

THE DESIGN AND IMPLEMENTATION OF A VISUAL ANALYTICS TASK TO SUPPORT  
EXPERIMENTAL RESEARCH ON HUMAN REASONING WITH UNCERTAIN  
KNOWLEDGE

BY

SHUO FENG

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Electrical and Computer Engineering  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Professor Alex Kirlik  
Professor Lav R. Varshney

## **ABSTRACT**

This research project involved designing and implementing a web-based application to support research using visual analytics, or the use of interactive visualizations, to support human cognition. More specifically, the interactive visualization that was created was motivated by the problem that humans often express overconfidence in both judgments and predictions based on uncertain knowledge. The interactive visualization presents experimental participants with a series of binary (yes/no, T/F, etc.) general knowledge or prediction questions, and requires participants to answer these questions and also provide a probability or confidence estimate between 50% and 100%. The output of the software created is a quantitative measure of human performance in terms of both accuracy and latency. This web-based application, which can also be used in stand-alone (non-networked) mode, is expected to pave the way for a set of additional future research projects involving experiments with human participants, with the eventual goal of interface design approaches and guidelines for eliciting unbiased information from knowledgeable people when either their subjective knowledge or the judgment or prediction task itself is characterized by uncertainty.

## **ACKNOWLEDGMENTS**

This research was supported by National Science Foundation grant #1330077, “CPS: Synergy: Collaborative Research: Engineering Safety-Critical Cyber-Physical-Human Systems” to the University of Illinois, Dr. Alex Kirlik, Principal Investigator. The author would like to thank John Nguyen and Shelby Abrahamsen for their assistance in this work.

## TABLE OF CONTENTS

CHAPTER 1: BACKGROUND .....	1
CHAPTER 2: SCORING RULES AND OVERCONFIDENCE .....	3
CHAPTER 3: SOFTWARE IMPLEMENTATION .....	5
CHAPTER 4: EXPERIMENTAL INSTRUCTIONS .....	6
CHAPTER 5: CONCLUSIONS AND FUTURE WORK .....	11
REFERENCES .....	12
APPENDIX: CODE SNIPPET .....	13

## **CHAPTER 1: BACKGROUND**

Overconfidence is a prevalent phenomenon in judgment and forecasting tasks in experimental settings as well as in professional contexts (Tetlock, 2006) (Griffin & Tversky, 1992). In this thesis, a “cognitive engineering” intervention (Lee & Kirlik, 2013) is presented in terms of a computational experimental task based on visual analytics to support future research aimed at reducing the overconfidence bias in both subjective knowledge judgments and predictions of future events. The visual analytics approach is an innovative technique for investigating the possibility that one factor contributing to overconfidence is the opaque nature of the evaluation techniques used to assess the quality of probabilistic judgments in a variety of professional contexts. There are many real-world scenarios where this intervention can be applied and make an impact. For example, organizations conducting surveys will benefit if they can get more accurate knowledge and predictions from clients. In another case, individuals will avoid making senseless decisions when investing money or planning tasks (Barber & Odean, 2000). And online shoppers would be less likely to be fooled by websites (Tan et al., 2012).

This research involved implementing visual analytics intervention by introducing a novel elicitation approach for debiasing confidence judgments associated with whether performers have answered a knowledge-based question correctly. The interactive visualization system we have created includes the ability to experimentally address overconfidence in both subjective, factually-based knowledge, as well some predictions about objectively uncertain future events.

A reframing of the Brier score (Brier, 1950) is used to incentivize truth-telling in subjective reports. The reformulation normalizes the score to expected error due to the chance or base rate, which is the foundation for the visual analytics tool.

The next few chapters are organized as follows: We first address details of the scoring rules and the overconfidence phenomenon. Then we summarize the implementations and programming. Finally, we discuss the results and future work.

## CHAPTER 2: SCORING RULES AND OVERCONFIDENCE

In terms of measuring the accuracy of confidence judgments, it is important to model the bias using mathematical functions. In academia and industry, there are multiple ways of doing that such as spherical, logarithmic, and the Brier (or quadratic) scores. These are especially favorable due to the fact that they are incentive-compatible under the risk neutrality assumption.

The most popular function to measure the accuracy is the Brier score, which is most commonly formulated as follows:

$$BS = \frac{1}{N} \sum_{t=1}^N (f_t - o_t)^2.$$

As illustrated in the function, the squared term models the deviation between predicted probabilities for a set of events and their true values. The term  $f_t$  represents the probability provided by the judges whereas  $o_t$  represents the actual binary outcome of the event (e.g., 0 or 1 for binary events, which are the focus of the research).  $N$  is the number of judgments made. Perfect performance results in zero, while the worst possible performance gives a value of unity.

The rationale behind using the Brier score is as follows. When making a random guess for a particular binary question, the users will get equal amounts of rewards and penalties. When claiming 100% confidence, they will be given more penalty if their answer is wrong, which causes the penalty to be 0.75 and the reward to be 0.25. Due to this asymmetrical property of the Brier rule, we hypothesize that experimental participants provided with the score will be more

cautious and well-calibrated users. If the hypothesis is confirmed, this experimental tool can be developed for real-world use by professionals and others who make judgments and predictions.



### **CHAPTER 3: SOFTWARE IMPLEMENTATION**

In order to achieve the goal of the experiment, we designed an interactive web application. The web application provides users with questions and gives them immediate feedback when they choose their confidence levels. More specifically, users will be given either a factual question or a predictive question at each step. They will then use either a parabolic slider or a linear slider to input their confidence level. The linear slider is used for the control group. The user will be given a certain amount of rewards based on the correctness and confidence of their answers. These rewards will be saved and added to get the overall rewards. After all the subjects finish their questions, the report of their performance will be generated and can be used for significance testing.

The code is mainly written in HTML, CSS, JavaScript, and C3.JS. The HTML is used to define the layout of the page. The CSS is used to define the style of the components on the page such as color, fonts, and layout. The JavaScript is used for displaying the graphics and business logic. C3.JS is mainly used for drawing the slider. The benefits of C3 are that it wraps the complex JavaScript and uses the canvas to display the plot. A snippet of code is shown in the Appendix. Users can see the convenience of using C3 such as appending attributes to the canvas to change the plot properties. The challenging part about the parabolic slider is how to handle a series of events such as hovering the mouse over the slider, dragging and releasing the mouse, and re-dragging and releasing the mouse. Each of these events can trigger a signal in JavaScript. If the user drags more than once, we need to update the results of the previous drag. We designed our program very carefully and covered a lot of corner cases similar to the one mentioned above.

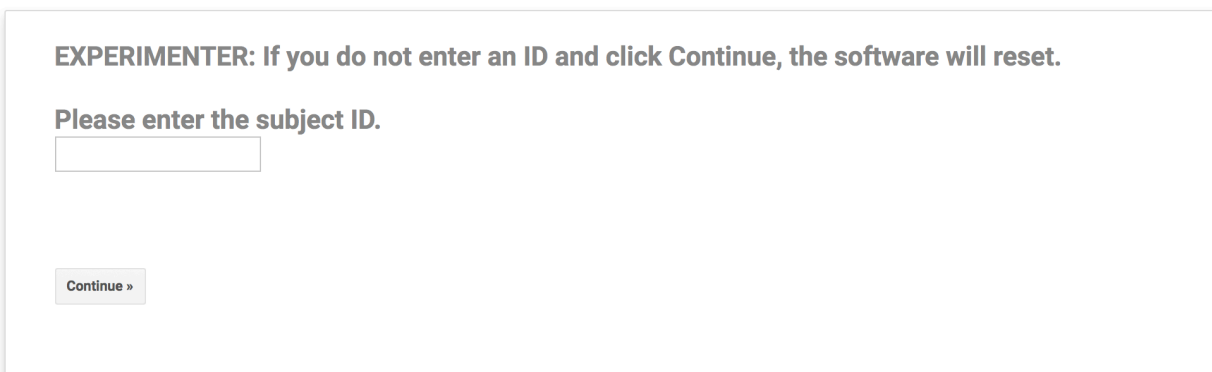
## CHAPTER 4: EXPERIMENTAL INSTRUCTIONS

### 4.1 Process the Experimental Input

Questions and answers are read automatically once the program is started. The input file is a simple Excel file which is automatically converted into the proper file format for the program. Users put their input files under the designated folder directory.

### 4.2 Preparation

There are some steps to process users' information and get users familiar with the system. Experimenters input the subject ID and the type of slider they want. It can be either a linear or a parabolic slider. After that, users will be given a few sample questions and learn how to answer the questions. They will learn the gain and loss tradeoff from the plot and drag the slider to show their confidence level. The prompts to type in ID and to choose groups are illustrated in Figures 1 and 2. Then a detailed introduction is given as illustrated in Figure 3.



**EXPERIMENTER: If you do not enter an ID and click Continue, the software will reset.**

**Please enter the subject ID.**

**Continue »**

Figure 1. The prompt for experimenters to input subject ID

**EXPERIMENTER:** If you do not select a group and click Continue, the software will reset.

**What is the subject's group?**

- ☐ Group 1: Linear Slider
- ☐ Group 2: Parabolic Slider

Continue »

Figure 2. The prompt for experimenters to choose linear slider or parabolic slider

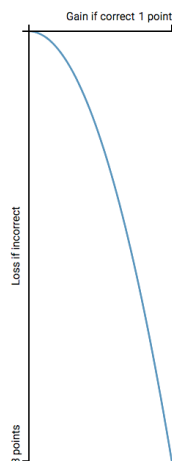
**Part 1)**

**Example 1: Was Donald J. Trump over 40 years old when he became the U.S. president?**

- ☐ Yes
- ☐ No

**Part 2)**

Click anywhere on the curve so that the slider appears, then by moving the slider, specify your answer to Part 2. As you move the slider on the curve, the green bar always shows how many points you will earn if your answer is correct. The red bar shows how many points you will lose if your answer is incorrect. Move the slider on the curve to a point where the relative sizes of the green and red bar represents what you are truly willing to gain or lose in case your answer is correct or incorrect.



Based on your knowledge and experience, you may believe it is more likely that Donald Trump was over 40 when he became president, so choosing the "Yes" answer is more appropriate.

Since you have some evidence to support the "Yes" answer, it may be acceptable to move the slider all the way to the bottom of the curve.

The right place for the slider is somewhere in between where your belief about the likelihood of winning versus losing justifies the length of red and green bars at that point. The more you believe your answer is correct, the more you want to move towards the bottom.

Continue »

Figure 3. Instruction screen with explanation for participants to get familiar with the rules

After the first few steps, the factual and predictive questions will alternate and users need to answer a set of questions until they are required to rest for a while. Users are divided into two groups. The interfaces shown in Figures 3 and 4 ask participants to first provide answers to a binary question. Then they use the slider to adjust their confidence between 50% and 100%. When participants click anywhere on the graph, the slider will appear. The experimental group uses the parabolic slider while the control group uses the regular slider. For the control group, the number on the slider indicates the percentage confidence participants have. Once participants are done, they click the Continue button to proceed to the next question. The slider will disappear, and participants must click somewhere on the graph to make it appear again.

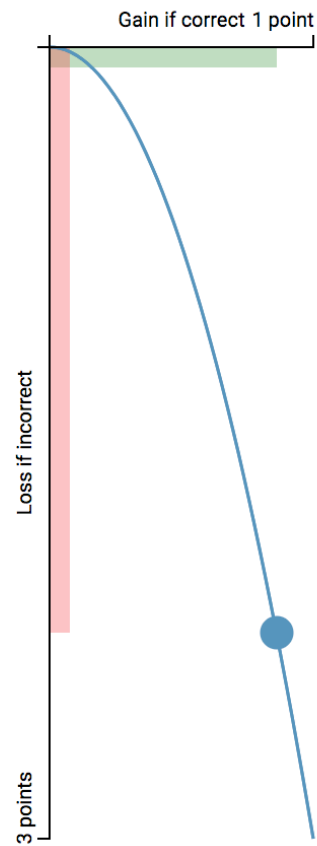
Figure 4. The baseline (control condition) linear slider elicitation

On the other hand, users are given a parabolic slider to adjust their confidence. As shown in Figure 5, the horizontal axis is the gain corresponding to the probability, while the vertical axis represents the loss. The graph shows that the ratio of loss to gain is three when users have 100% confidence in their judgments. Meanwhile, there is no difference between loss and gain when users input 50% confidence. The color bars indicate the loss or gain and will always align with the coordinate of the slider.

Is our moon larger than Jupiter's moon Europa? \*

☒ Yes

☐ No



Continue »

Figure 5. The parabolic slider visual analytics elicitation interface

### 4.3 Generate the Experimental Output Report

The report of the result will be automatically generated and exported as a JSON file. The report stores the correct answer, the user's answer, their corresponding confidence, and the time elapsed for each question. The result can be easily exported into statistical analysis software such as SPSS or R to find the significance for each one. The average Brier scores for both factual and predictive questions are included separately as items in the output file for each subject. Sample output is shown in the Appendix.

### 4.4 Deploy on Server

The application can also be easily hosted on a server so that anyone can access it through the Internet. This will enable non-local data collection and distribution of the task to research colleagues working in related areas to use this task for their own experimental purposes.

The source code can be downloaded at <https://github.com/data-slinky/confidence-elicitation.git>.

## **CHAPTER 5: CONCLUSIONS AND FUTURE WORK**

We designed and implemented a visualization tool for confidence elicitation using Brier score. The application can be used and deployed easily to conduct experiments on eliciting knowledge from experimental participants in uncertain judgment and prediction tasks. By transforming the presentation of the Brier scoring rule from a pure loss (error) domain into the domain of gain/loss tradeoffs, and communicating this concept to participants through an interactive visualization intended to make transparent the asymmetric and nonlinear nature of the Brier penalty function, we hope to find beneficial effects on judges' performance.

This project can be extended to perform more tasks. For instance, more debiasing mechanisms can be integrated into the systems so that we can make comparisons among them and examine the overall effects. One example is to use a bank to directly show the results after a certain number of questions. For example, Bickel (2007) compared quadratic scoring and spherical and logarithmic scoring rules. We can modify our work and perform statistical analysis to see which one has better effects.

## REFERENCES

- Barber, B. M., and T. Odean. (2000). Trading is hazardous to your wealth: The common stock investment performance of individual investors. *Journal of Finance* 60(2), 773-806.
- Bickel, J. (2007). Some comparisons among quadratic, spherical, and logarithmic scoring rules. *Decision Analysis* 4(2), 49–65.
- Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review* 78, 1-3.
- Griffin, D., and A. Tversky. (1992). Weighing of evidence and the determinants of confidence. *Cognitive Psychology* 24, 411-435.
- Lee, J. D., and A. Kirlik, eds. (2013). *The Oxford Handbook of Cognitive Engineering*. New York: Oxford University Press.
- Tan, W.K., C.-H Tan, and H.-H Teo. (2012). Consumer-based decision aid that explains which to buy: Decision confirmation or overconfidence bias? *Decision Support Systems* 53(1), 127-141.
- Tetlock, P. E. (2006). *Expert Political Judgment: How Good is It? How Can We Know?* Princeton, NJ: Princeton University Press.



## APPENDIX: CODE SNIPPET

CODE: Sample Input:

```
{  
  Does Canada have a greater land area than the U.S.?2, TRUE, TRUE,single-select,"""Yes""",  
  ""No"""]"  
  "According the 2010 U.S. Census, were there more men than women living in the United  
  States?" ,3,TRUE,TRUE,single-select,"""Yes""", ""No"""]"  
}
```

Sample Output:

```
{"0":["GROUP","Group 2: Parabolic Slider",0.24615384615384617,0],  
"1":["Yes","Yes",0.19038461538461537,3644438],  
"2":["Yes","Yes",0.19038461538461537,3644438],  
}
```

C3.JS sample code:

```
var line = d3.svg.line()  
  .x(function (d) {  
    return x(d.q);  
  })  
  .y(function (d) {  
    return y(d.p);
```

```
});  
container.append("path")  
    .datum(data)  
    .attr("class", "line")  
    .attr("id", "lineId")  
    .attr("d", line);
```

```
handle1 = [{  
    x: 0,  
    y: 0  
    }];
```