How to use probability density functions to track performance self-assessment

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 performance after they have fulfilled a task. Probability density functions is a good way to measure the own evaluation of performance. To our knowledge, this has not yet been implemented. Which is why we wanted to test if and how one can use probability density functions in an experiment to extract the self-evaluation of task performance.

We designed a questionnaire where subjects have to draw probability density functions after executing simple sorting tasks. We evaluated each task using a norm scoring function and assigned each task 5 points. Last, we calculated the Brier score, a score for measuring uncertainty, and came to the conclusion that humans are about average in evaluating their performance.

The main aspect of this report is the design of a questionnaire which uses probability density functions to assess performance self-evaluation.

1 Introduction

- Measuring metacognition is important because .. ?
- Metacognitive sensitivity ?
- Metacognitive bias?
- Metacognitive efficiency?
- Useful starting point for all measures of metacognitive sensitivity: 2x2 confidence-accuracy table ("type 2" SDT table)
- Simplest measure of association between rows and columns of the table is the phi(ϕ) correlation.
- G: Goodman-Kruskall gamma coefficient
- Both are affected by metacognitive bias
- Standard way to remove influence of bias: Apply STD (d')
- d' will be constant given different biases
- Several approaches to metacognitive sensitivity
- But type 2 d' is also affected by changes in metacognitive bias
- One solution: Apply non-parametric analysis that does not make equal-variance gaussian assumptions (e.g. use ROC analysis)
- Can be applied to type 2 data.
- Further complication: ϕ, G, d' and type 2 ROC are affected by the task performance.

2 Method

Task: describe your ideas and your realization of the task

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 question 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 **??**.

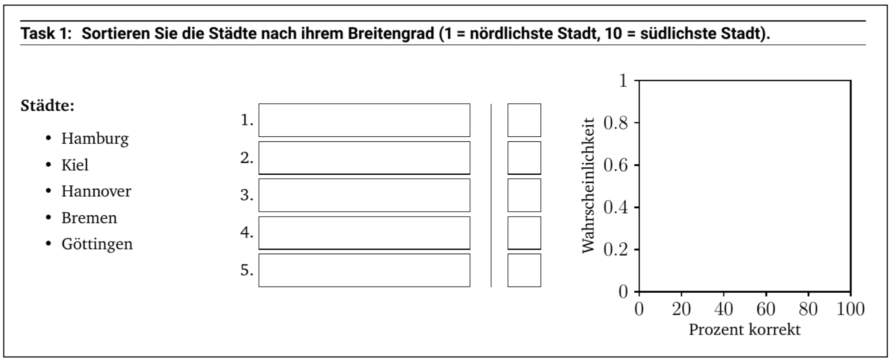


Figure 1: Example closed-form sorting task.

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 **??**.

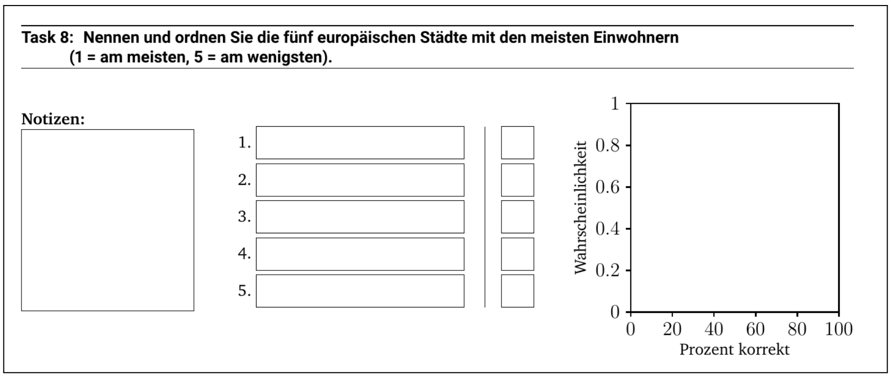
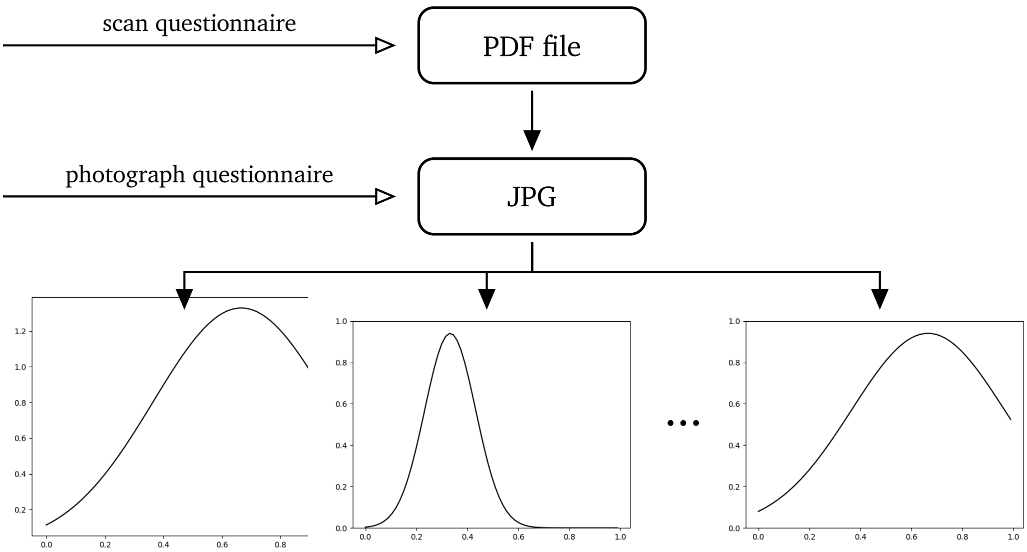


Figure 2: Example open end sorting task.

2.2 Processing Pipeline



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.

3 Results

4 Discussion

What to do?

- discuss challenges that you faced during implementation,
- reflect your solution
- give an outlook

Our universal goal was to design an experiment which can understand how well people estimate their performance. Therefore is our study a preliminary study. For this we designed an experiment, where subjects had to solve several five item sorting tasks. After finishing each task, the subject had to draw a probability density function over their performance. The questionnaire had two conditions. The first condition was sorting without active recall. The five answers to the question were already given in a randomized order and had to be sorted in the write order. The second condition was sorting with active recall. The question was open and the subject had to know the items and the correct ordering. In total we had 7 questions for condition one and one question for condition two.

After collecting the answers from XX subjects, our program read the probability density functions with the help of computer vision and we analyzed the the Brier score of the data.

Objectivität Finding the correct type of questions.

Finding the correct questions.

Finding the correct metric.

Explaining the concept of probability density functions.

Computer Vision.