4] In the notebooks example $\mathbf{x}_{\text{price}} = \{0, 1, 2\}$ for the prices $\{1199, 1089, 999\}$. Note the order, i.e. the least preferred price was assigned an x_{price} of 0.

As the rationale of a lexicographic strategy lies in identifying the attribute that — disregarding all other attributes — best predicts choices, minimizing Z_{1k} determines the most important attribute (which may vary across decision makers). Let k_1 denote an attribute for which Z_{1k} has the smallest values across all $k \in \{1, \dots, m\}$. That is,

$$Z_{1k_1} = min\{Z_{1k} | k \in S_0\}. \tag{18}$$

Select attribute k_1 as the first lexicographic (i.e. most predictive) attribute, arbitrarily breaking ties if necessary. Set the utility induced by the first attribute

$$u_1(\mathbf{x}) = \frac{x_{k_1}}{n_{k_1}}, \tag{19}$$

and

$$S_1 = S_0 \setminus \{k_1\}. \tag{20}$$

Thus, S_1 is the set of attributes remaining after k_1 is eliminated from S_0.

Recursion step. The procedure described above is then recursively applied to find the second, third (and so on) most predictive attribute out of the set of remaining attributes. For each alternative $\mathbf{x}$, compute

$$u_{tk}(\mathbf{x}) = u_{t-1}(\mathbf{x}) + \frac{x_k}{n_{k_1} n_{k_2} \cdots n_{k_{t-1}} n_{k_t}} \forall k \in S_{t-1} \tag{21}$$

where

$$S_{t-1} = \{1, \dots, m\} \setminus \{k_1, \dots, k_{t-1}\}. \tag{22}$$

is the set of attributes still not selected after step $t-1$, and the utility induced by attributes $1, 2, \dots, t-1$

$$u_{t-1}(\mathbf{x}) = \frac{x_{k_1}}{n_{k_1}} + \cdots + \frac{x_{k_{t-1}}}{n_{k_1} \cdots n_{k_{t-1}}}. \tag{23}$$

For each each $(\mathbf{x}, \mathbf{x}')$, for all $\forall k \in S_{t-1}$, let

$$d_{tk}(\mathbf{x}, \mathbf{x}') = \begin{cases} 1 & \text{if } u_{tk}(\mathbf{x}) < u_{tk}(\mathbf{x}') \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Compute

$$Z_{tk} = \sum_{(\mathbf{x}, \mathbf{x}') \in \Omega} d_{tk}(\mathbf{x}, \mathbf{x}'), \forall k \in S_{t-1}. \tag{25}$$

Let k_t denote an attribute for which Z_{tk} has the smallest value across all $k \in S_{t-1}$. That is,

$$Z_{tk_1} = min\{Z_{tk} | k \in S_{t-1}\}. \tag{26}$$

Select attribute k_t as the tth lexicographic attribute, arbitrarily breaking ties if necessary. Set the utility induces by attributes $1, 2, \dots, t$

$$u_t(\mathbf{x}) = \frac{x_{k_1}}{n_{k_1}} + \cdots + \frac{x_{k_t}}{n_{k_1} \cdots n_{k_t}}, \tag{27}$$

and

$$S_t = S_{t-1} \setminus \{k_t\}. \tag{28}$$

Termination step. Stop when there are no more alternatives left (in which case the attributes not considered are irrelevant to the problem), or if $t = m$ (in which case all alternatives have been considered).

A.2.2 Calculating utility

The above algorithm was used for identifying individual relative (ordinal) ranks of attributes from pairwise comparisons. Assuming that decision makers have a stable rank-preferences over attributes in sufficiently narrow choice problems, and I deliberately used the very same attributes in the Mouselab and in the conjoint decision problems to describe similar products, the attribute ranks obtained from the conjoint choice exercise could be used to make choice predictions for the product set available in the Mouselab environments. By applying a simple lexicographic utility function on the multi-attribute choice matrices used in part one of this experiment (Mouselab part), I could calculate utility values for all products in the choice sets.

Similar to the procedure used to infer attribute ranks, let n_k denote the number of different levels attribute k has in its attribute values. Arrange the levels of each attribute $k \in S_0$ in increasing preference order, and assign to them the sequence of integers $0, \ldots, n_k - 1$. Each level of attribute $k \in S_0$ is thus identified by a unique value $x_k \in \{0, \ldots, n_k - 1\}$.[5] Let $\mathbf{x} = (x_1, \ldots, x_m)$ denote a product profile in which attribute $k \in S_0$ has the level associated with x_k.

Using equation (27) I can calculate the utility derived from the 1^{st} to the tth attribute. Thus taking into account the utility derived from all m attributes I can use

$$u(\mathbf{x}) = \frac{x_{k_1}}{n_{k_1}} + \cdots + \frac{x_{k_m}}{n_{k_1} \cdots n_{k_m}}, \tag{29}$$

to calculate one utility value each, for all alternatives in the choice set.

[5]As the products used in the Mouselab tasks were real world products, their variety in attribute levels was on average larger than the artificially created products in the conjoint task. In order to apply the procedure explained above, I used percentile splits to transform every actual attribute level into a corresponding value x_k. For example, assume that the products of a Mouselab task had nine different prices and the products of the same category in the conjoint tasks (by design) had 3 different prices. In that case all price below the 33^{rd} percentile would be coded as 2, prices greater than or equal to the 33^{rd} and smaller than the 66^{th} percentile would be coded as 1 and prices greater than or equal to the 66^{th} percentile would be coded as 0. For variables with 2 or 4 levels in the conjoint task, the median or the first, second and third quartile were used to cluster attribute levels of the Mouselab task, respectively. This ensured $\forall k : n_{k_{\text{Mouselab}}} = n_{k_{\text{conjoint}}}$.

B Instructions and screenshots

All instructions were displayed on the computer screen. I therefore present the instructions in the form of screenshots to also provide a feeling how the screens looked like. As the screenshots are in the original experimental language (German) an English translation in plain text is provided after each screenshot.

B.1 Part 1: Mouselab part

Figures 1 – 5 represent screenshots of the first part of the experiment. The instructions to the Mouselab environment were followed by a training matrix (5 $\times$ 6) using digital cameras. After that participants were asked to select their favorite products in four 8 $\times$ 12 matrices. Subsequently, a questionnaire that contained open as well as closed questions had to be filled before the participants were paid for completing part one.

Figure 1: Screenshot of the training session instructions

Translation

Welcome to the practicing part
Please read the following information carefully! Shortly, you will be offered 6 digital cameras which are described by 5 attributes each. These attributes are:

- Price [Euro]
- Resolution (of the sensor) [Megapixel]
- Optical zoom [factor, x-times]
- Display size [Inches]
- Weight [Gram]

Your task is to select one of the cameras that are on average worth 200 Euros. Choose the one you would like to buy the most.
All relevant information regarding the cameras is presented on the next page using a matrix display whereby each camera is described by one column and each attribute is displayed as one line. Initially, all fields are covered such that you only get informed which information can be revealed by opening which cell. (e.g. *Price 3* or *Resolution 5*). All cells of the matrix can be opened by moving the mouse cursor over it. In case of opening a cell the actual content of that cell is displayed (e.g. *199 Euros* or *12 MP*). Hence, it is possible to look-up all information by simply moving over the mouse cursor. Leaving a cell with the mouse cursor again covers the cell. However, you can re-open all cells as many times as you want.
Once you have collected enough information you can choose one of the 6 cameras by clicking on the relevant button at the bottom of the screen. Should you want to change your choice, please click on the cameras that you really want to choose. Only the last selection counts. After that please click the button *Continue* which you can find on the left bottom of the screen! The purpose of the product matrix displayed on the next page is to make you familiar with the handling of the experiment. Please click *Continue with the practicing matrix* to start the training!

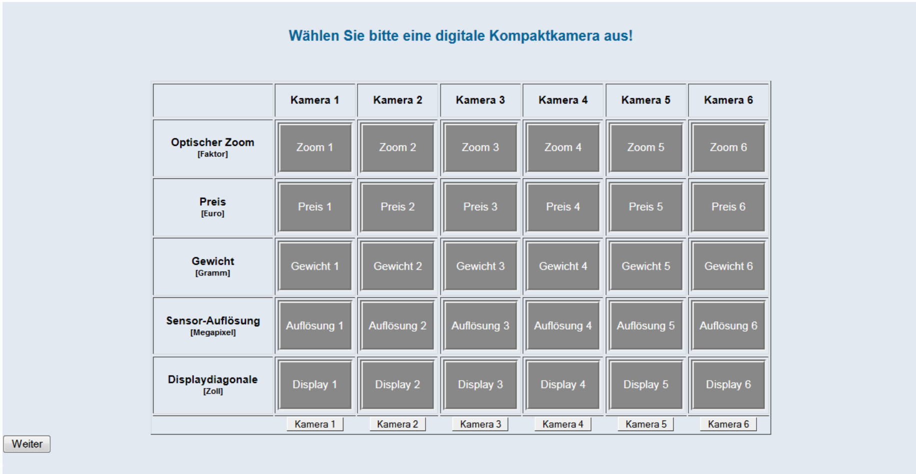

Figure 2: Screenshot of the training session: a 5×6 matrix on digital cameras; currently, all cells are covered

Translation

Please choose one digital camera!

- Optical zoom
- Price
- Weight
- Resolution of the sensor
- Size of the display

Auswahl eines Fernsehgerätes (unter 700 Euro)

Bitte lesen Sie folgende Informationen sorgfältig durch!

In Kürze werden Ihnen insgesamt 12 LCD-Fernsehgeräte angeboten, die sich jeweils durch 8 Merkmale auszeichnen. Diese Merkmale sind:

⇒ Preis [Euro]
⇒ Kontrast [Verhältnis ? : 1]
⇒ Helligkeit [Candela/m2]
⇒ Bildfrequenz [Hertz]
⇒ Bildschirmdiagonale [cm]
⇒ Bildschirmauflösung [Bildpunkte]
⇒ Stromverbrauch [Watt]
⇒ sowie die verfügbaren Anschlussmöglichkeiten wie z.B. USB, HDMI etc. (Anschlüsse)

Ihre Aufgabe ist, jenes der zwölf im Durchschnitt 580 Euro teuren Fernsehgeräte auszuwählen, welches Sie am liebsten kaufen würden.
Die Informationssuche und Auswahl funktioniert wie im Übungsbeispiel der Digitalkameras.

Aber ACHTUNG:
Im Unterschied zur Trainingsphase, kann man hier einen Gutschein über den halben Kaufpreis eines tatsächlich gewählten Produktes gewinnen. Unter allen Teilnehmer/inne/n dieses Experiments wird ein Gutschein über 50% des Kaufpreises, eines während des Experiments gewählten Fernsehgerätes verlost. Wichtig dabei ist jedoch, dass der/die Gewinner/in nur einen 50% Gutschein für jenes Produkt erhält, das er/sie zuvor im Rahmen dieses Experiments gewählt hat. Der Gutschein kann dann z.B. ausschließlich zum Kauf eines LCD-Fernsehers verwendet werden, dessen Merkmale mit jenem während des Experiments gewählten übereinstimmen.
Folglich sollten Sie gewissenhaft bei der Auswahl vorgehen und ehrlich das persönlich wünschenswerteste Gerät ausgewählt werden, da sie nur genau jenes gewinnen können.

Bitte klicken Sie auf *Weiter zur Produktmatrix* um mit dem Experiment zu beginnen!

Weiter zur Produktmatrix

Figure 3: Screenshot of the instructions before the 8×12 matrix on relatively cheap TV-sets

Translation

Choice of a TV-set (below 700 Euros)
Please read the following information carefully!
Shortly you will be offered 12 LCD TV-sets which are described by 8 attribute each. These attributes are:

- Price [Euro]
- Contrast [Ratio ? : 1]
- Brightness [Candela/m^2]
- Frame rate [Hertz]
- Screen diagonal [cm]
- Screen resolution [Pixels]
- Energy consumption [Watt]
- and the available connectors such as USB, HDMI etc. (connectors)

Your task is to select one of the TV-sets that are on average worth 580 Euros. Choose the one you would like to buy the most.
The search for information works exactly the same way as for the digital cameras in the practicing example.
CAUTION: In contrast to the practicing phase you can win a voucher worth half of the value of the actually chosen product. One voucher will be drawn amongst all participants of this experiment. It is important to note that the winner will get a voucher worth 50% of the purchasing price of exactly the product he or she chose during the experiment. The voucher can only be used for buying a product that has exactly the same attribute values as the model chosen during the experiment.
Hence, you should deliberate carefully and choose honestly, because you can win a voucher that is valid only for exactly the chosen product.
Please click *Continue to the product matrix* to start the experiment!

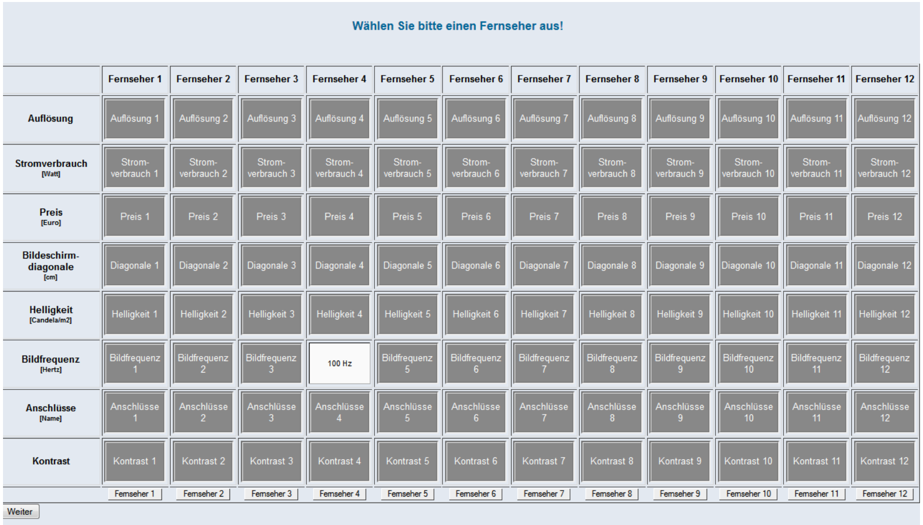

Figure 4: Screenshot of the 8 × 12 matrix on relatively cheap TV-sets; the mouse cursor currently reveals the frame rate of TV-set 4

Translation

Please choose a TV-set!

- Resolution
- Power consumption
- Price
- Screen diagonal
- Brightness
- Frame rate
- Connectors
- Contrast

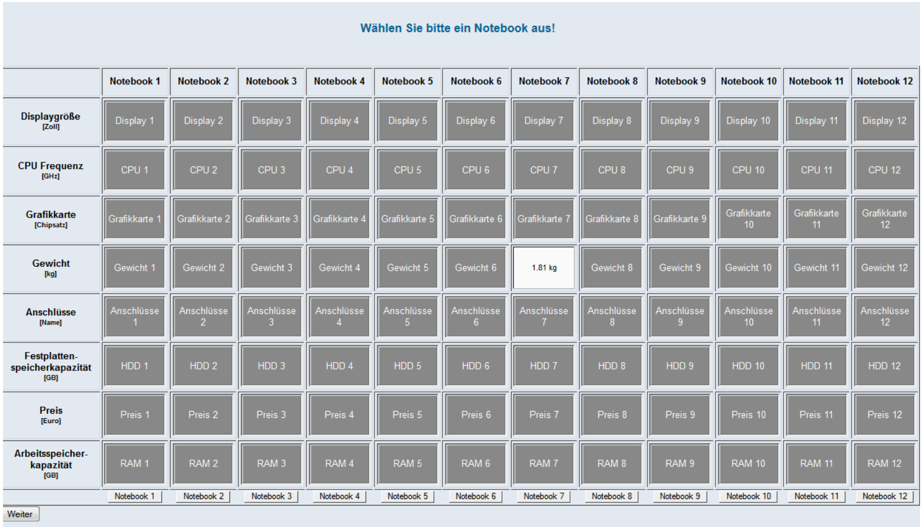

Figure 5: Screenshot of the 8×12 matrix on relatively cheap notebooks; the mouse cursor currently reveals the weight of notebook 7

Translation

Please choose a Notebook!

- Display size
- CPU frequency
- Graphics
- Weight
- Connectors
- Harddisk Capacity
- Price
- Memory Capacity

B.2 Part 2: Binary choice tasks

Figures 6 and 7 present screenshots of the second part of the experiment. After displaying the instructions for part 2 participants had to make twenty-four decisions in a binary choice task. Subsequently, participants engaged in an incentivized ambiguity aversion task and finally were paid.

Figure 6: Screenshot of the instructions of the binary choice section in part two

Translation

Welcome!
Please read the following information carefully!
Shortly, you will be offered 2 notebooks which are described by 8 attributes each. These attributes are:

- Price [Euro]
- CPU-frequency (CPU) [Count of processor cores x GHz]
- Graphics [Chipset and memory capacity]
- Screen size [Inches]

- Weight [kg]
- Memory capacity (RAM) [GB]
- Harddisk capacity (HDD) [GB]
- and the available connectors such as USB, HDMI etc. (connectors)

Your task is to decide which of the two notebooks you prefer over the other. Please deliberate carefully before choosing one of the two notebooks.
All relevant information about both notebooks are presented in tables on the following pages where no attribute names but just the relevant attribute values (including units) are displayed. For example a table entry of *2.5 GHz*, although it is not described explicitly, indicates that this notebook has a processor with a CPU frequency of 2.5 GHz. The same principal applies to all other attributes. In contrast to the first part of this experiment all information is visible from the very start. Thus, all decision-relevant information can be used immediately.
Be aware that each product, i.e. each combination of attribute values, is only displayed ONCE. Thus, all products that will be displayed soon are different with respect to at least one and at most with respect to all their characteristics. Please do *under no circumstances* choose one of the two options *arbitrarily*!
As soon as you have identified preferring one notebook over the other, please click the radio button beneath the column your preferred product is described in. Should you change your mind and wanted to choose the other product simply click the other radio button. Only the final selection counts. Subsequently, please click *Continue* to continue with the next product comparison.
Please click *Continue with pairwise comparisons* to start with the experiment!


Paarweises Vergleichen von Notebooks

Notebook	Notebook
2 GB	2 GB
Intel GMA 3100 (shared)	Nvidia GeForce GTX 4585M (1GB)
1299 EUR	1189 EUR
640 GB	640 GB
15.6 Zoll	17.3 Zoll
2.1 kg	2.1 kg
4 x USB, LAN, WLAN, HDMI, Bluetooth	2 x USB, LAN, WLAN
2 GHz	2 GHz

Figure 7: Screenshot of one of twenty binary choices that had to be made

References

Rajeev Kohli and Kamel Jedidi. Representation and Inference of Lexicographic Preference Models and Their Variants. *Marketing Science*, 26(3):380–399, May 2007. ISSN 0732-2399. doi: 10.1287/mksc.1060.0241.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In P Arembka, editor, *Frontiers on Econometrics*, pages 105–421. Academic Press, New York, 1974.