Predicting Task Performance - How to Quantify Metacognition Using Drawn Confidence Distributions

Applied Cognitive Science
Yannik P. Frisch, Maximilian A. Gehrke
March 10, 2020

TECHNISCHE UNIVERSITÄT DARMSTADT

Centre for Cognitive Science

Abstract

Several studies suggest that humans are rather poor when judging their own performance after they have fulfilled a task. We present a new method to use drawn probability density functions to measure the subjects' confidence of their performance on several tasks of a questionnaire that we designed especially for that purpose. Subsequently we rate their answers using a scoring function that is unknown to the subjects, before we evaluate their confidence by statistical measurements of the discrepancy between their expected and their actual performance on the tasks. After evaluating these, we come to the conclusion that [... TODO] and give suggestions for further experiments on this topic, using a wider variety of tasks and a higher number of subjects.

1 Introduction (Yannik P. Frisch)

The question of how to predict peoples performance on a task has been of great interest in the psychology literature. To be able to make qualified statements about this ability, one has to define it first. This can be achieved by the framework of *metacognition* including the frequently used terms *metacognitive sensitivity*, *metacognitive bias* and *metacognitive efficiency*, as presented by [5], that will be explained during this first section.
Metacognitive sensitivity is used to express how good a subject is at differing between his or her own correct and incorrect answers. For example, imagine a classical experiment from *signal detection theory (SDT)* where the subject's task is to rate a stimulus to come from a class A or a distinct class B. Poor metacognitive sensitivity would result in a bad distinction between the possible origins, even though the classes are easy to separate for an average observer. A useful initial approach to quantify *metacognitive sensitivity* is the $2x2$ confidence-accuracy table, labeled *type-2 SDT table* by [5], that is the equivalent of the usual *type-1 SDT table* [x], shown in the upper part from figure 1, applied to the *metacognition* framework. The resulting table is displayed in the lower part of 1. Typical SDT measurements of the association between the rows and the columns of the table in the type-1 case are the ϕ-*correlation*, defined e.g. in [4], and the *Goodman-Kruskall gamma coefficient* G from [7]. Imagine a subject reporting a subjective confidence for multiple trials of the former example stored in a vector, e.g (A, B, B, A), and the vector of the actual correct classifications (A, B, A, B). If we encode $A = 1$ and $B = 0$, the ϕ-correlation is the *Pearson r-correlation* between both vectors. The advantage of using G instead is the abundance of any distributional assumptions on the data and the possibility to easily extend the measurement to a confidence rating scale (e.g. from 0 to 100), rather than a binary encoding (e.g. 0/1). Nevertheless it is well known that both measurements are affected by bias [5]. [16] and [14] could show that this also holds for the type-2 application of these measurements.

Table 3.1 Possible outcomes for the type 1 task

Stimulus	Response	
	"S1"	"S2"
S1	Correct rejection (CR)	False alarm (FA)
S2	Miss	Hit

Table 3.2 Possible outcomes for the type 2 task

Accuracy	Confidence	
	Low	High
Incorrect	Type 2 correct rejection	Type 2 false alarm
Correct	Type 2 miss	Type 2 hit

Figure 1: The type-1 and type-2 2x2 SDT-tables [TODO: REMOVE TABLE 3.1 / TABLE 3.2 CAPTIONS FROM IMAGES][TODO: MAKE OWN PLOT!].

A standard way to remove the influence of the bias in classic SDT is using d' as in [8] which will be constant given different biases and also has several approaches to metacognitive sensitivity, e.g. [12] defined type-2 d' as

$$d' = z(H2) - z(FA2)$$

A visualization of this measurement is shown on the left side of figure 2.

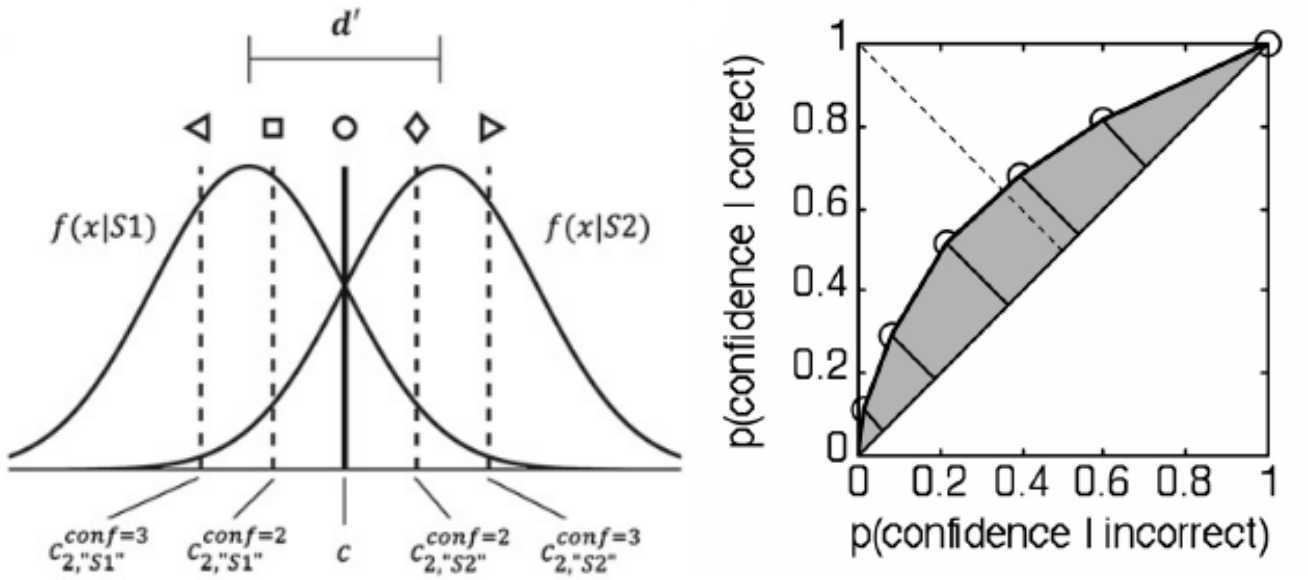


Figure 2: The left side of this figure shows an example for the d' measurement, calculated as the difference between the (inverse of) the hit rate (HR) and the false alarm rate (FA). The right side displays an example for a type-2 ROC curve, displaying the relationship between FA2 and HR2. The area under the ROC, called AUROC gives a measurement of a subject's metacognitive sensitivity.

Despite this advantage, d' can not easily applied to the type-2 data, because it includes assumptions of gaussian distributions with equal variance, which is very unlikely to be the case in type-2 data, as shown by [6]. Experiments of these [1] showed that it is indeed affected by changes in the metacognitive bias.

One way to remove this issue is the use of non-parametric analysis, that does not make these assumptions, e.g. the *Receiver operating characteristic (ROC)* analysis [x] in classic SDT. This approach can be applied to type-2 data analogue to [3] and an example is given in the right part of figure 2

A further complication for using the above methods to measure metacognitive sensitivity is the fact that all these measures are affected by the task performance [6]. This can be addressed by explicitly modeling the connection between a subject's performance and metacognition in a model-based approach. The *meta-*d' measure, defined by [13], makes use of the fact that given gaussian variance assumptions at type-1 level, the shapes of the type-2 distributions are known even if they are not themselves gaussian. Therefore the optimal type-2 performance is constrained by one's type-1 performance. E.g. given a particular type-1 variance structure and bias, the form of the type-2 ROC can be completely determined. So, given a subject's actual type-2 performance, one could reconstruct the underlying type-1 sensitivity, labeled meta-d', that is robust to changes in the bias and recovers simulated changes in metacognitive sensitivity, e.g. by altering task difficulties. For a metacognitive ideal observer, meta-d' should be equal to d'. To measure this quantity, [5] defined *metacognitive efficiency* as the ratio of meta-d'/d', or by the more stable variants meta-$d' - d'$ or logmeta-d'/d'. However, this measurement is unable to discriminate between different causes of a change in metacognitive efficiency. For example, trial-to-trial variability in the placement of confidence criteria results in decreasing efficiency as well as additional noise in the evidence used to make the confidence rating. A similar bias-free approach to model metacognitive accuracy is the *Stochastic Detection and Retrieval Model (SDRM)* from [10] which we do not want to cover here.

A somewhat different approach uses so-called *one-shot* discrepancy measures to quantify metacognition, e.g. in [11]. A general confidence rating (Asked before of after the trials) is compared to the actual performance on a variety of tasks, but it should be clear from the above that using a single rating of performance will not result in a good distinction between the bias and sensitivity, nor will it enable to measure the efficiency. In contrast, collecting trial-by-trial measures of performance and metacognitive judgments allows to get a picture of an individuals bias, sensitivity and efficiency.

To get a different view-point on the domain, one could formalize metacognitive confidence as the ability to make good probability judgments directed towards the accuracy of one's own actions.

1.1 Formalizing metacognition as probability judgments

In this framework *metacognition* gets a normative interpretation as the accuracy of a probability judgment about one's own performance and one advantage is the possibility to elicit a meaningful measure of the bias. A lot of literature is a available on how to measure this accuracy, but in the following we want to focus on one of them, called the *Brier Score*. To define this score, [5] first defined the *Probability Score (PS)*, analogue to [9], as the squared difference between a probability rating f for an event, and it's actual

occurrence c (0 or 1 for binary events):

$$PS = (f - c)^2$$

Based on this, the *Brier Score (BS)* from [2] can then be defined as the mean value of the PS averaged across all estimates:

$$BS = \frac{1}{N}\sum_i (f_i - c_i)^2$$

This score is an equivalent of the ϕ or G measurements explained above and can be decomposed, as shown by [15], in the following way:

$$BS = O + C - R$$

where O is the *Outcome Index*, reflecting the variance of the outcome event c:

$$O = \bar{c}(1 - \bar{c})$$

The *Calibration* expresses the goodness of fit between the probability assessments and the corresponding proportion of correct responses. It quantifies the discrepancy between the mean performance level in a category (e.g. 60%) and its associated rating (e.g. 80%):

$$C = \frac{1}{N}\sum_{j=1}^{N} N_j(f_j - \bar{c_j})$$

Last, the *Resolution* R encodes the variance of the probability assessments, measuring the extent to which correct and incorrect answers are assigned to different probability categories:

$$R = \frac{1}{N}\sum_{j=1}^{N} N_j(\bar{c_j} - \bar{c})$$

As mentioned earlier, this framework allows us to extract more meaningful quantities about a subject's metacognitive sensitivity. One such quantity is the calibration of a subject, as shown in figure 3, that directly shows over- or underconfident tendencies of a person.

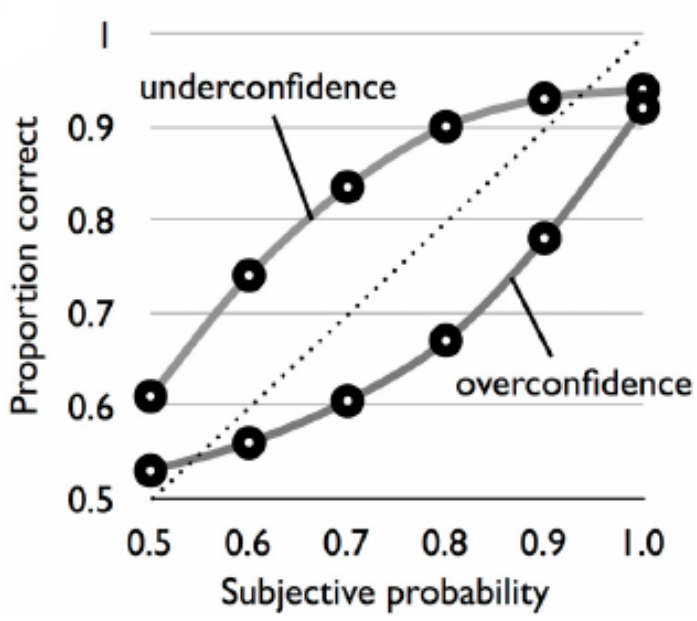


Figure 3: TODO: CAPTION

multiple possible outcome classes, the brier score can be calculated as shown by [2] by summing over all classes j:

$$BS = \frac{1}{N}\sum_j \sum_i (f_{ij} - c_{ij})^2$$

Note that this formulation does have a range of scores from 0.0 to 2.0, where the analogue values of the one-class formulation are twice as high, e.g. a brier score of 0.5 in the one-class case would correspond to a score of 1.0 in the multi-class formulation.
The next chapter gives explains our methods to actually capture a subjects confidence and calculate a measure of the brier score out of it. We present an experiment that we evaluated on several subjects for different tasks, before we display the results of our measurements in chapter 3 and finally come to a conclusion on our attempt to measure metacognition in chapter 4.

2 Method (Maximilian A. Gehrke)

2.1 Designing the Questionnaire

We started out by designing a questionnaire in LaTeXand using the corporate design of our university. We separated each task clearly from the one another and wrote the instructions in the headline. The body of each tasks consists of task related information on the left, space for the answer in the middle and an empty coordinate system on the right. It was important to us to keep this structure to increase reliability across the tasks.

We decided to use sorting tasks, because of their high objectivity (see section 4 for a discussion on task types). The first seven tasks were closed-form sorting tasks. Directly after executing each task, the subjects were asked to fill out the coordinate system with a probability density function over their performance. We decided to ask the subjects to sort five terms in a predefined order. We decided to use two easy items, three medium and two hard items. We randomized the location of the items as well as the order of the answer possibilities. An example can be seen in figure 4.

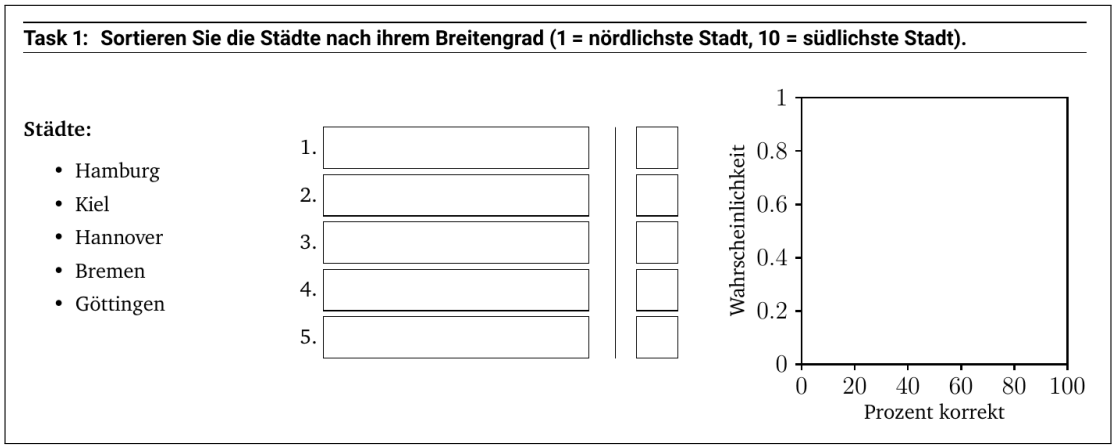


Figure 4: Example closed-form sorting task.

After the seven tasks, we asked the subjects to estimate their performance over all the previous tasks, by drawing another probability density function. With this we want to learn how well humans can average their performance on several task.

The eighth task was an open end sorting tasks, which we decided to incorporate out of curiosity how the self-assessment would change in comparison to closed form sorting tasks. The task can be seen in figure ??.

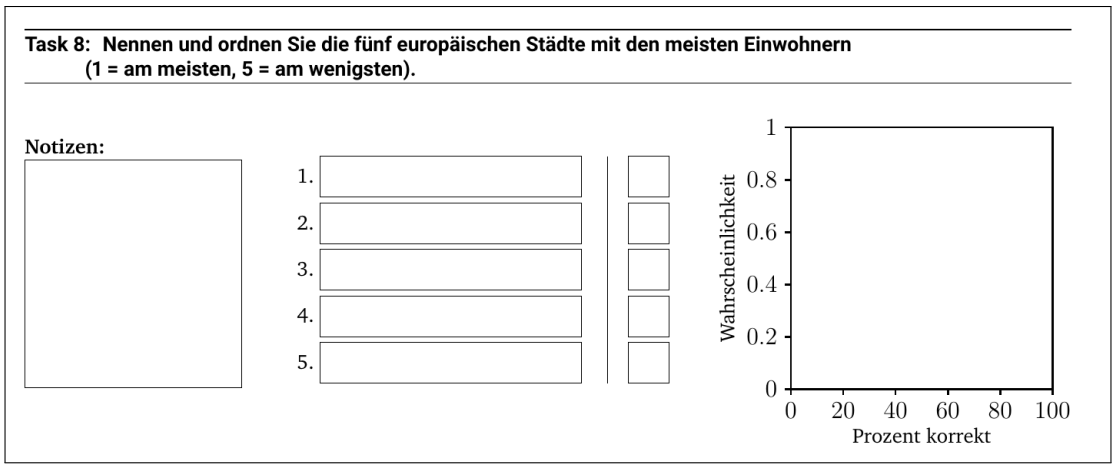
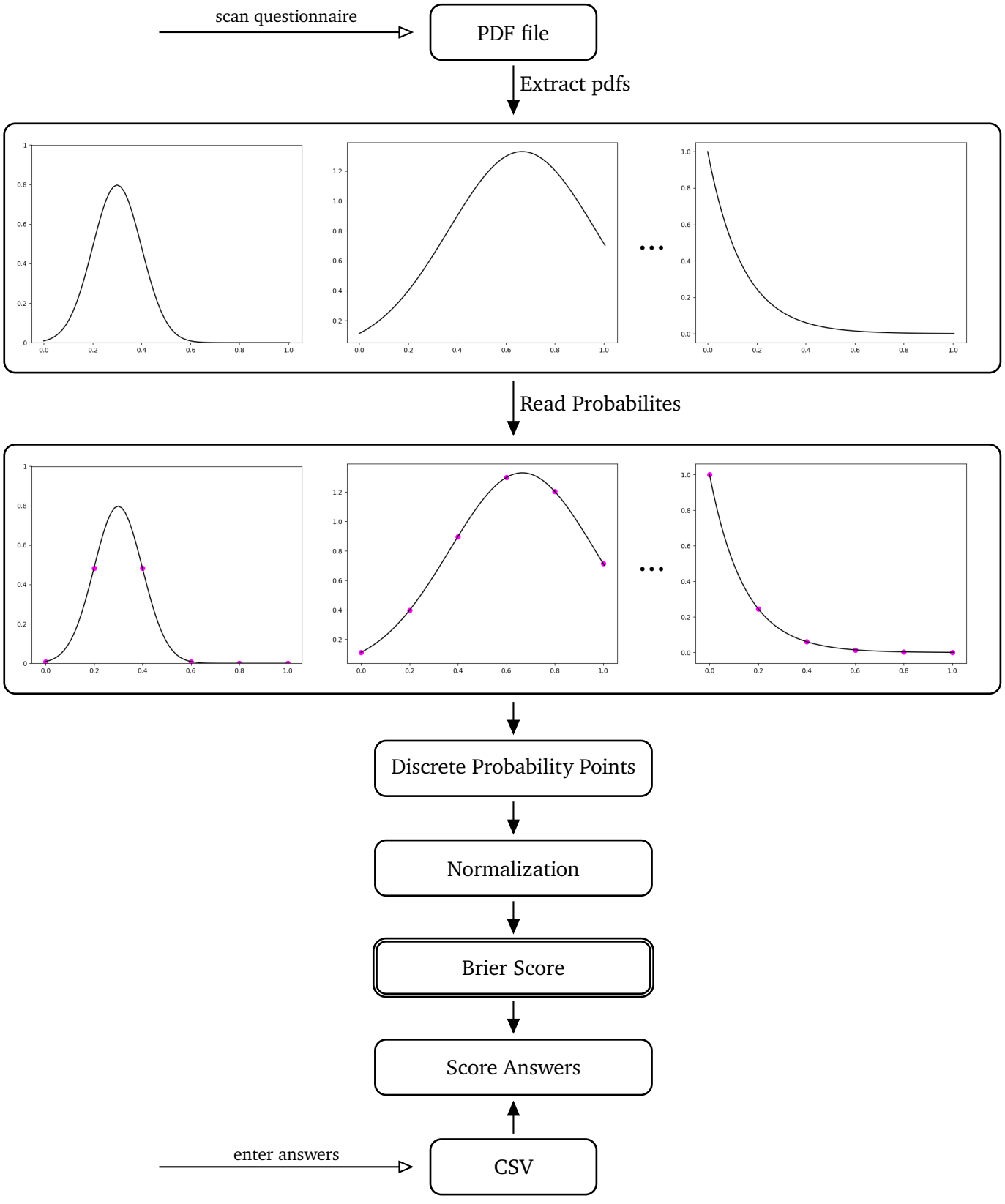


Figure 5: Example open end sorting task.

2.2 Processing Pipeline



The pipeline that we used to process the questionnaire can be seen in the figure above. We start out by digitizing the image. This is important so that we are able to apply computer vision to extract the probability density functions. For convenience we provided two ways: either scan the questionnaire as a PDF file or photograph each page of the questionnaire and upload the pages as JPEG's. In the first case, we use the python package "pdf2image" to convert each PDF page into a single JPEG page, which is not necessary in the second case, cause the JPEG's are already available.

After the digitization of the questionnaire, the probability density functions are detected on the questionnaire and extracted as single images. From these images we build a digital representation of each probability density functions and read discrete probabilities at the desired locations. Afterwards we normalize the probabilities in order for them to sum to one. This way, the subjects do not need to care about drawing a normalized probability function and can rather concentrate on which area they assign a higher percentage and which areas they assign a lower percentage.

For a meaningful comparison, we need need to insert the answers of the subjects. We do this by filling a CSV file for each subject by hand. The answers from the CSV file are imported into our program and subsequently scored. An automatic scoring improves the objectivity and avoids errors. Finally, we use the points that each person achieved for each tasks and the normalized probabilities from the probability density functions to calculate the Brier score, which quantifies uncertainties.

The remaining segments of this sections examine the processing pipeline in more detail.

2.3 Extracting Probability Density Functions

To extract the probability density functions from a scanned PDF into an image, we used computer vision to detect the probability density functions on the JPEG and simply cut out the detected regions from the JPEG. The cutout process is fairly easy. In python, images are stored as an array of numbers. As soon as we get the area of the pdf as pixels, we can simply enter these pixel indices in the array and extract that part of the image. Then we save it using OpenCV a free computer vision library in python.

The hard part is to identify the probability density functions in the image. However, we designed our questionnaire in a way that reduces the detection of the pdf to the detection of a big square. If we can reliably detect the coordinate system, which the subjects use to draw their pdfs in, we can extract the pdf if we only look at the pixels which lie inside this square.

To detect squares, we need to detect vertical and horizontal lines first. We did exactly that and looked at all contours that could be build with horizontal and vertical lines. This will output all lines on their own, but also all triangles, rectangles and squares. Everything that forms a contour. Now we sorted the contours. It is important to sort the contours (or later pdfs), regarding their position on the page. To the computer all pdfs look the same, so we have to make sure that we assign the correct pdf to the correct task. Because all tasks are ordered in ascending order, we can sort the pdfs from north to south.

Next, we had to find the correct square from the contours. We did this by iterating over the contours and testing each contour regarding some constraints.

1. The horizontal and vertical length of the contour had to be approximately the same length. We allowed for exactly 8% variation. So one side could be up to 8% longer than the other side. This factor is necessary, because of the difficulty to scan a paper perfectly aligned.

2. The coordinate system has a height of 4.5cm. A Din A4 paper is exactly 29.7cm high. This means that each pdf takes about 0.15% of the height of the image. Allowing for some deviation, the height of the contour had to be between 10 and 20% of the height of the JPEG.

3. The coordinate system has a width of 4.5cm. A Din A4 paper is exactly 21cm wide. This means that each pdf takes about 0.21% of the width of the image. Allowing for some deviation, the width of the contour had to be between 15 and 25% of the width of the JPEG.

4. Last, we excluded all contours which were lying on the left half of the page.

Constraint 1 makes sure that we find squares. Constraint 2 and 3 make sure we find the squares with the correct size. Constraint 4 makes sure that we do not take any squares from the left side of the page. We designed the questionnaire in a way that all pdfs are located at the right half of the page.

Because we calculate the extraction of the pdfs with percentages relative to the size of a Din A4 paper, it is independent of the DPI and resolution of the scanner or photo camera with which the questionnaire is copied.

2.4 Extracting Probabilites from probability density functions

2.5 Scoring the answers

To score the answers, we applied the Euclidean norm (L2-norm) to the distance between the actual position of the items and the chosen position of the items. The correct order of a sorting task is $[A, B, C, D, E]$, but the subject wrote $[B, D, E, A, C]$. Translated into numbers the correct ordering always translates to $[1, 2, 3, 4, 5]$ and in this case the answer of the subject translates to $[2, 4, 5, 1, 3]$.

Now we take the L2-norm, which is displayed for our toy example in equation (1). The better the placement of the answers, the lower is the L2-norm. A result of $L2 = 0$ represents a perfect fit.

$$L2 = \sqrt{(1-2)^2 + (2-4)^2 + (3-5)^2 + (4-1)^2 + (5-3)^2} \approx 4.69 \tag{1}$$

We then calculated the worst L2-norm possible. That is the L2 norm of the difference between the actual ordering $[1, 2, 3, 4, 5]$ an the worst ordering possible $[5, 4, 3, 2, 1]$. This resulted in the worst possible L2-norm of L2-max-norm ≈ 6.32. We then split the interval of $[0,$ L2-max-norm$]$ into an array of a fixed amount of equidistant numbers (in our case 6). We chose six, because we assigned a score of 0 to 5 points for each task. This led to the array of $[0, 1.27, 2.53, 3.79, 5.06, 6.32]$. The discrete rating for a task was then the 'id' of the closest value of that array compared to the L2-norm of the answer. Therefore, if a subject would get a L2-norm of $L2 = 4.69$ (equation 1), the discrete score for the task would be 4 points.

2.6 Calculating the Brier Score

To calculate the accuracy of the subjects predictions, we used the Brier score. The Brier score is a function that has been designed to do exactly that. It takes the mean squared differences between the actual outcome $o_i \in 0, 1$ and the probability $p_i \in [0, 1]$ assigned to each outcome:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(p_i - o_t)^2 \tag{2}$$

An outcome of $o_i = 1$ represents the occurrence of an outcome and analogous constitutes an outcome of $o_i = 0$ that an outcome did not occur. A probability of $p_i = 1$ describes a 100% certainty that a specific outcome will occur and a probability of $p_i = 0$ describes the certainty that an outcome will not occur. A Brier score of $BS = 0$ specifies a perfect correlation between actual outcomes and predictions. A Brier score of $BS = 1$ characterizes the worst possible accuracy of predictions, that is depicting a probability of $p_i = 1$ for an outcome that did not occur and assigning all other outcomes a probability of $p_i = 0$, including the outcome that actually occurred.

We now show how we applied the Brier score to our experiment. The participant could get 0 to 5 points for each task. Figure 6 shows an example task result.

	Possible task points					
	0	1	2	3	4	5
o_i	0	0	1	0	0	0
p_i	0.2	0.6	0.2	0	0	0

Figure 6: Example task result. The subjects certainty is encoded in the probabilities p_i for each possible task score. The actual outcome is encoded with $o_i \in 0, 1$, where 1 displays an occurring event and 0 a non-occurring event.

The actual score that the participant got for the task is encoded with $o_i = 1$ and the other possible scores that did not occur are represented by $o_i = 0$. The participant however was underconfident and gave a possible task score of 0 points a probability of 0.2, a possible task score of 1 point a probability of 0.6 and the possible task score of 2 points (which actually occurred) a probability of 0.2. If we now apply the Brier score to our data, we get a Brier score of $BS \approx 0.17$ which can be seen in equation 3.

$$BS = \frac{(0.2-0)^2 + (0.6-0)^2 + (0.2-1)^2 + (0-0)^2 + (0-0)^2}{6} \approx 0.17 \tag{3}$$

3 Results (Yannik P. Frisch)

In the following we present several plots showing the results of our experiment evaluations, starting with some information about our demographic data, before displaying general averaged results, followed by a demonstration of individual evaluations of the task performance and bier scores.

3.1 Demographic Data

Our questionnaire shown in ?? has been evaluated on 14 subjects. Their average age was 31.5 years within a range of our youngest 22 year old subject and our oldest subject of 62 years. The group consisted of 6 females and 8 males, all with german nationality and german as their mother language. Their professions included one computer scientist, one scientific coworker, one social worker, one curative educator, one insurance agent and nine students. The subjects of the later included *Psychology in IT* 5 times and *Computer Science* two times. One student's subject was *Social Works*, one was studying *Psychology* and one student came from the field of *Cognitive Science*.

3.2 Averaged Task Performance and Brier Score

The average task performance of our subjects is shown in 7, with an absolute mean of 2.938 points, averaged across all subjects and tasks, and a standard deviation of 1.391.

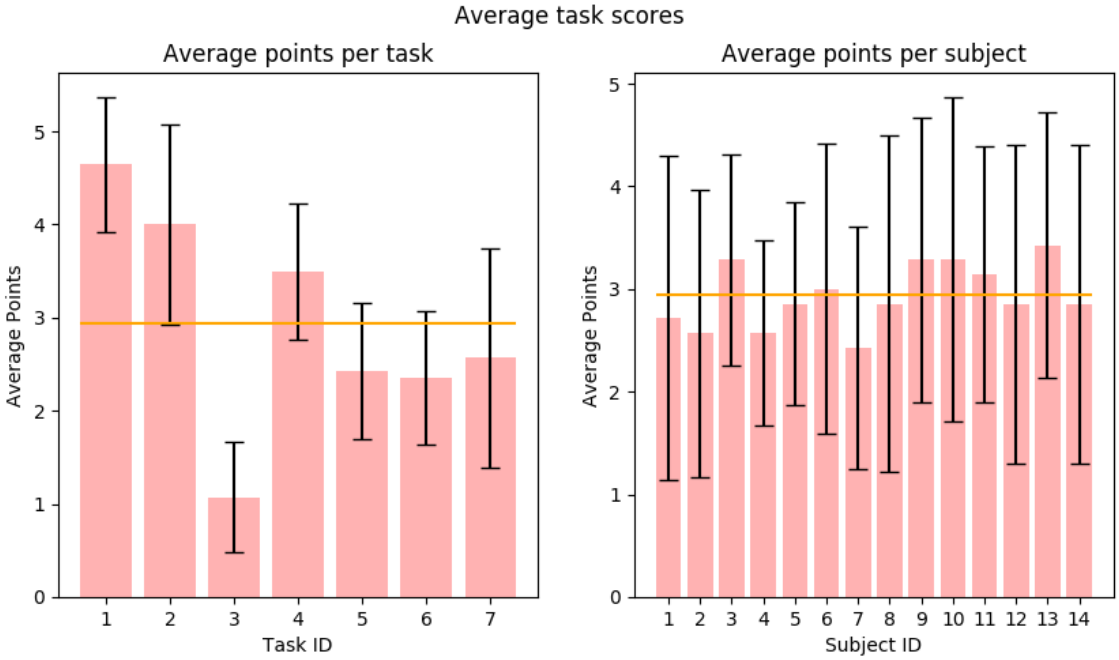


Figure 7: Means and standard deviations of the achieved task scores, averaged over subjects in the left and over tasks in the right plot. The orange line displays the overall mean of [TODO], averaged over both variables.]

The subjects abilities to judge about their own performance, encoded by the average brier score, is displayed in figure 8 with an overall average of 0.864 and a standard deviation of 0.264.

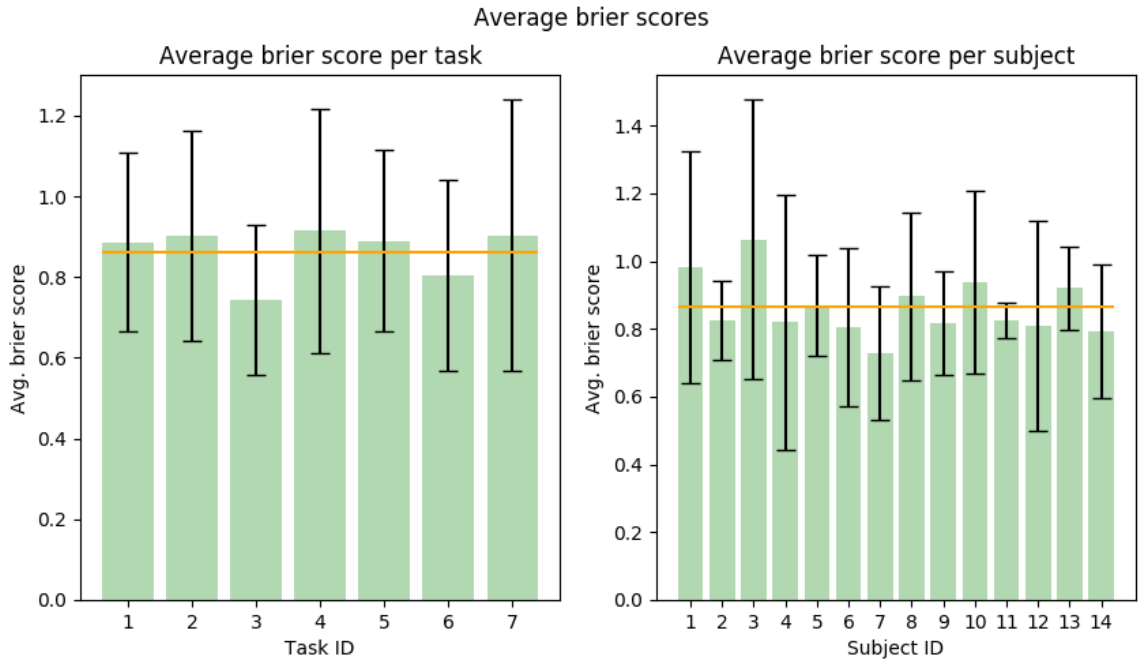


Figure 8: Means and standard deviations of the calculated brier scores, averaged over subjects in the left and over tasks in the right plot. The orange line displays the overall mean of 0.14, averaged over both variables.

Please note that we excluded the 8th task from our questionnaire in figure ??. For interpretation of these values, please refer to our discussion of the experiment in section 4.

3.3 Individual Evaluations

We did individual evaluations of our subjects' task performance, brier scores and confidence ratings. While all figures can be found in appendix ??, figure 9 displays an example for one subject.

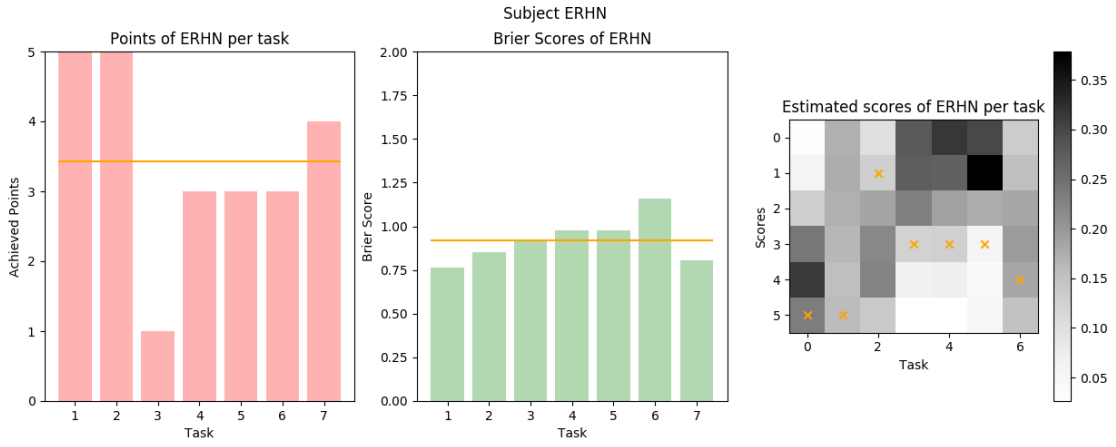


Figure 9: Example graphic of our individual evaluations for subject ERHN. We again plotted the achieved discrete scores per task on the left and the calculated brier-scores per task in the middle. The orange line displays the mean values for both bar plots, averaged over all tasks. The subject's estimated discrete probabilities of achieving a score are shown in the right graphic, where the orange x marks the actual achieved score in the task. [TODO: FIX X-RANGE OF RIGHT PLOT!]. One could already conclude a tendency of this subject to give underconfident ratings.

3.4 Confidence Plots

To get more compact insights about our subjects' confidence, we also created scatter plots of the type (Expected Rating) vs (Actual Rating), see figure 10 for an example. A subject's expected rating for a task is calculated out of one column of his or her probability matrix, see the above subsection for an example. The individual plots for all our subjects can be found in appendix ??.

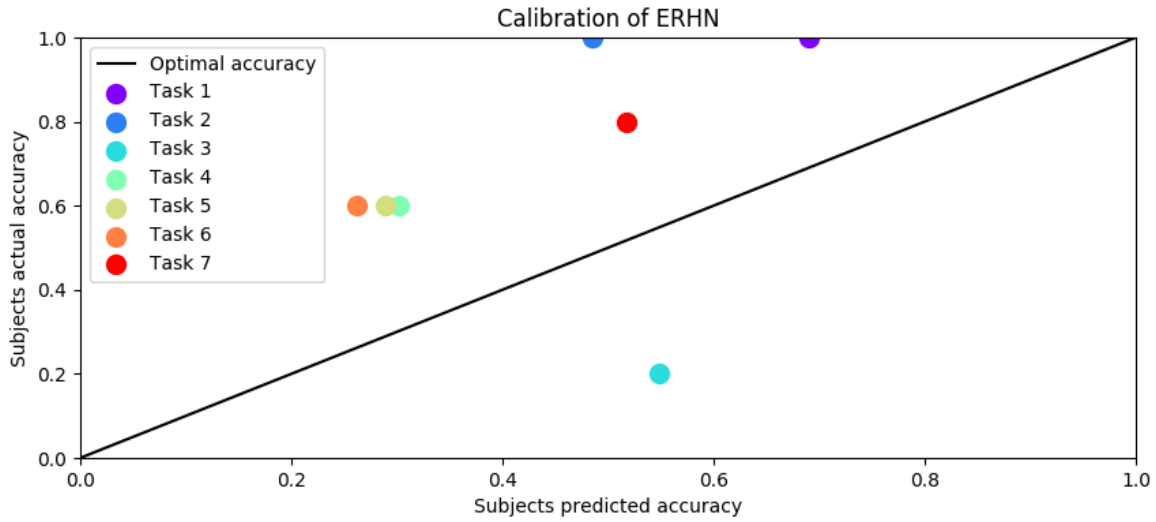


Figure 10: Example graphic of our individual evaluations on the confidence of subject ERHN. We created a scatter plot with the subject's expected rating for each task on the x-axis and his or her actual achieved rating on the y-axis. Optimal confidence is indicated by the solid black line, where expectation meets the actual outcome. Scatter plots of tasks above this line indicate underconfidence for a task, while scatter plots below the line are a sign of overconfidence.

These average confidence per task is also displayed in figure 11.

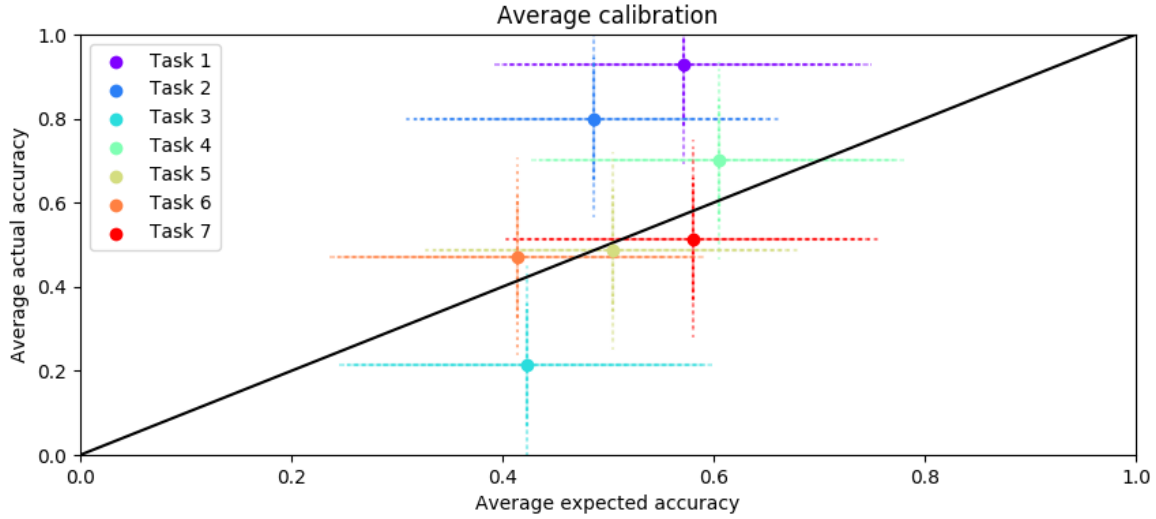


Figure 11: Scatter plot of average (Expected Rating) vs (Actual Rating) relationship with error bars in both dimensions. Note that the errors are not actual lines around the scores, but the ratings are very likely to be inside the ellipsoids spanned by these standard deviation error bars.

Because we might have found a coincidence between the performance on a task and the confidence of a subject about his rating, we try to get further insights into this by our next evaluation, presented in the next subsection.

3.5 Task performance vs Brier Score

To further examine a possible relationship as stated above, we also show the average relationship of (Task Performance) vs (Brier Score) in figure 12.

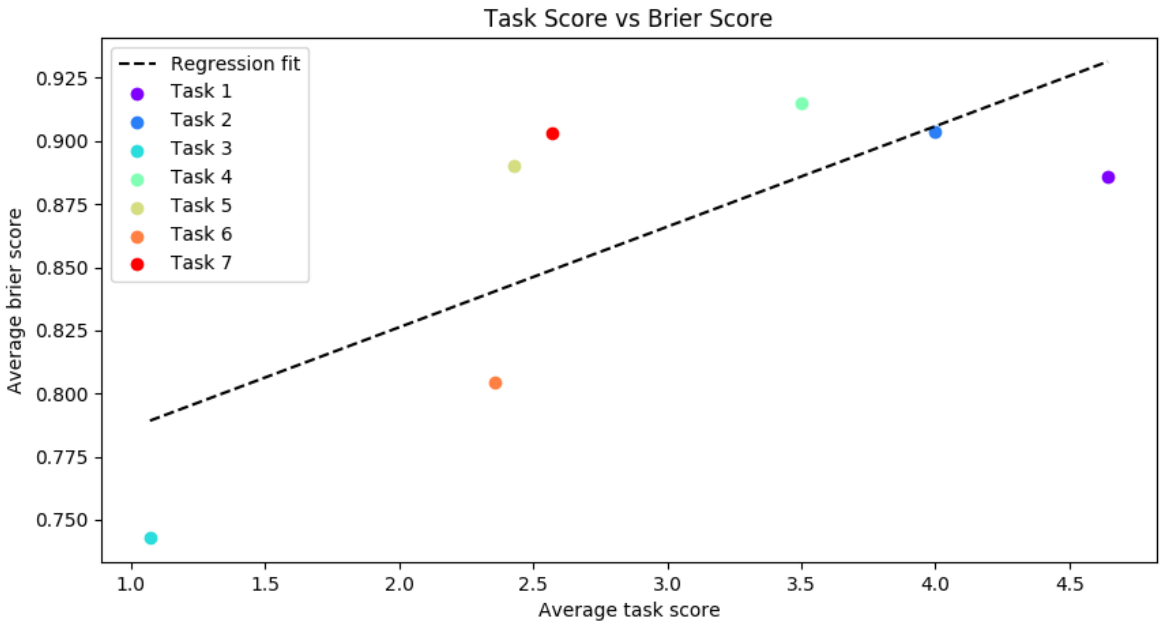


Figure 12: Scatter plot per task by it's mean task score (x-axis) and it's mean brier score (y-axis), averaged over all subjects. Black dashed line shows a linear regression fit on the data.

We also fit a linear regression function on the data, resulting in a seemingly positive correlation between the actual achieved task score and the goodness of the subjects' prediction on it.

3.6 Overal Estimation

We also asked our subjects to draw their confidence about the overall performance on the questionnaire. The plots can be found in the filled questionnaires in appendix **??**, but are not used in our evaluations. The values of these pdfs could be used in further experiments on the topic.
We finish with a following discussion about our methods and the presented results in the next section.

4 Discussion (Maximilian A. Gehrke)

Our goal was to design and evaluate an experiment which measures how well people estimate their performance. We created a questionnaire, where subjects had to solve eight sorting tasks. The questionnaire had two conditions: sorting without and with active recall. When sorting without active recall, we provided the answers to the task in a randomized order. The subjects simply needed to bring the given items into the correct order. When sorting with active recall, we did not provide any hints to the solution. The subjects had to actively recall the items in question and bring them into the correct order. In total we collected seven tasks for condition one and one task for condition two. After finishing a task, the subjects had to estimate their performance by drawing a probability density function.

After collecting the answers from fourteen subjects, our program read the probability density functions, extracted the probabilities at discrete points and normalized these probabilities. Additionally, the program scored the answers of each subject for each task. Then we used these two quantities to calculate the Brier score, a score to measure the accuracy of probabilistic predictions, for each participant and each task.

Unfortunately, the active recall item of condition two turned out to be inappropriate for determining the accuracy of performance self-evaluation, which is why we excluded it from the main part of our evaluations (for more detail see next section).

In sections 4.1 to 4.3 we discuss the creation and design of the questionnaire. We review different task types (4.1), discuss

the detection of suitable tasks (4.2) and emphasize the importance of explaining probability density functions (4.3). Next, we talk about finding the correct metric to score the subjects answers (4.4), argue why we deem the Brier score a good function to measure accuracy of probabilistic predictions (4.5) and stress the importance of computer vision for a task like this (4.6). Finally, we discuss the results of the experiments and it's implication for the future in section 4.7.

4.1 Finding the task type

Finding a task type that ensures a high level of evaluation objectivity was quite challenging. This is mainly due to the nature of our preconditions:

(1) Each task must be answerable in a few minutes.

(2) The task type allows the design of easy, moderate and difficult tasks.

(3) The task type does not allow subjects to easily assess the exact number of points they achieve for a task.

We decided to add precondition (1) for practical reasons. Precondition (2) enabled us to ask questions with difficulties and thus eliminating the potential bias that the prediction of task performance is correlated with task difficulty. The idea behind precondition (3) is to avoid counting of correct answers and to introduce enough uncertainty that leads to a meaningful conclusion about metacognition when self-evaluating task performance. Counting the exact number of points and specifying this number does not give us insights about the metacognitive processes of the human mind.

Some of our top choices for task types were text-based math, spelling, grammar, translation, estimation, mapping and fill out tasks. We discarded all of these types, because none of them allowed us to define an objective evaluation procedure while ensuring that subjects could not easily track their exact performance. The problem of most task types is the requirement of active recall for two reasons:

- Active recall items lack evaluation objectivity. It is not possible to write an evaluation scheme that considers all possibilities. In addition, even if we manage to create such an evaluation scheme, it would need plenty of resources and is susceptible to errors. Any possible answer that we forget has to be incorporated into the evaluation scheme and considered in all previous evaluations. Also, the experiment instructor would need to have a certain level of expertise in the domain, which limits the evaluation objectivity and again takes a great deal of resources. Last, we would not be able to ask questions of different areas. In the end, we excluded task types that did not allow for a simple evaluation metric.

- Active recall produces zero-point-answers and flooring effects. If a task type requires active recall, participants that do not know anything about the task, will get zero points. This is undesirable, because participants can perfectly estimate their performance for such tasks. In other words, flooring effects (as well as ceiling effects, if the task is too easy, e.g. simple math tasks), violate precondition (3). Subjects can too easily assess their performance of the task and therefore estimate their performance perfectly.

To confirm our assumptions, we added an active recall sorting task at the end of our questionnaire. And indeed, the task induced a flooring effect and all subjects depicted high accuracy in task performance with a mean Brier score of XXX. Maybe with a better scoring metric the item would render useful (we calculated the L2-norm as for our other sorting tasks, for more detail see section 4.4), however we could not figure out a metric that did not seem arbitrary. That is why we suspended the item from the remaining evaluation.

The other tasks of our questionnaire were sorting tasks without active recall. Sorting tasks allow for a fine grained assessment when applying a suitable metric (see section 4.4). At the same time it is hard for participants to estimate their exact performance, because it is hard to incorporate the distance between the correct placement of an item and the chosen placement of an item. Further, flooring effects are avoided by providing the items in question in a randomized order.

We are interested in other suitable task types and if other researchers share our preconditions. So far it was not possible for us to locate the reasoning behind the task types of other experiments that target the prediction of task performance.

4.2 Finding suitable tasks

We want to take the time to discuss what we define as suitable tasks. We think this is a big issue when designing performance evaluation questionnaires and should be taken more into account. The design of tasks needs to be done very carefully in order to draw the coherent conclusions after executing the experiment. Especially when comparing studies, people need to be able to examine exactly how the tasks were structured. One might argue that in self-evaluation experiments it is only necessary to look at the performance predictions as well as the task result and that the specific task assignment is negligible. However, it is crucial to know why a person achieved a certain level of points to resolve biases and with it differences between conclusions of different studies. Only with the knowledge of task designs biases can be tracked and eliminated.

In our study, we specifically set the focus on the capacity of the short term memory, a broad selection of knowledge, a variety of different task difficulties and the prevention of floor and ceiling effects.

Many studies show that humans can hold about 7 ± 2 items in their short term memory. Following this reasoning, we decided to target the lower bound and provided 5 items per task to sort. This way we would only test the knowledge of the subject and not the ability to use memory efficiently. At the moment, we do not know if subjects are better in evaluating their performance regarding knowledge or regarding memory capacity. Only if we report the task design, we are able to draw coherent conclusions between different studies. The same issue displays regarding the knowledge we are inquiring. Maybe self-evaluation works better in some areas than in others. To prevent this possible bias, we decided to incorporate different areas. Analogue with this idea, we provided tasks with different difficulties to avoid the degree of difficulty as a potential bias factor. It is however very interesting how subjects evaluate their task performance when they are good at a task and when they are poor at a task.

In principle, tasks that exhibit floor or ceiling effects do not corrupt our conclusion. The reason behind this is that we are not so much interested in the tasks outcomes, but rather how subjects rate their performance. However, floor and ceiling effects still pose a problem, because it induces bias. If subjects absolutely do not know anything about a task, they will most probably get zero points. Not knowing anything at a task is a strong indicator for getting zero points, so subjects will also rate their performance with about zero points. Analog, if a task is very easy, subjects receive a good score and rate their performance with the maximum number of points. Hence, ceiling and floor effects unnaturally increase the accuracy of task predictions and therefore bias the results. Tasks should be constructed and tested in a way that prevent floor and ceiling effect. This reasoning is the same as for a suitable selection of task types (see last section).

Future research could target different ways to design questionnaires. Our design targets the performance in knowledge tasks across domains with different difficulties and no ceiling and flooring items, however it does not shed light onto the questions, if level of difficulty, incorporation of memory (e.g. more items or different tasks) or specific knowledge domains change the self-assessment of performance.

4.3 Explaining probability density functions

The idea of using probability density functions for gathering uncertainty was very intriguing to us. To incorporate probability density functions, we needed to make sure that subjects understand the concept of probability density functions and specifically how we use them in our experiment.

We designed a one page explanation on how probability density functions work and provided examples. We took care of normalizing the probability density functions after administering the experiment and let the subjects know that they did not need to worry about it. However, subjects still struggled to understand how probability density functions work. Even people who knew probability density functions beforehand needed some time to understand how they are supposed to be used in this case. The confusion stemmed from the similar labels of the x and y axes. The x axis delineates the percentage of points that the subject could get for the task $(0 - 100\%)$. The y axis shows the probability, ranging from 0 to 1, that the subject assigns for possibly having X% (ranging from 0 to 100) of the task correct (for more detail see section 2).

Probability density functions are certainly not easy to understand. We have two suggestions for future administration of the questionnaire:

- We highly recommend using an example task. We already incorporated example probability density functions, which was not enough. Subjects reported that they would have liked an example task, where it is possible to drawn a probability density function for practice.
- We recommend to orally check if the subjects correctly understood the usage of probability density functions. We did this with most of our subjects and are quite certain that it increased the overall understanding of probability density functions in our experiment.

It is important to make sure that the task is clear and that the subject has understood how to utilize probability density functions. Otherwise we do not measure the uncertainty of performance prediction, but instead how well participants have understood the usage of probability density functions.

Additionally, it is important to convey an understanding how the probability functions look once they are normalized. This can also be done implicitly by explaining in the instructions that participants should draw the peak of a probability density function twice as high then the rest, if they are twice as confident about a certain value than the remaining values.

4.4 Finding the evaluation metric

After deciding to use sorting tasks, we needed to find a scoring metric that induces enough uncertainty and does not simply count the number of items that were assigned to the correct positions. We need enough uncertainty to draw meaningful conclusions about metacognitive processes when self-assessing task performance (see first paragraphs of section 4.1 for more information). Our idea was to find a metric that incorporates the distances between the answer given and the correct answer.

We accomplished this by taking the L2-norm between the positions of the given answers and the positions of the correct items (an example can be seen in section 2). One caveat however is the continuous scale of the L2-norm. To calculate the Brier score (our scoring function to measure accuracy of probabilistic predictions, see next section), we need discrete values. That is why we calculated the worst L2-norm possible and split the interval of [0, L2-max-norm] into an array of a fixed amount of equidistant numbers (in our case 6). The discrete rating for a task is then the 'id' of the closest value of that array compared to the L2-norm of the answer.

We found this metric and scoring procedure highly adaptable. It scales well with both the number of items to sort and the amount of points we want to assign to each task. It can also be used to assign the same amount of points to tasks with different amount of items. Embedding this would be a good starting point to administer further research on this project.

To avoid bias, it is important that the subjects know the scoring procedure of the tasks. The exact scoring function (in our case a function including the L2-norm) is not as important as letting the subjects know the factors that are included into the evaluation. In our case, we incorporated the distance between the correct answer and the given answer. An answer that is close to it's correct placement will still give some points. This drastically shifts the perception of what a good or bad answer is. It is important that subjects incorporate this knowledge when drawing the probability density functions to draw valid conclusions.

We forgot to write this information into the instructions, which we would propose for all future research. During our first session, the subject asked for the grading procedure and from that onward we orally gave this information before the people started filling out the questionnaire.

4.5 Brier Score

We use the Brier score to measure the self-assessment of subjects. The Brier score is a proper score function that measures the accuracy of probabilistic predictions, which is exactly what we want to know. Because the Brier score only takes discrete values, we calculate and assign each answer a discrete point value. How exactly we do that is described in the section above.

The Brier score measures the mean squared differences between the occurrence of an event and the assigned probabilities. The best possible value of the Brier score is zero and the worst possible value is one. The Brier score assigns events that occurred a value of 1 and events that did not occur a value of 0. Therefore, the best score of $BS = 0$ can only be reached if the subject assigns a probability of 1 to the occurring event and all others 0. In the form of probability density functions subjects would need to draw a very steep summit at the location of the occurring event and with zero probability everywhere else. This is theoretically possible, but in practice subjects will draw a more smooth function. Hence, a Brier score of 0 is rarely reached, even if subjects assigned most of the probability to the correct answer.

The Brier score is more forgiving for two small errors than one large error. Equation (4) shows this in numbers:

$$(0.05 - 0)^2 + (0.05 - 0)^2 = 0.005 < 0.01 = (0.1 - 0)^2 \tag{4}$$

Two probabilities of 5% (= 0.05) ascribed to a non-occurring event (= 0) result in a smaller Brier score proportion than one large error of 10% (= 0.1) ascribed to a non-occurring event. This is due to the fact that the Brier score calculates the squared difference between the occurrence of an event and the probability ascribed to it. This feature is very helpful when examining self-assessment of task performance with probability functions. We want to credit a subject most for drawing the peak at the correct location. If a subject sill assigns some low percentage for all other events, this will not have a strong impact on the result.

4.6 Analysis with Computer Vision

One of our main difficulties was to find a good method to gather probability density functions. We wanted a digital representation of the probability density functions that the subjects would draw. This way, the computer could read the probabilities of the probability density functions without a human painfully measuring the distances with a ruler.

We initially had the idea that subjects draw the probability density functions on a computer inside a dedicated window. However, we decided against it, because it is rather hard to draw on a computer with a mouse. Drawing by hand is more accurate and easier for subjects. We developed our questionnaire in a way that let us use computer vision to detect and extract the probability density functions as images. We then used these probability density function images to read the probability at certain discrete points and calculate the metrics we were interested in.

We can recommend this or a similar approach. An automatic processing pipeline is much less error prone and ensures a high level of objectivity. If correctly implemented the complete evaluation process is independent of the instructor.

In the future, we think it would be beneficial to incorporate even more computer vision. When designing the questionnaire, we added little squares on the right of the answer panels. We used these to evaluate the subject's answers by assigning numbers from one to five for the correct placement of each item. This helped us to created a CSV file with the ordering the subjects selected for each answer. Originally we intended to read the squares with the help of computer vision and then apply Machine Learning to recognize the hand written digits. However, due to time constraints we were not able to do so. This would be a great next step to do or extracting and recognizing the answers in the answer panels.

4.7 Experiment results

On average, our subjects received a Brier score of $BS \approx 0.14$ for each task. This is better than we expected. A Brier score of $BS \approx 0.14$ is equal to ascribing the occuring event a probability of 70% and two other events 15%. If we would distribute the remaining 30% across all remaining events, the Brier score would even fall further.

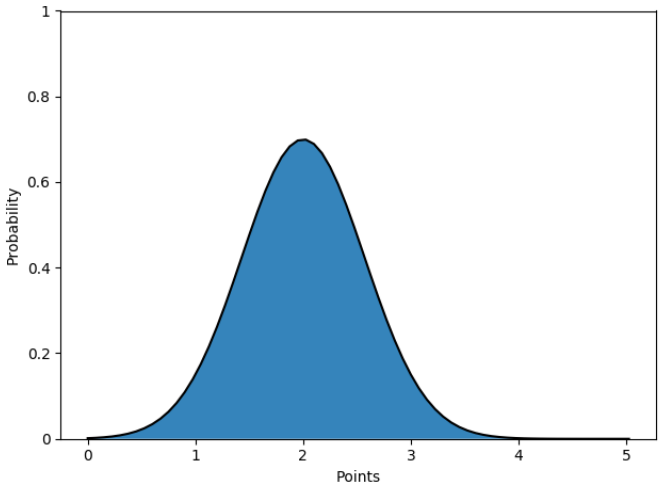


Figure 13: A Brier score of 0.14 expressed as a probability function over a task that would give 0 to 5 points with the correct answer at 2.

This is in comparison to what we find in the literature. In the existing studies, researchers show that humans are quite bad when self-assessing their performance after executing task. We do not know why this discrepancy exists and would like to see some further research in this field that trail possible biases by executing controlled studies. Assigning the occurring event a probability of 50%, would be at least result in a Brier score of $BS = 0.25$. Allocating the same probability for all possible events always results in a Brier score of $BS \approx 0.84$.

In future research we would like to administer a study with more subjects, find a not so homogeneous group (we only administered white German people, mainly from university).

References

[1] Paul Azzopardi and Simon Evans. Evaluation of a'bias-free'measure of awareness. *Spatial vision*, 20(1-2):61–77, 2007.

[2] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[3] Frank R Clarke, Theodore G Birdsall, and Wilson P Tanner Jr. Two types of roc curves and definitions of parameters. *The Journal of the Acoustical Society of America*, 31(5):629–630, 1959.

[4] Harald Cramér. *Mathematical methods of statistics*, volume 43. Princeton university press, 1999.

[5] Stephen M Fleming and Hakwan C Lau. How to measure metacognition. *Frontiers in human neuroscience*, 8:443, 2014.

[6] Susan J Galvin, John V Podd, Vit Drga, and John Whitmore. Type 2 tasks in the theory of signal detectability: Discrimination between correct and incorrect decisions. *Psychonomic bulletin & review*, 10(4):843–876, 2003.

[7] Leo A Goodman and William H Kruskal. Measures of association for cross classifications iii: Approximate sampling theory. *Journal of the American Statistical Association*, 58(302):310–364, 1963.

[8] David Marvin Green, John A Swets, et al. *Signal detection theory and psychophysics*, volume 1. Wiley New York, 1966.

[9] Nigel Harvey. Confidence in judgment. *Trends in cognitive sciences*, 1(2):78–82, 1997.

[10] Yoonhee Jang, Thomas S Wallsten, and David E Huber. A stochastic detection and retrieval model for the study of metacognition. *Psychological Review*, 119(1):186, 2012.

[11] Justin Kruger and David Dunning. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of personality and social psychology*, 77(6):1121, 1999.

[12] Craig Kunimoto, Jeff Miller, and Harold Pashler. Confidence and accuracy of near-threshold discrimination responses. *Consciousness and cognition*, 10(3):294–340, 2001.

[13] Brian Maniscalco and Hakwan Lau. A signal detection theoretic approach for estimating metacognitive sensitivity from confidence ratings. *Consciousness and cognition*, 21(1):422–430, 2012.

[14] Michael EJ Masson and Caren M Rotello. Sources of bias in the goodman–kruskal gamma coefficient measure of association: Implications for studies of metacognitive processes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(2):509, 2009.

[15] Allan H Murphy. A new vector partition of the probability score. *Journal of applied Meteorology*, 12(4):595–600, 1973.

[16] Thomas O Nelson. A comparison of current measures of the accuracy of feeling-of-knowing predictions. *Psychological bulletin*, 95(1):109, 1984.