Contents

Supplementary Information for: ANDURIL - A MATLAB Toolbox for ANalysis and Decisions with UnceRtaInty: Learning from expert judgments

Georgios Leontaris, Oswaldo Morales-Nápoles

Civil Engineering and Geosciences, Delft University of Technology

Abstract

This document provides supplementary information regarding Cooke's classical model and the developed MATLAB toolbox ANDURIL[1]. The reader can find a detailed description of the functions that constitute ANDURIL as well as examples regarding the use of these developed functions, based on a recent real-life application. Different advantages of ANDURIL as well as its limitations and possible extensions are discussed.

1. Cooke's classical model for Structured Expert judgment

In practice, engineers, scientists and decision makers are often confronted with problems where sufficient relevant field data (measurements) are not available. In these cases, modeling or expert judgments become an alternative source of valuable data. For these reasons, Cooke in [1] has developed a method (i.e. Cooke's classical model for structured expert judgment) to

[1]In order to avoid confusion of the minority of people, who are not familiar with the universe of Lord of the Rings by J.R.R. Tolkien, the authors would like to clarify the inspiration for the name of the developed Matlab toolbox. Andúril was the name of the sword of Aragorn, the son of Arathorn, which was reforged from the shards of Narsil (the sword that was used by Isildur to cut the One Ring from Sauron's hand). Excalibur is also the name of the legendary sword of king Arthur. Similarly to the sword, the source code of EXCALIBUR software remained accessible only to a few worthy ones. Therefore, the researchers and practitioners could only admire and use the software without being able to further investigate and explore developments of the method. To change this, the existing software had to be "broken to pieces" and then "reforged". Naturally, the name of the resulting new open-source Matlab toolbox is ANDURIL. Hopefully, this will help in bringing peace to troubled researchers and practitioners of Cooke's classical model.

aggregate expert judgments based on performance measures. Cooke's classical model is the most widely used method in practice. It has been used in many fields including the nuclear sector, chemical & gas industry, hydraulic engineering, aerospace and aviation, occupational safety, health, banking and volcanology to name some. Up to 2008 a total of 45 applications were collected in a databese [2]. Since then, at least 33 more applications have been performed [3].

In Cooke's Classical model, the experts assess their uncertainty over two types of continuous quantities. The first type corresponds to *target variables*. These are variables whose uncertainty cannot be sufficiently described using current models or field data and hence expert judgements are required. The second type of variables queried in the classical model are the so called *seed variables*. These are variables from the experts' field which are known to the (group) of analysts at the moment of the elicitation (or will be known to them post hoc) but whose true values are not known to the experts at the moment of the elicitation.

Experts are thus scored according to their performance in assessing uncertainty over seed variables. Their opinions are weighted and later combined on the basis of their performance. The purpose of the classical model is to enable rational consensus. According to [1], any methodology for structured expert judgment that aims at enabling rational consensus should comply with the following requisites:

1. Scrutability: All data and *processing tools* are open to peer review and results must be reproducible by competent reviewers.
2. Empirical control: Quantitative expert assessments are subjected to quality controls.
3. Neutrality: The method for evaluating expert opinions should encourage experts to state their true opinions.
4. Fairness: Expert opinions are not judged, prior to processing the results of their assessments.

In the majority of past studies the closed source software EXCALIBUR (freely available at http://www.lighttwist.net/wp/excalibur) that is only available for Windows OS, has been used for the analysis and aggregation of expert judgments. Recently, a number of cross validation studies have been conducted using Eggstaff's MATLAB code [4, 3]. However, this code is not publicly available and it still does not implement important features of the model such as the item weighting scheme [3].

Precisely in the spirit of contributing to guarantee that the condition of scrutability is further met, the MATLAB toolbox presented in this paper was developed. We believe that it is important for researchers to have

open access to a code that makes transparent the calculations of performance measures and the aggregation of expert judgments, so that current methods can be made more accessible and different approaches or extensions to current methods can be further explored. Therefore, the purpose of ANDURIL toolbox is to assist researchers or practitioners who are interested in Cooke's classical model, in applying the method or investigating further developments to it irrespective of their choice of operating system.

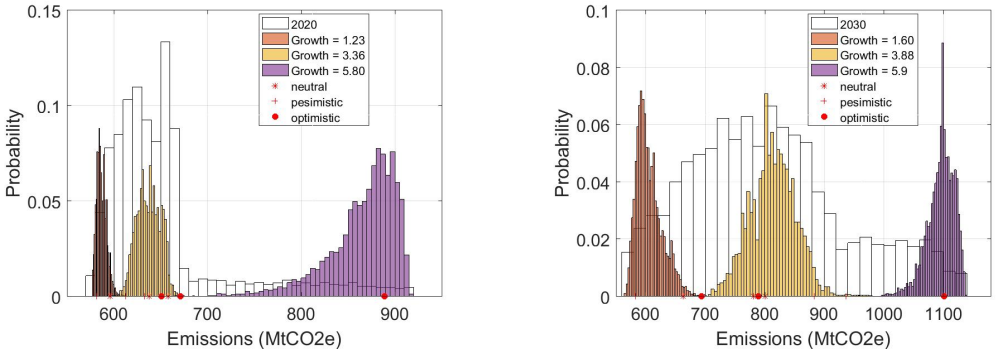


Figure 1: Uncertainty Distributions for GHG emissions in Mexico for 2020 and 2030

For the purpose of our presentation we will use a recent example related to estimation of uncertainty in green house gas (GHG) emissions presented in [5]. Figure 1 presents uncertainty distributions for GHG emissions for Mexico in 2020 and 2030 as obtained in [5]. These were constructed based on uncertainty estimates obtained through structured expert judgment using Cooke's classical model and the analysis was done using EXCALIBUR. Output such as the one presented in Figure 1 are typical results from the application of Cooke's method.

For the remainder of this supplement in section 2 we present the main theory in which Cooke's method and hence EXCALIBUR and ANDURIL are based on. Section 3 describes the main features of our developed MATLAB toolbox ANDURIL. Section 4 compares the output of ANDURIL with the one from EXCALIBUR for the data presented in [5]. Section 5 discusses some features available in ANDURIL not available in EXCALIBUR. Section 6 summarizes our main findings and discussions.

2. Statistical Accuracy & Information

Although some good descriptions of this structured expert judgment method can be found in literature, the main concepts of Cooke's classical model are summarized below. This with the purpose of making available to

the reader the main elements of the method and the code. For details and extensive discussion the reader is referred to [1] and supplementary material for [3].

In Cooke's classical model experts are asked to provide assessments of their uncertainty concerning continuous quantities in the form of a number of percentiles of their uncertainty distribution. Most commonly the 5^{th}, 50^{th} and 95^{th} percentiles are queried.

The percentiles are assessed for uncertain quantities which are in fact the *target variables* (or *variables of interest*). These percentiles are also queried for quantities whose value is known to the analysts (or will be known to the analysts within the time frame of the research), but is not known to the experts at the moment of the elicitation. These are called *seed* or *calibration variables* and are used to ensure empirical control of experts' uncertainty assessments. Examples of a seed variable and a variable of interest concerning the example study used in this paper for economic growth in Mexico are:

1. Seed variable: Quarterly growth rates of gross domestic product in Mexico have been below -5% in four instances between the first trimester of 1994 and the third trimester of 2013. What was the average value of the 28-day Mexican Federal Treasury Certificates (CETES) interest rate in these four trimesters? Indicate the 5^{th}, 50^{th} and 95^{th} percentiles of your uncertainty distribution.
2. Target Variable: Consider a scenario in which, at the end of 2020, the Mexican (commercial) interest rate is between 3.5 and 4.0 percent, the unemployment rate is between 5.4 and 5.6 percent, the inflation growth rate is between 3.0 and 3.3, and growth rates of gross domestic product in the USA are between 2.8 and 3.3 percent. Please provide your estimates (5^{th}, 50^{th} and 95^{th} percentiles of your uncertainty distribution) of average gross domestic product growth rate in Mexico up to 2020.

Seed variables are used to compute two measures of performance: *statistical accuracy* or *calibration* and *information*. We discuss these measures next.

2.1. Statistical accuracy

Assume we have answers from $e = 1, \ldots, E$ experts on $i = 1, \ldots, N$ seed variables and $1, \ldots, N_1$ target variables. Assume further that we assess three quantiles: $q_{i,5}$, $q_{i,50}$ and $q_{i,95}$ for the 5^{th}, 50^{th} and 95^{th} quantiles of each uncertain quantity. That is including the target variables. There are thus $j = 1, \ldots, 4$ interquantile bins. The procedures described next may be easily extended by assuming more quantiles are assessed from each expert. For each

quantity, each expert divides her belief range into four interquantile intervals, for which the corresponding probabilities of occurrence $p = [p_1, \ldots, p_4]$ are: $p_1 = 0.05$ for a realization value $\leq 5^{th}$ percentile, $p_2 = 0.45$ for a realization value $\in (5^{th}, 50^{th}]$ percentiles, $p_3 = 0.45$ for a realization value $\in (50^{th}, 95^{th}]$ bin, and $p_4 = 0.05$ for a realization value $> 95^{th}$ percentile. The empirical version of $p = (p_1, \ldots, p_4)$ for expert e, is denoted $s(e) = (s_1, \ldots, s_4)$, where $s_j(e)$ is equal to the number of realizations of seed variables falling in the j^{th} interquantile assessed by expert e divided by the total number of seed variables.

$$
\begin{aligned}
s_1(e) &= \frac{\text{Number of realizations} \leq 5^{th} \text{ quantile}}{N} \\
s_2(e) &= \frac{\text{Number of realizations} \in (5^{th}, 50^{th}] \text{ quantile}}{N} \\
s_3(e) &= \frac{\text{Number of realizations} \in (50^{th}, 95^{th}] \text{ quantile}}{N} \\
s_4(e) &= \frac{\text{Number of realizations} > 95^{th} \text{ quantile}}{N}
\end{aligned}
$$

One way to measure the difference between p and $s(e)$ is through relative information or entropy, which is a measure of the disagreement between them.

$$I(s(e), p) = \sum_{j=1}^{4} s_j(e) \ln \frac{s_j(e)}{p_j} \tag{1}$$

Experts' assessments are treated as statistical hypotheses. Consider for each expert the null hypothesis H_0 : The inter quantile interval containing the true value for each variable is drawn independently from the probability vector p.

The quantity $2NI(s(e), p)$ where $I(s(e), p)$ is given in equation (1) is asymptotically χ^2_3 (the degrees of freedom are the number of interquantile intervals minus 1). This quantity can be used to test H_0 and it defines the calibration score:

$$C(e) = P\{2NI(s(e), p) > r\} \tag{2}$$

The probability in equation 2 can be evaluated by a χ^2_3 distribution. The calibration score $C(e)$ is the probability that a deviation at least as large as r could be observed on N realizations if H_0 were true. Where r is the percentile of interest in the χ^2 distribution of interest obtained from

evaluating $2NI(s(e), p)$ for the data corresponding to a particular expert. Values of calibration close to zero mean that it is unlikely that the experts' probabilities are correct.

2.2. Information

The information score measures the degree to which a distribution is concentrated (or spread out) with respect to a background measure. In the classical model and in EXCALIBUR the uniform or log-uniform background measures are used. An intrinsic range is calculated for each expert's density. The intrinsic range is obtained by adding a $k\%$ overshoot to the smallest interval containing all quantiles and realizations (when available), where k is selected by the analyst (typically $k\% = 0.1$). The lowest (l) and highest (h) values for the intrinsic range are $l_i = min\{q_{i,5}(e), v_i\}$ and $h_i = max\{q_{i,95}(e), v_i\}$ where v_i is the realization of interest. Then $q_{l_i} = l_i - k(h_i - l_i)$ and $q_{h_i} = h_i + k(h_i - l_i)$. The *information score* is then computed as:

$$I(e) = \frac{1}{N} \sum_{i=1}^{N} \left[\ln(q_{h_i} - q_{l_i}) + p_1 \ln \frac{p_1}{q_{5,i} - q_{l,i}} + \ldots + p_4 \ln \frac{p_4}{q_{h,i} - q_{95,i}} \right] \quad (3)$$

Notice that the information score does not depend on the realizations (other than in terms of calculating the intrinsic range when available) and hence may also be computed for the target variables. When target variables are also considered, the summation in equation 3 runs to N_1 which includes target variables. This is actually commonly done in the classical model and implemented in EXCALIBUR and ANDURIL as will be seen later. Also notice that in equation 3 a uniform Background measure is applied. For a log-uniform background measure the log of $q_{\cdot,i}$ would be used instead.

2.3. Combination

In the classical model the combination of experts' assessments is called a *Decision Maker* (DM). This is a weighted average of individual estimates. When the weights are determined based on the performance of experts in the seed variables, we speak of *performance-based* DM. The DM probability densities $f_{DM,i}$, for every item i, are thus:

$$f_{DM,i} = \frac{\sum_{e=1}^{E} w_\alpha(e) f_{e,i}}{\sum_{e=1}^{E} w_\alpha(e)} \quad (4)$$

Observe that the weighs for each expert $w_\alpha(e)$ are given by the product of calibration and information scores when a certain threshold in calibration is attained. That is:

$$w_\alpha(e) = 1_{\{C(e)>\alpha\}} C(e) I(e). \tag{5}$$

Where $1_{\{A\}}$ denotes the indicator function for A. Values of $C(e) < 0.05$ would fail to confer the study the required level of confidence. Note that the DM can also be evaluated in terms of calibration and information. For this reason the DM is referred to as the "virtual expert". In the performance based DM the value of α is chosen such that the calibration score of the DM is maximized. The weights in Cooke's model are weakly asymptotically strictly proper. This property ensures that if an expert wishes to maximize her long run expected weight then she should do this by stating her true beliefs as answer to the seed variables [1].

EXCALIBUR and ANDURIL support four types of DMs. The simplest ones are equal weighting and user-defined weights which fall outside of the performance-based DMs. The *Global Weights* DM is computed as described above while the *Item Weights* DM computes the scores in equation 5 using the information score per item rather than the average information score (equation 3). The difference between DMs will be discussed further in section 4.

Once the different combination schemes have been investigated with Cooke's method it is common practice to perform robustness analysis. This refers to the process of excluding one seed variable or one expert at the time and re-do the analysis with the methods described in this section. EXCALIBUR supports excluding one expert or one item at the time. There is however no reason to think about robustness as a "leave one out at the time" procedure. This has been discussed extensively in the context of out of sample performance of Cooke's method in recent years [4, 3]. ANDURIL supports robusntess analysis leaving k experts or items at the time. This will be further discussed in section 4.

3. Description of ANDURIL

In almost all of the studies, which utilized the Cooke's method, the analysis and synthesis of expert opinions based on experts' performance in judging uncertainty were performed with the free software EXCALIBUR. Hence, the value of EXCALIBUR over the past 25 years is undeniable. However, there are some limitations that stem from the fact that EXCALIBUR is a closed source software. First, EXCALIBUR being a closed source software makes

the understanding of the method more difficult and time consuming to researchers who are recently introduced to the method. Moreover, it is impossible to modify it (for example) in order to expand its features or investigate different approaches for combination of expert judgments. For these reasons, the authors strongly believe that an open source software for Cooke's classical model that is transparent and easily modifiable (such as ANDURIL) will be of benefit for practitioners and researchers.

3.1. Software Architecture

ANDURIL does not have a user interface yet, but there is a main script named `ANDURIL_Main` that can be used by the user to enter the data and run the desired analysis. The supported functionalities of Cooke's classical model by ANDURIL which can be accessed by `ANDURIL_Main` as well as the required inputs of this script are presented below.

`ANDURIL_Main`

Description: This is the main script that can be used to apply Cooke's classical model to analyze and synthesize expert judgments by using ANDURIL. ANDURIL supports the following features:

1. Calculation of DM using global weights
2. Calculation of DM using item weights
3. Calculation of DM using equal or user defined weights
4. Optimization of DM
5. Robustness check itemwise
6. Robustness check expertwise
7. Plotting assessments itemwise
8. Plotting robustness results

Input(s): The inputs that are required in order to do the analysis of expert judgments and combine their opinions are the following:

- `Cal_var`: a three-dimensional array that contains the assessments of three quantiles (in columns) of every expert (in rows) concerning every *seed item* (in the third dimension of the array). For clarification, see Figure 2.

- `TQs`: a three-dimensional array that contains the assessments of three quantiles (in columns) of every expert (in rows) concerning every *target item* (in the third dimension of the array). Similarly to Figure 2.

- `realization`: a cell array with as many entries as the realizations of the seed items and as many empty cells as the target questions. Please note that the order of this should be the same as the order of the items; the first entry should be the realization of the first calibration variable.

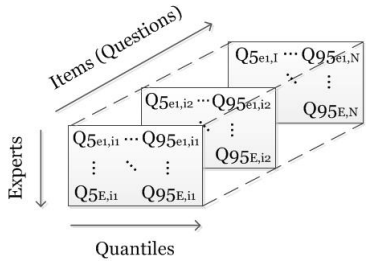


Figure 2: Structure of the three-dimensional arrays for `Cal_var` and `TQs`

- `back_measure`: a cell array that show the background measure (either `uni` or `log_uni`) of every item.
- `weight_type`: a variable that indicates which weighting scheme should be used to obtain the distributions of DM. It is possible to choose between `'equal'`, `'global'` and `'item'`. See section 2 for a description of different weighting schemes.
- `alpha`: significance level for the indicator function in eq. 5. It should be noted that this value cannot be larger than the highest calibration score observed in the pool of experts, because that would result in zero weight for every expert.
- `k`: overshoot for the intrinsic range (see section 2.2). Typically equal to 10%.

3.2. Components of ANDURIL

In this section an overview of the functions which constitute ANDURIL is presented. These functions were grouped in the following subsections according to their purpose.

3.2.1. Import values

ANDURIL supports two methods of importing data in the format required for the analysis. The function `formulate_data` can be used in combination with import data interface of MATLAB for values which were exported directly from EXCALIBUR or values which were saved in spreadsheets. The function `import_ascii_files` can be used for importing data from the .dtt and .rls EXCALIBUR files.

`formulate_data`

Syntax: `[Cal_var, TQs] = formulate_data(var_excalib, realization, N_ex, N_seed, N_tqs)`

Description: A function that formulates the data which were exported from EXCALIBUR (or saved in a spreadsheet) in the appropriate format to perform the analysis with ANDURIL.

Input(s):

- A matrix (`var_excal`) that contains the assessments of every expert concerning all the items, in EXCALIBUR format. The user may export the assessments of every expert regarding every item from EXCALIBUR and import these in MATLAB.
- A cell array (`realization`) that contains the realization of every seed question and as many empty cells ([ ]) as target variables. This cell array should be created by the user.
- The number of experts `N_ex` which participated in the study.
- The number of seed items `N_seed`.
- The number of target items `N_tqs`.

Output(s):

- A three-dimensional array (`Cal_var`) that contains the assessments of the experts for every seed item.
- A three-dimensional array (`TQs`) that contains the assessments of the experts for every target variable.

`import_ascii_files`

Syntax: `[Cal_var, TQs, realization, back_measure] = import_ascii_files(filename_quant, filename_real)`

Description: Another function to import data from EXCALIBUR is available. This was developed to allow for reading the ascii input files from EXCALIBUR so that older files can be easily imported into ANDURIL. *Please note* that in order for this function to work, the descriptions for every item should be erased from the .dtt and .rls files and these should be saved as .txt files. Notepad++ is one of the tools that can be used to create the required .txt files (without the descriptions of the items) easily.

Input(s):

- A string `filename_quant` with the name of the .txt file (or its path if the file is in a different folder) that contains the assessments of the experts. This is the .txt file (without the descriptions of the items) that was created from the .dtt file.

- A string `filename_real` with the name of the txt file (or its path) that contains the assessments of the experts. This is the txt file (without the descriptions of the items) that was created from the .rls file.

Output(s):

- A three-dimensional array (`Cal_var`) that contains the assessments of the experts for every seed item.
- A three-dimensional array (`TQs`) that contains the assessments of the experts for every target variable.
- A cell array (`realization`) that contains the realization of every seed question and as many empty cells ([ ]) as target variables.
- A cell array (`back_measure`) that contains the background measure of every item.

3.2.2. Analysis and synthesis of judgments

calscore

Syntax: `CS = calscore(M, cal_power)`

Description: This function calculates the statistical accuracy (or calibration score) of expert e over the set of seed items.

Input(s):

- An $E \times B$ matrix `M` that contains the number of the realizations captured in every bin that is formed by the quantiles (see section 2.1) provided by every expert e. Where E is the number of the experts and B the number of the bins formed by the provided quantiles.
- A scalar `cal_power` with the power of the calibration test. The power of the calibration test is defined as the ratio N'/N, where $N' < N$. This ratio subsitutes the number of seed questions N in eq. 2. Therefore, the default value of the calibration power is equal to 1. However, the user is able to choose a different value between [0.1, 1] to investigate the influence on the calibration score.

Note: If this function is used to compute the calibration score of a DM that was obtained from a case where only one expert had a non-zero weight and one of the quantiles is exactly equal to the realization, attention should be paid to the calculation of matrix `M`. Due to precision of the calculating engine, it might occur that the resulting quantile from integrating the density has a minor difference with the initial assessment that will result in different

elements of matrix `M` and subsequently a different calibration score. To solve this, the user could use `digits` or `roundn` MATLAB functions to set the precision of the obtained quantiles such that it is relevant for the values of the variables under consideration.

Output(s): A scalar `CS` with the statistical accuracy (or calibration score) of expert e over the set of seed items.

`calculate_information`

Syntax: `[Info_score_real, Info_score_tot] = calculate_information(Cal_var, TQs, realization, k, back_measure)`

Description: This function calculates the relative information (or information score) of expert e over the set of seed items.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.
- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable.
- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables.
- `k` overshoot.
- A cell array `back_measure` with the background measure of every item.

Output(s):

- Information score over the seed variables `Info_score_real`
- Information score over all items (i.e. seed variables and target variables) `Info_score_tot`

`calculate_information_seed`

Syntax: `Info_score_real = calculate_information_seed(Cal_var, TQs, realization, k, back_measure)`

Description: This function is a modified version of `calculate_information function`, with the purpose to be used within the `DM_optimization` function. This function has the same inputs as the `calculate_information` function and it calculates only the required information score of every expert over the seed items `Info_score_real`, in order to reduce computation time.

global_weights

Syntax: `W = global_weights(Cal_var, TQs, realization, alpha, background_measure, k)`

Description: The function `global_weights` calculates the calibration score, the information score over the seed items and subsequently the weight of every expert e.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.
- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable.
- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables.
- Significance level `alpha`.
- A cell array `back_measure` with the background measure of every item.
- `k` overshoot.

Output(s): A table `W` with the calibration score (first column) the information score over all the items (second column), the information score over the seed items (third column), un-normalized weight (fourth column), normalized weight (fifth column) for every expert (in a different row of this table).

global_weights_for_opt

Syntax: `W = global_weights_for_opt(Cal_var, realization, alpha, background_measure, k)`

Description: This function is a modified version of global_weights function, with the purpose to be used in the optimization function. This function calculates the calibration score, as well as the global weight of every expert by using the `calculate_information_seed` function to compute only the required information score of every expert over the seed items, in order to reduce computation time. This function has the same inputs as the `calculate_information`, excluding the three dimensional matrix with the assessments regarding the target variables.

`calculate_DM_global`

Syntax: `[f_DM, F_DM, X, DM, W_incl_VE] = calculate_DM_global(Cal_var, TQs, realization, w, k, back_measure, alpha)`

Description: This function calculates the distribution of the DM for every item, using the global weights or equal weights weighting schemes.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.
- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable.
- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables.
- A row vector `w` with the normalized weights of every expert. In case that global weight are used for calculating the DM, this vector is the transposed 5^{th} column of table `W` that was produced from `global_weights` function. If equal weights are used to calculate the DM, then a row vector with equal weights for every expert should be provided.
- `k` overshoot.
- A cell array `back_measure` with the background measure of every item.
- Significance level `alpha`. It should be noted that this variable must have the same value as the `alpha` that was used as input to `global_weights` function.

Output(s):

- A cell array `f_DM` that contains the density of the DM for values `X`.
- A cell array `F_DM` that contains the cumulative probability of the DM for values `X` of every item.
- A cell array `X` that contains all the unique values provided by the experts with non-zero weights for every item.
- A matrix `DM` with the quantiles of the obtained DM. This matrix has the q_{li}, 5%, 50%, 95% and q_{hi} quantiles of the DMs distribution for every item i.

- The table `W_incl_VE`. This is actually the table `W` updated with the obtained DM (in the last row).

`item_weights`

Syntax: `[unorm_w, W_itm, W_itm_tq] = item_weights(Cal_var, TQs, realization, alpha, back_measure, k)`

Description: This function calculates the item weights of every expert e for every item. The main difference with the global weights weighting scheme is that the weights are different for every item. In this way the opinion of every expert has a different weight for every item. This is achieved by using the relative information of every particular item.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.
- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable.
- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables.
- Significance level `alpha`.
- A cell array `back_measure` with the background measure of every item.
- `k` overshoot.

Output(s):

- Unormalized weights `unorm_w`. This $E \times N$ matrix contains the weights of every expert e for every seed item i, where $e = 1, ..., E$ and $i = 1, ..., N$.
- A $E \times N$ matrix `W_item` with the normalized weights of every expert e for every seed item.
- A $E \times N_{tq}$ matrix `W_item_tq` with the normalized weights of every expert e for every target item i_{tq}, where $e = 1, ..., E$ and $i_{tq} = 1, ..., N_{tq}$.

`calculate_DM_item`

Syntax: `[f_DM, F_DM, X, DM, W_incl_VE] = calculate_DM_item(Cal_var, TQs, realization, W_itm, W_itm_tq, k, back_measure, alpha)`

Description: This function calculates the distribution of the DM for every item using the item weights weighting scheme.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.
- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable.
- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables.
- A $E \times N$ matrix `W_itm` with the normalized weights of every expert e for every seed item i.
- A $E \times N_{tq}$ matrix `W_itm` with the normalized weights of every expert e for every target item i_{tq}.
- `k` overshoot.
- A cell array `back_measure` with the background measure of every item.
- Significance level `alpha`.

Output(s):

- A cell array `f_DM` that contains the density of the DM for values `X`.
- A cell array `F_DM` that contains the cumulative probability of the DM for values `X` of every item.
- A cell array `X` that contains all the unique values provided by the experts with non-zero weights for every item.
- A matrix `DM` with the quantiles of the obtained DM. This matrix has the q_{li}, 5%, 50%, 95% and q_{hi} quantiles of the DM's distribution for every item.
- A table `W_incl_VE` that contains the global weights for all the experts including the item weights DM. This is actually the table `W` updated with the obtained DM (in the last row).

DM_optimization

Syntax: `[F_DM_opt, X_DM_opt, DM_opt, W_opt, W_withDM, new_alpha] = DM_optimization( Cal_var, TQs, realization, k, back_measure, weight_type)`

Description: This function calculates the distribution of the DM for every item using the significance level `alpha` that optimizes the DM in terms of statistical accuracy.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.
- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable
- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables
- `k` overshoot.
- A cell array `back_measure` with the background measure of every item
- A string `weight_type` that indicates which weighting scheme should be used to obtain the distributions of the DM. This can be either 'global' or 'item'.

Output(s):

- A cell array `F_DM_opt` that contains the cumulative probability of the optimized DM for values `X` of every item.
- A cell array `X` that contains all the unique values provided by the experts with non-zero weights for every item.
- A matrix `DM` that contains the quantiles of the obtained optimal (in terms of statistical accuracy) DM. This matrix has the q_{li}, 5%, 50%, 95% and q_{hi} quantiles of the optimal DM's distribution for every item i.
- An $E \times 5$ matrix `W_opt` containing the calibration score (1^{st} column), the information score over all the items (2^{nd} column), the information score over the seed items (3^{rd} column), the un-normalized weight (4^{th} column) and normalized weight (5^{th} column) for every expert e.

- A matrix `W_withDM` that is an updated version of `W_opt` containing the scores and weights of every expert when the virtual expert (DM) enters the pool of experts. The values of performance measures and the weights which concern the optimized DM are presented in the last row of `W_withDM`.

- The value of significance level `new_alpha` that optimizes the statistical accuracy of the DM.

3.2.3. Post-processing

`Checking_Robustness_items`

Syntax: `Robustness_table = Checking_Robustness_items(Cal_var, TQs, realization, k, alpha, back_measure, N_max_it, weight_type, optimization, incl_cal_power)`

Description: This function calculates the performance measures (calibration score, information score over seed variable and over all variables with respect to the background measure) of the DM that occurs when up to `N_max_it` seed item(s) are excluded at most. It calculates the performance measures for every possible combination, starting from excluding one up to `N_max_it` seed items at a time.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.

- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable

- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables

- `k` overshoot.

- The significance level `alpha`.

- A cell array `back_measure` with the background measure of every item.

- The maximum number of items to be removed `N_max_it` for investigating robustness.

- The weighting scheme `weight_type` that was considered to obtain the distributions of DM. It can be either 'global' or 'item'

- A sting `optimization` that can be either 'yes' or 'no' showing if the DM under investigation is optimized in terms of statistical accuracy or not.

- A string `incl_cal_power` that indicates whether or not the calibration power should be taken into account while computing the calibration score of the "perturbed" DM. If this sting is 'yes', the calibration power is taken into account when the calibration score of the "perturbed" DM is computed. Otherwise, the calibration power remains equal to the initial one, as it was defined by the user in the ANDURIL_Main. The default value of calibration power is equal to one. See also section 4.3

Output(s): `Robustness_table` is a cell array that contains the performance measures for every possible combination, starting from excluding one up to `N_max_it` seed item(s) at a time. The 1×4 cell array of every row presents: the id number of the excluded seed item(s) (1^{st} column), the information score over all items with respect to the background measure for the obtained DM (2^{nd} column), the information score over the seed items with respect to the background measure for the obtained DM (3^{rd} column) and the calibration score of the obtained DM (4^{th} column).

`Checking_Robustness_experts`

Syntax: `Robustness_table_ex = Checking_Robustness_experts(Cal_var, TQs, realization, k, alpha, back_measure, N_max_ex, weight_type, optimization)`

Description: This function is the same as `Checking_Robustness_items` function but it calculates the performance measures of the DM that occurs when up to `N_max_ex` expert(s) are excluded at most. It must be mentioned that incl_cal_power is not included because the number of items N remains the same.

`plotting_itemwise`

Syntax: `plotting_itemwise(Cal_var, TQs, realization, DM_set, DM_str, ystr)`

Description: This function produces as many plots as the total number of items (i.e. seed and target items). Every plot presents the assessments (i.e. 5^{th}, 50^{th},95^{th} percentiles) of every expert e as well as every DM, for every particular item i.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.

- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable.
- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables.
- A three-dimensional array `DM_set` that is structured as the `Cal_var` and `TQs` arrays and contains the quantiles of the DMs that were calculated.
- A string `DM_str` that contains the color and the type of the markers for the DMs (*Example*: `DM_str = 'c-s','g-p','r-o'`)
- A string `ystr` that contains the names of every expert and DMs that will be shown on the y-axis of the plots (*Example*: `ystr2 = {'','Exp. 1','Exp. 2','Exp. 3', 'Exp. 4','Exp. 5', 'DM1-global', 'DM2-item', 'DM3-equal', 'Realization', ''}`)

`robustness_plots`

Syntax: `robustness_plots(Cal_var, Robustness_table, W_incl_DM, N_max_it)`

Description: This function produces three box-plots. Each plot corresponds to one measure of performance in judging uncertainty. Namely statistical accuracy, information score over all items and information score over seed items. Each box-plot presents how the values of every measure vary with the number of excluded items (x-axis). In these plots a horizontal line is also plotted, that shows the values of the DM whose robustness is under investigation. Finally, a magenta marker shows the geometric mean for every number of removed items.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.
- The `Robustness_table` obtained from `Checking_Robustness_items` function.
- The table `W_incl_DM` of the DM whose robustness was investigated using the function `Checking_Robustness_items`.
- The maximum number of excluded seed item(s) that was used to obtain the `Robustness_table`.

alter_calc_DM_global

Syntax: `[f_DM, F_DM, X, DM, W_incl_VE] = alter_calc_DM_global(Cal_var, TQs, realization, w, k, back_measure, alpha, alter_calc)`

Description: This function is a modified version of `calculate_DM_global` function. It was created to investigate the effect of obtaining the distribution of global weight DM for every item when calculating the intrinsic ranges of every item by: i) taking into account the realization and the judgments of experts with non-zero weights and ii) taking into account only the judgments of the experts with non-zero weights.

Input(s):

- A three-dimensional array `Cal_var` that contains the assessments of the experts for every seed item.
- A three-dimensional array `TQs` that contains the assessments of the experts for every target variable.
- A cell array `realization` that contains the realization of every seed question and as many empty cells ([ ]) as target variables.
- A row vector `w` with the normalized weights of every expert. In case that global weight are used for calculating the DM, this vector is the transposed 5^{th} column of table `W` that was produced from `global_weights` function. If equal weights are used to calculate the DM, then a row vector with equal weights for every expert should be provided.
- `k` overshoot.
- A cell array `back_measure` with the background measure of every item.
- Significance level `alpha`. It should be noted that this variable must have the same value as the `alpha` that was used as input to `global_weights` function.
- A string `alter_calc` that indicates how the intrinsic range of each item will be calculated. It can be either '`exp_realz`' that calculates the intrinsic range of every item based on the quantiles of the experts with non-zero weights and the realizations or '`exp_only`' that calculates the intrinsic range of every item based only on the quantiles of experts with non-zero weights.

Output(s):

- The density of the DM `f_DM` for values `X`.

- The cumulative probability of the DM `F_DM` for values `X`.
- The values `X` of the DM for every item.
- A matrix `DM` with the quantiles of the obtained DM. This matrix has the q_{li}, 5%, 50%, 95% and q_{hi} quantiles of the DMs distribution for every item i.
- The table `W_incl_VE`. This is actually the table `W` updated with the obtained DM (in the last row).

4. Illustrative Example for Validation

ANDURIL is a newly developed open source toolbox, therefore it should be validated with EXCALIBUR. For this purpose a recent SEJ study concerning the estimation of GHG emissions in Mexico for 2020 and 2030 [5] was used as a test case (Figure 1). The part of the study that is used to validate ANDURIL is the one concerning the estimation of Gross Domestic Product. In this study 9 experts participated and provided the 5^{th}, 50^{th} and 95^{th} percentiles of their uncertainty distribution regarding 13 seed variables and 6 target variables. In the following subsections (4.1, 4.2 and 4.3), the results obtained from applying ANDURIL to the test case are presented and compared with those obtained from EXCALIBUR. These results can be reproduced by using the `ANDURIL_example` script and the dtt and rls files of EXCALIBUR provided as a supplement.

4.1. Distributions of obtained DMs

Five different DMs were calculated using ANDURIL. DM_1 was calculated using the function `calculate_DM_global`, DM_2 was calculated using the function `calculate_DM_item`, DM_3 was calculated using the function `calculate_DM_global` with equal weights to every expert, DM_4 was calculated using the function `DM_optimization` and DM_5 was calculated using the function `calculate_DM_global` with user-defined weights to every expert. As far as the user-defined weights are concerned, experts 5 and 6 received a weight equal to 0.4 and 0.6 respectively, while the remaing experts were assigned a zero weight. The details of every DM are summarized in Table 1. Also, it should be noted that the background measure for every item is uniform. However, the same DMs were calculated and validated when log-uniform background measure was used for every item (except seed item 3, because the 5% quantile of an expert was negative for this particular item and hence a log-uniform measure cannot be used).

The comparison of the obtained quantiles using ANDURIL and EXCALIBUR is presented in Table 2. As it can be seen, there are very small differences most probably due to differences in precision of the calculating engine. The maximum difference is 0.0005 in absolute value.

Furthermore, Figure 3 and Figure 4 show the comparison of the obtained plots for every individual expert and DMs concerning seed item 5 and target item 1 respectively. The plots of ANDURIL were produced using the function `plotting_itemwise` and show that the same results are obtained with EXACLIBUR.

Name	Type of weights	Optimization	Significance Level (`alpha`)
DM_1	global weights	No	0.05
DM_2	item weights	No	0.05
DM_3	equal weights	-	0.00
DM_4	global weights	Yes	-
DM_5	user weights	-	0.00

Table 1: Overview of details of calculated DMs.

	EXCALIBUR			ANDURIL		
Name	q_5	q_{50}	q_{95}	q_5	q_{50}	q_{95}
DM_1	3.02	5.431	8.000	3.0201	5.4311	8.000
DM_2	3.063	5.327	8.000	3.0633	5.3275	8.000
DM_3	2.297	4.684	7.463	2.2971	4.6840	7.4626
DM_4	3.021	5.44	7.999	3.0209	5.4395	7.9994
DM_5	3.098	6.026	7.928	3.0978	6.0263	7.928

Table 2: Comparison of the four DMs' quantiles regarding seed item 5 using ANDURIL and EXCALIBUR.

4.2. Measures of performance and weights

In this subsection the obtained measures of performance and the weights of every expert (including DMs) are compared with the results from EXCALIBUR. We present in this section results concerning the different Decision Makers. Table 3 (and 4, 5, 6 and 7) presents results obtained with the output `W_withDM` for the global weights DM (item, equal and global optimized respectively for tables 4 to 6) as calculated by ANDURIL. Similarly, Figure 5 and 6, 7, 8 and 9 present equivalent tables obtained from EXCALIBUR.

It can be seen that the resulting measures of performance as well as the weights present again small differences as before due to precision. Also, it

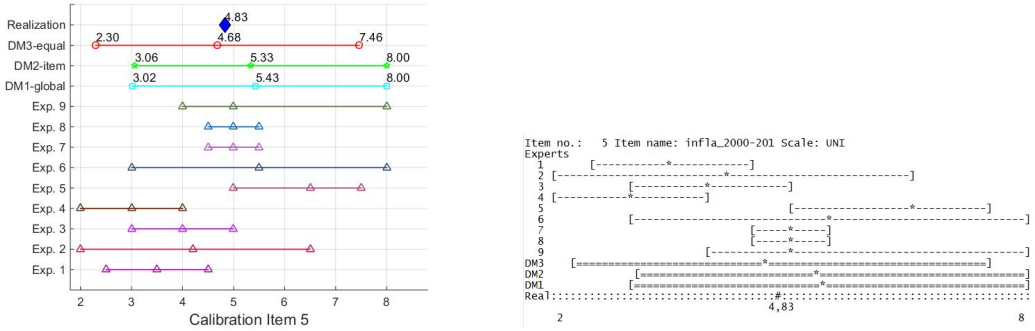


Figure 3: Comparison of obtained plots for the assessments of all experts and DMs concerning seed item 5.

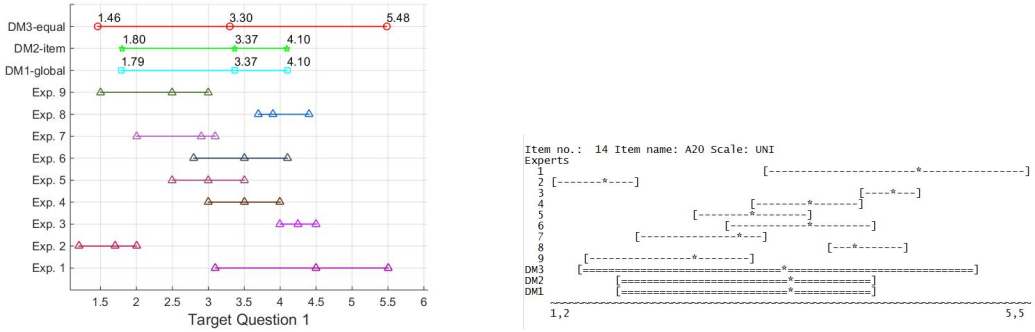


Figure 4: Comparison of obtained plots for the assessments of all experts and DMs concerning seed item 5.

must be mentioned that the column entitled "Normalized weights without DM" that is presented in Figures 5 (and 6 to 8) from EXCALIBUR was also obtained from ANDURIL using the `global_weights` and `item_weight` functions. That is, before the virtual expert entered the pool of experts. The column is however not included in Tables 3 (and 4 to 6) in order to be consistent with the output as provided by ANDURIL's functions.

4.2.1. DM_1: Global weights

The code that was used to calculate DM_1 is the following:

```
W = global_weights(Cal_var, TQs, realization, alpha,
        back_measure, k);

[f_DM1, F_DM_out1, X_out1, DM1, W_incl_DM1] =
calculate_DM_global(Cal_var, TQs, realization,
W(:,5)', k, back_measure, alpha);
```

Expert ID	Calibration Score	Information Score (All items)	Information Score (Seed items)	Un-normalized Weights	Normalized Weights incl. DM
Expert 1	5.03797e-12	1.60819	1.93478	0	0
Expert 2	1.03264e-05	1.03006	1.00562	0	0
Expert 3	9.21494e-05	1.66612	1.65122	0	0
Expert 4	1.03264e-05	1.30231	1.48421	0	0
Expert 5	0.00150	1.22279	1.300709	0	0
Expert 6	0.27665	1.08076	1.28486	0.35546	0.52529
Expert 7	0.00015	1.82512	1.94417	0	0
Expert 8	1.36251e-05	1.69804	1.91298	0	0
Expert 9	0.05306	1.07772	1.28461	0.06816	0.10073
DM_1	0.26503	0.80631	0.95483	0.25306	0.37397

Table 3: Table W including DM_1 from ANDURIL.

Results of scoring experts
Bayesian Updates: no Weights: global DM Optimisation: no
Significance Level: 0,05 Calibration Power: 1

Nr.	Id	Calibr.	Mean relative total	Mean relative realization	Numb real	UnNormalized weight	Normaliz.weig without DM	Normaliz.weig with DM
1	A	5,039E-012	1,608	1,935	13	0	0	0
2	B	1,032E-005	1,03	1,006	13	0	0	0
3	C	9,215E-005	1,666	1,651	13	0	0	0
4	D	1,032E-005	1,302	1,484	13	0	0	0
5	E	0,001505	1,223	1,301	13	0	0	0
6	F	0,2766	1,081	1,285	13	0,3554	0,8391	0,5253
7	G	0,0001545	1,825	1,944	13	0	0	0
8	H	1,363E-005	1,698	1,913	13	0	0	0
9	I	0,05304	1,078	1,285	13	0,06813	0,1609	0,1007
10	DM1	0,265	0,8063	0,9548	13	0,253		0,374

Figure 5: Table from EXCALIBUR including DM_1.

4.2.2. DM_2: Item weights

The code that was used to calculate DM_2 is the following:

```
[unorm_w, W_itm, W_itm_tq] = item_weights(Cal_var, TQs,
 realization, alpha, back_measure);
if isequal(W_itm(:,1),zeros(size(W_itm,1),1))
  error('Significance Level value should be smaller
    than the highest calibtration score')
end
```

```
[f_DM2, F_DM_out2, X_out2, DM2, W_incl_DM2] =
calculate_DM_item(Cal_var, TQs, realization,
 W_itm, W_itm_tq, k, back_measure, alpha);
```

Expert ID	Calibration Score	Information Score (All items)	Information Score (Seed items)	Un-normalized Weights	Normalized Weights incl. DM
Expert 1	5.03797e-12	1.60819	1.93478	0	0
Expert 2	1.03264e-05	1.03006	1.00562	0	0
Expert 3	9.21494e-05	1.66612	1.65122	0	0
Expert 4	1.03264e-05	1.30231	1.48421	0	0
Expert 5	0.00150	1.22279	1.300709	0	0
Expert 6	0.27665	1.08076	1.28486	0.355468	0.37107
Expert 7	0.000154	1.82512	1.94417	0	0
Expert 8	1.36251e-05	1.69805	1.91298	0	0
Expert 9	0.05306	1.07771	1.284617	0.068165	0.071158
DM_2	0.52856	0.84605	1.01086	0.534310	0.557767

Table 4: Table W including DM_2 from ANDURIL.

Results of scoring experts
Bayesian Updates: no Weights: item DM Optimisation: no
Significance Level: 0,05 Calibration Power: 1

Nr.	Id	Calibr.	Mean relative total	Mean relative realization	Numb real	UnNormalized weight	Normaliz.weig without DM	Normaliz.weig with DM
1	A	5,039E-012	1,608	1,935	13	0		0
2	B	1,032E-005	1,03	1,006	13	0		0
3	C	9,215E-005	1,666	1,651	13	0		0
4	D	1,032E-005	1,302	1,484	13	0		0
5	E	0,001505	1,223	1,301	13	0		0
6	F	0,2766	1,081	1,285	13	0,3554		0,3711
7	G	0,0001545	1,825	1,944	13	0		0
8	H	1,363E-005	1,698	1,913	13	0		0
9	I	0,05304	1,078	1,285	13	0,06813		0,07114
10	DM2	0,5285	0,8461	1,011	13	0,5342		0,5578

Figure 6: Table from EXCALIBUR including DM_2

4.2.3. DM_3: Equal weights

The code that was used to calculate DM_3 is the following:

```
for i = 1:size(Cal_var,1)
    eq_w(i) = 1/size(Cal_var,1);
end
alpha_eq = 0;
[f_DM3, F_DM_out3, X_out3, DM3, W_incl_DM3] =
 calculate_DM_global(Cal_var, TQs, realization, eq_w,
 k, back_measure, alpha_eq);
```

Expert ID	Calibration Score	Information Score (All items)	Information Score (Seed items)	Unnormalized Weights	Normalized Weights incl. DM
Expert 1	5.03797e-12	1.60819	1.93478	9.747e-12	2.149e-11
Expert 2	1.03264e-05	1.03006	1.00562	1.0385e-05	2.2896e-05
Expert 3	9.21494e-05	1.66612	1.65122	1.5216e-04	3.3548e-04
Expert 4	1.03264e-05	1.30231	1.48421	1.5327e-05	3.3792e-05
Expert 5	0.00150	1.22279	1.300709	0.0019568	0.004314
Expert 6	0.27665	1.08076	1.28486	0.355468	0.783728
Expert 7	0.000154	1.82512	1.94417	3.0029e-04	6.6207e-04
Expert 8	1.36251e-05	1.69805	1.91298	2.6065e-05	5.7467e-05
Expert 9	0.05306	1.07771	1.284617	0.068165	0.150290
DM_3	0.06317	0.33777	0.434773	0.02746	0.06055

Table 5: Table W including DM_3 from ANDURIL.

4.2.4. DM_4: Global weights optimized

The code that was used to calculate DM_4 is the following:

```
weight_type = 'global';
[F_DM_out4,X_DM_out4,DM4_opt,W_opt,W_withDM,new_alpha]=
 Anduril_DM_Optimization(Cal_var, TQs, realization,
 back_measure, weight_type);
```

Results of scoring experts
Bayesian Updates: no Weights: equal DM Optimisation: no
Significance Level: 0 Calibration Power: 1

Nr.	Id	Calibr.	Mean relative total	Mean relative realization	Numb real	UnNormalized weight	Normaliz.weig without DM	Normaliz.weig with DM
1	A	5,039E-012	1,608	1,935	13	9,75E-012	0,1111	2,15E-011
2	B	1,032E-005	1,03	1,006	13	1,038E-005	0,1111	2,289E-005
3	C	9,215E-005	1,666	1,651	13	0,0001522	0,1111	0,0003356
4	D	1,032E-005	1,302	1,484	13	1,532E-005	0,1111	3,378E-005
5	E	0,001505	1,223	1,301	13	0,001958	0,1111	0,004317
6	F	0,2766	1,081	1,285	13	0,3554	0,1111	0,7837
7	G	0,0001545	1,825	1,944	13	0,0003004	0,1111	0,0006625
8	H	1,363E-005	1,698	1,913	13	2,607E-005	0,1111	5,75E-005
9	I	0,05304	1,078	1,285	13	0,06813	0,1111	0,1503
10	DM3	0,0632	0,3378	0,4348	13	0,02748		0,0606

Figure 7: Table from EXCALIBUR including DM_3

Results of scoring experts
Bayesian Updates: no Weights: global DM Optimisation: yes
Significance Level: 0,001505 Calibration Power: 1

Nr.	Id	Calibr.	Mean relative total	Mean relative realization	Numb real	UnNormalized weight	Normaliz.weig without DM	Normaliz.weig with DM
1	A	5,039E-012	1,608	1,935	13	0	0	0
2	B	1,032E-005	1,03	1,006	13	0	0	0
3	C	9,215E-005	1,666	1,651	13	0	0	0
4	D	1,032E-005	1,302	1,484	13	0	0	0
5	E	0,001505	1,223	1,301	13	0,001958	0,004601	0,001972
6	F	0,2766	1,081	1,285	13	0,3554	0,8353	0,358
7	G	0,0001545	1,825	1,944	13	0	0	0
8	H	1,363E-005	1,698	1,913	13	0	0	0
9	I	0,05304	1,078	1,285	13	0,06813	0,1601	0,06863
10	DM4	0,614	0,7797	0,924	13	0,5673		0,5714

Figure 8: Table from EXCALIBUR including DM_4

4.2.5. DM_5: User-defined weights

The code that was used to calculate DM_5 is the following:

```
user_w = [0 0 0 0 0.4 0.6 0 0 0];
if sum(user_w) not equal to 1
  error('User defined weights should add up to 1')
end
alpha_ud = 0;
[f_DM_user,F_DM_user,X_DM_user,DM_user,W_incl_DM_user]=
calculate_DM_global(Cal_var, TQs, realization, user_w,
k, back_measure, alpha_ud);
```

4.3. Robustness analysis

This subsection presents the tables for robustness analysis concerning DM_1 (i.e. DM with global weights, 0.05 significance level and uniform background measure for every item), when one seed item is excluded at a time. The obtained table for robustness using the `Checking_Robustness_items` function of ANDURIL is presented in Table 8. The robustness analysis table from EXCALIBUR is presented in Figure 10. It can be seen that the results are almost identical with the maximum absolute difference being equal to

Expert ID	Calibration Score	Information Score (All items)	Information Score (Seed items)	Un-normalized Weights	Normalized Weights incl. DM
Expert 1	5.03797e-12	1.60819	1.93478	0	0
Expert 2	1.03264e-05	1.03006	1.00562	0	0
Expert 3	9.21494e-05	1.66612	1.65122	0	0
Expert 4	1.03264e-05	1.30231	1.48421	0	0
Expert 5	0.00150	1.22279	1.300709	0.0019568	0.0019706
Expert 6	0.27665	1.08076	1.28486	0.355468	0.357981
Expert 7	0.000154	1.82512	1.94417	0	0
Expert 8	1.36251e-05	1.69805	1.91298	0	0
Expert 9	0.05306	1.07771	1.284617	0.068165	0.0686476
DM_4	0.6140	0.77970	0.92401	0.56738	0.571399

Table 6: Table W including DM_4 from ANDURIL.

Results of scoring experts

Bayesian Updates: no Weights: user DM Optimisation: no

Significance Level: 0 Calibration Power: 1

Nr.	Id	Calibr.	Mean relative total	Mean relative realization	Numb real	UnNormalized weight	Normaliz.weigl without DM	Normaliz.weig with DM
1	A	5,039E-012	1,608	1,935	13	9,75E-012	0	9,451E-012
2	B	1,032E-005	1,03	1,006	13	1,038E-005	0	1,006E-005
3	C	9,215E-005	1,666	1,651	13	0,0001522	0	0,0001475
4	D	1,032E-005	1,302	1,484	13	1,532E-005	0	1,485E-005
5	E	0,001505	1,223	1,301	13	0,001958	0,4	0,001898
6	F	0,2766	1,081	1,285	13	0,3554	0,6	0,3445
7	G	0,0001545	1,825	1,944	13	0,0003004	0	0,0002912
8	H	1,363E-005	1,698	1,913	13	2,607E-005	0	2,528E-005
9	I	0,05304	1,078	1,285	13	0,06813	0	0,06605
10	DM5	0,6894	0,747	0,8785	13	0,6057		0,5871

Figure 9: Table from EXCALIBUR including DM_5

0.0004, concerning the Information Score over all items with respect to the background measure. Notice however that if the calibration power is taken into account when calculating the calibration score of the "perturbed" DM when 1 item is excluded at a time the results of ANDURIL are different than those of EXCALIBUR. In other words, EXCALIBUR does not consider the calibration power in performing robustness analysis. For these reasons, it was decided to give the option to the user to decide if he/she wants to take

Expert ID	Calibration Score	Information Score (All items)	Information Score (Seed items)	Unnormalized Weights	Normalized Weights incl. DM
Expert 1	5.03797e-12	1.60819	1.93478	9.747e-12	9.4478e-12
Expert 2	1.03264e-05	1.03006	1.00562	1.0385e-05	1.0065e-05
Expert 3	9.21494e-05	1.66612	1.65122	1.5216e-04	1.4748e-04
Expert 4	1.03264e-05	1.30231	1.48421	1.5327e-05	1.4856e-05
Expert 5	0.00150	1.22279	1.300709	0.0019568	0.001896
Expert 6	0.27665	1.08076	1.28486	0.355468	0.344542
Expert 7	0.000154	1.82512	1.94417	3.0029e-04	2.9106e-04
Expert 8	1.36251e-05	1.69805	1.91298	2.6065e-05	2.5263e-05
Expert 9	0.05306	1.07771	1.284617	0.068165	0.06607
DM_5	0.6894	0.74695	0.8785	0.6056	0.5870

Table 7: Table W including DM_5 from ANDURIL.

into account a different calibration power (equal to N'/N) when robustness of items is investigated.

Excluded item	Information Score (All items)	Information Score (Seed items)	Calibration Score
Excluded seed item 1	0.8057	0.9936	0.2982
Excluded seed item 2	0.8377	1.0135	0.2982
Excluded seed item 3	0.8112	0.9913	0.3509
Excluded seed item 4	0.8459	1.0257	0.2982
Excluded seed item 5	1.1334	1.3808	0.3111
Excluded seed item 6	0.8261	0.9963	0.1776
Excluded seed item 7	1.0746	1.2926	0.3111
Excluded seed item 8	0.6480	0.7418	0.5213
Excluded seed item 9	0.7183	0.8345	0.2982
Excluded seed item 10	1.0138	1.2014	0.1776
Excluded seed item 11	0.7362	0.8515	0.5419
Excluded seed item 12	0.8120	0.9932	0.3509
Excluded seed item 13	0.8090	0.9938	0.3509

Table 8: Robustness table from ANDURIL regarding DM_1 excluding one item at a time.

In the next section we present some aspects that we believe ANDURIL has improved over EXCALIBUR and some limitations of our toolbox.

Robustness analysis on seed Items
Bayesian Updates: no Weights: global DM Optimisation:
Significance Level: 0.0500 Calibration Power: 1.0000

Nr.	Id of excl. item	Rel.info/bgr. total	Rel.info/bgr. realization	Calibr.
1	inter_crecnega	0,8057	0,9937	0,2983
2	inter_crec1996	0,8377	1,014	0,2983
3	infla_crecneg8	0,8112	0,9913	0,351
4	infla_1996-201	0,8459	1,026	0,2983
5	infla_2000-201	1,133	1,381	0,311
6	inflación_8189	0,8261	0,9963	0,1776
7	PEAdes_14mas	1,075	1,293	0,311
8	desocuppromedi	0,648	0,7418	0,5215
9	desocup_millon	0,7183	0,8345	0,2983
10	desocup_prome	1,014	1,201	0,1776
11	razon_desocup	0,7362	0,8515	0,5418
12	crecimientoUS	0,812	0,9932	0,351
13	sincroni_USMEX	0,8091	0,9938	0,351
14	None	0,8063	0,9548	0,265

Figure 10: Robustness table from EXCALIBUR regarding DM_1.

5. Improvements and limitations

5.1. Improvements using ANDURIL

As it was mentioned before, the value of EXCALIBUR software is undeniable. However, the fact that EXCALIBUR is a closed source software causes some limitations for researchers and practitioners of Cooke's classical model. These limitations can be investigated by using ANDURIL. In this subsection, it is illustrated how limitations regarding *intrinsic range*, *item weights*, *distributions of DMs* and *robustness* can be overcome.

Intrinsic Range. The bounds of the intrinsic range for every item i (i.e. q_{li} and q_{hi} introduced in section 2.2) are calculated by considering the assessments of every expert, even the ones with zero weights. Moreover, the intrinsic range for a calibration item takes into consideration the realization of the seed variable. One could argue that for the calculation of the DM's distribution only the assessments of the experts with non-zero weights could be used. This is not possible to be investigated using EXCALIBUR.

For this reason, the `calculate_DM_global` function of ANDURIL was modified in order to investigate the effect of calculating the intrinsic ranges of every item by: i) taking into account the realization and the judgments of only those experts with non-zero weights (that produces `DM1_alt1`) and ii) taking into account only the judgments of the experts with non-zero weights (that produces `DM1_alt2`). This new function was named `alter_calc_DM_global`. It should be noted that in order to investigate the effect of this alternative calculation on the DM_2, the `calculate_DM_item` should be modified similarly. The code that was used to calculate `DM1_alt1` and `W_incl_DM1_alt1` is as follows:

```
W = global_weights(Cal_var,TQs,realization,alpha,
   back_measure, k);
[f_DM1_alt1,F_DM1_alt1,X_alt1,DM1_alt1,W_incl_DM1_alt1]=
 alter_calc_DM_global(Cal_var,TQs,realization,W(:,5)',
 k,back_measure,alpha,'exp_realz');
```

The code that was used to calculate `DM1_alt2` and `W_incl_DM1_alt2` is as follows:

```
W = global_weights(Cal_var, TQs, realization, alpha,
 back_measure, k);
[f_DM1_alt2,F_DM1_alt2,X_alt2,DM1_alt2,W_incl_DM1_alt2]=
 alter_calc_DM_global(Cal_var, TQs, realization, W(:,5)',
  k,back_measure, alpha, 'exp_only');
```

Tables 10, 11 and 12 present the quantiles of DM_1, `DM1_alt1` and `DM1_alt2` respectively. Some differences can be observed, especially (as expected) in quantiles q_h and q_l of every item. Particularly, the maximum absolute difference between DM_1 and `DM1_alt2` concerns the q_h quantile of seed item 8. One may investigate whether these small differences between DM_1 and `DM1_alt2` (or `DM1_alt1`) concerning q_5, q_{50} and q_{95} quantiles would result or not in differences in the measures of performance of the DMs. To investigate this, in Table 9 the measures of performance in judging uncertainty are presented for each DM. Some expected small differences can be observed in the information scores, because the intrinsic range of every item reduces when the quantiles of the experts with zero weights are not taken into account. However, a large absolute difference (equal to 0.189) was observed when comparing the calibration score of DM_1 with that of `DM1_alt1` or `DM1_alt2`. The reason of this 71.3% increase in calibration score, is that the changes in Q5 of `DM1_alt1` and `DM1_alt2` regarding seed item 10 caused the realization to fall into the first interquantile range. The calibration score in equation 2 is a fast function. Small changes in the model may lead to changes in orders of magnitude of the score. Especially when the number of seed variables is low as is usually the case in applications. It should be mentioned that such large differences in values for the intrinsic range may not be always observed in different applications. Nor the consequences of choices for intrinsic ranges in performance measures should necessarily follow the same pattern as in our presentation. This issue has not been discussed in literature for example in those related to out of sample performance of Cooke's model [4, 3]. This is a subject that could be further explored with the aid of ANDURIL.

	Calibration Score	Information Score (All items)	Information Score(Seed items)	Un-normalized Weights
DM_1	0.2650	0.8063	0.9548	0.2531
`DM1_alt1`	0.4540	0.8366	0.9920	0.4504
`DM1_alt2`	0.4540	0.8413	0.9988	0.4535

Table 9: Measures of performance of DMs.

	q_l	q_5	q_{50}	q_{95}	q_h
Seed Item 1	-4.7	3.35	27.46	59.67	87.7
Seed Item 2	-1.15	16.40	36.06	49.83	54.65
Seed Item 3	-10.1	25.89	52.10	89.44	99.1
Seed Item 4	0.2	4.33	10.84	19.86	21.8
Seed Item 5	1.4	3.02	5.43	8.00	8.6
Seed Item 6	1.00	50.00	74.46	99.69	109
Seed Item 7	1.2	4.00	5.35	6.00	10.8
Seed Item 8	-5.957	5.01	5.95	6.99	103.927
Seed Item 9	-1.92	1.80	2.44	3.50	30.72
Seed Item 10	2.1	5.17	6.03	8.60	12.9
Seed Item 11	0.03	0.95	1.35	1.50	9.27
Seed Item 12	1.49	2.51	3.74	5.19	6.41
Seed Item 13	0.344	0.48	0.88	0.95	1.016
Target Item 1	0.77	1.79	3.37	4.10	5.93
Target Item 2	0.675	1.23	2.44	3.20	4.575
Target Item 3	1.6	3.90	4.58	6.00	6.4
Target Item 4	0.86	3.00	3.88	4.50	4.94
Target Item 5	0.67	1.60	2.79	3.70	4.63
Target Item 6	1.17	3.14	4.85	5.90	6.33

Table 10: Quantiles of DM_1.

	q_l	q_5	q_{50}	q_{95}	q_h
Seed Item 1	-2.7	3.39	27.46	59.47	65.7
Seed Item 2	12.6	16.95	36.05	49.82	53.4
Seed Item 3	18.5	27.15	52.10	89.40	96.5
Seed Item 4	2.4	4.49	10.84	19.86	21.6
Seed Item 5	2.5	3.03	5.43	8.00	8.5
Seed Item 6	45	50.00	74.46	99.65	105
Seed Item 7	3.8	4.00	5.35	6.00	6.2
Seed Item 8	-3.977	5.01	5.95	6.99	103.747
Seed Item 9	1.63	1.80	2.44	3.48	3.67
Seed Item 10	4.6	5.36	6.03	8.27	9.4
Seed Item 11	0.84	1.01	1.35	1.45	1.56
Seed Item 12	2.2	2.52	3.74	5.10	5.8
Seed Item 13	0.345	0.48	0.88	0.95	1.005
Target Item 1	1.24	1.85	3.37	4.10	4.36
Target Item 2	0.78	1.25	2.44	3.20	3.42
Target Item 3	3.69	3.90	4.58	6.20	6.21
Target Item 4	2.85	3.00	3.88	4.50	4.65
Target Item 5	1.28	1.66	2.79	3.70	3.92
Target Item 6	2.71	3.27	4.85	5.90	6.19

Table 11: Quantiles of `DM1_alt1`.

	q_l	q_5	q_{50}	q_{95}	q_h
Seed Item 1	-2.7	3.39	27.46	59.47	65.7
Seed Item 2	12.6	16.95	36.05	49.82	53.4
Seed Item 3	18.5	27.15	52.10	89.40	96.5
Seed Item 4	2.4	4.49	10.84	19.86	21.6
Seed Item 5	2.5	3.03	5.43	8.00	8.5
Seed Item 6	45	50.00	74.46	99.65	105
Seed Item 7	3.8	4.00	5.35	6.00	6.2
Seed Item 8	4.8	5.08	5.95	6.88	7.2
Seed Item 9	1.63	1.80	2.44	3.48	3.67
Seed Item 10	4.6	5.36	6.03	8.27	9.4
Seed Item 11	0.84	1.01	1.35	1.45	1.56
Seed Item 12	2.2	2.52	3.74	5.10	5.8
Seed Item 13	0.345	0.48	0.88	0.95	1.005
Target Item 1	1.24	1.85	3.37	4.10	4.36
Target Item 2	0.78	1.25	2.44	3.20	3.42
Target Item 3	3.69	3.90	4.58	6.20	6.21
Target Item 4	2.85	3.00	3.88	4.50	4.65
Target Item 5	1.28	1.66	2.79	3.70	3.92
Target Item 6	2.71	3.27	4.85	5.90	6.19

Table 12: Quantiles of `DM1_alt2`.

Item Weights. When the *item weights* weighting scheme is used to combine the expert judgments, the information score of the obtained DM and the weight that is presented in the output table from EXCALIBUR are calculated using global weights [1]. For illustration, see Figure 6. Therefore, it is not possible for the user to know the exact weights that were used per item.

On the other hand, the `item_weights` function of ANDURIL provides the user with tables `W_itm` and `W_itm_tq` which contain the weights of each expert concerning the seed variables and target variables respectively. The code that was used is as follows:

```
[unorm_w, W_itm, W_itm_tq] = item_weights(Cal_var, TQs,
realization ,alpha, back_measure, k);
[f_DM2, F_DM_out2, X_out2, DM2, W_incl_DM2] =
 calculate_DM_item(Cal_var, TQs, realization, W_itm,...
 W_itm_tq, k, back_measure, alpha);
```

The normalized weights `W_itm` for every expert per seed item (which were used to obtain DM_2) are presented in Table 13. The experts with statistical accuracy below the significance level `alpha` will have a weight equal to zero.

	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6	It. 7	It. 8	It. 9	It. 10	It. 11	It. 12	It. 13
Exp. 1	0	0	0	0	0	0	0	0	0	0	0	0	0
Exp. 2	0	0	0	0	0	0	0	0	0	0	0	0	0
Exp. 3	0	0	0	0	0	0	0	0	0	0	0	0	0
Exp. 4	0	0	0	0	0	0	0	0	0	0	0	0	0
Exp. 5	0	0	0	0	0	0	0	0	0	0	0	0	0
Exp. 6	0.450	0.661	0.698	0.523	0.613	0.707	0.846	0.844	0.808	0.951	0.898	0.886	0.959
Exp. 7	0	0	0	0	0	0	0	0	0	0	0	0	0
Exp. 8	0	0	0	0	0	0	0	0	0	0	0	0	0
Exp. 9	0.550	0.339	0.302	0.477	0.387	0.293	0.154	0.156	0.192	0.049	0.102	0.114	0.041

Table 13: Table with weights of every expert per item regarding DM_2.

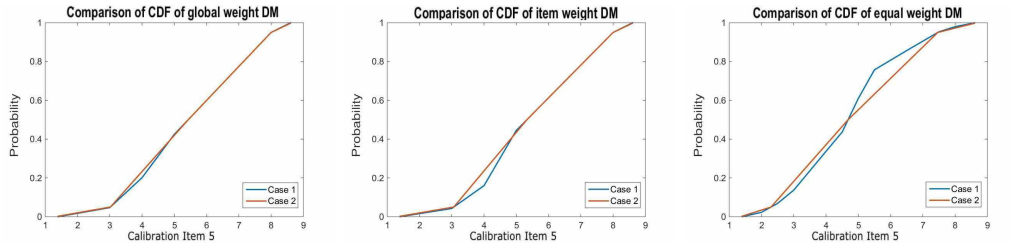
The experts with statistical accuracy above the significance level will have an un-normalized weight equal to the product of the statistical accuracy and the information score of each variable. In this test case, it can be seen that although only experts 6 and 9 have non-zero weights, the weights of these two experts differ significantly from item to item (e.g. item 1 and item 13). This type of information could be valuable to the analyst, in order to visualize the impact of informativeness of every expert on the weight per item.

Distributions of DMs. The cumulative distribution of a DM is calculated by integrating the density of the DM (equation 4). To achieve this, all the values of the quantiles of the experts with non-zero weights are taken into account and the cumulative probability of every unique value is computed. Hence, the $q_{i,5}$, $q_{i,50}$ and $q_{i,95}$ quantiles of the DM are obtained. In EXCALIBUR the output distributions of the DMs are calculated by linear interpolation between these three quantiles (i.e. $q_{i,5}$, $q_{i,50}$ and $q_{i,95}$) of the DM. This may lead to differences between the distributions obtained by integration (Case 1 in figure 11) and the distributions that are obtained by interpolating in between quantiles (Case 2 in the same figure). Functions `calculate_DM_global` and `calculate_DM_item` of ANDURIL provide the user with the DM distributions containing the quantiles of experts with non-zero weights. After using these functions for each DM, the code presented below can be used to plot and compare the distributions of DM_1 regarding seed item 5.

```
% example concerning global weights DM, DM1
figure()
% by integrating the density of the DM
plot(X_out1{5,1}, F_DM_out1{5,1})
hold on
```

```
% by linear interpolation
plot(DM1(5,:),[0.0025 0.05 0.5 0.95 0.9975])
```

Figures 11a, 11b and 11c present the two different distributions of DMs concerning seed item 5, combined with global, item and equal weights weighting schemes respectively. From these plots, it can be seen that interpolating linearly between $q_{i,5}$, $q_{i,50}$ and $q_{i,95}$ to obtain a distribution for the DM may cause significant variations in the resulting distributions, especially when the equal weight combination is considered. The integrated cumulative distribution contains more linear components since every percentile provided by every expert is considered in the density.



(a) Global weight DM (b) Item weight DM (c) Equal weight DM

Figure 11: Comparison of output cumulative distributions obtained by integration (Case 1) and interpolation (Case 2) concerning (a) global weights, (b) item weights and (c) equal weights.

Robustness itemwise. When investigating the robustness of the obtained DM, EXCALIBUR supports the exclusion of only one item at a time for recalculation. Hence, it is not possible to investigate how the performance measures (i.e. Statistical accuracy and Information scores) vary as more than one item are excluded at a time. For this reason, `Checking_Robustness_items` and `robustness_plots` functions of ANDURIL were developed. The latter produces three box-plots. Each plot corresponds to one measure of performance in judging uncertainty. Namely statistical accuracy, information score over all items and information score over seed items. Examples for our demonstration case are presented in Figures 12, 13 and 14 for statistical accuracy, information score (over all items) and information score (over seed items) respectively.

Each box-plot presents how the values of every measure vary with the number of excluded items (horizontal axis). In these plots a green horizontal line that shows the values of the initial DM whose robustness is under investigation. A magenta marker shows the geometric mean for every number of removed items.

It should be noted that when the number of excluded seed items increases there is the possibility that for some combinations (of excluded seed items) the calibration score of all experts reduces below the significance level alpha, resulting in zero weights for every expert. Hence, these combinations are not considered.

As it can be seen in Figures 12, 13 and 14 although the interval containing 95% of the recalculated scores increases as more items are removed at a time, the median remains close to the original value (shown by the green horizontal line) for every measure of performance.

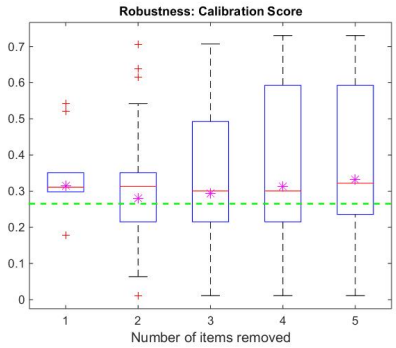


Figure 12: Robustness of calibration score with respect to the number of excluded seed items.

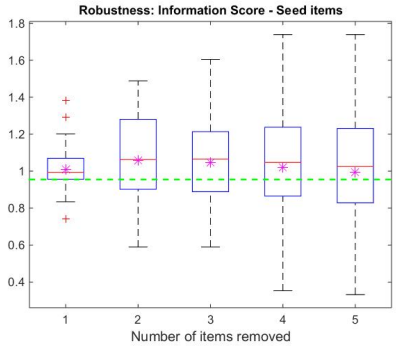


Figure 13: Robustness of information score over the seed items with respect to the number of excluded seed items

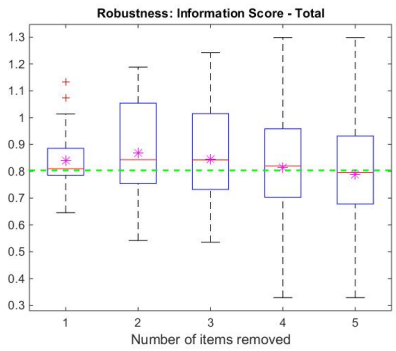


Figure 14: Robustness of information score over all items with respect to the number of excluded seed items

5.2. Limitations of ANDURIL

ANDURIL should not be seen as a replacement of EXCALIBUR. This is a first step towards a toolbox that will facilitate practitioners and researchers working further with Cooke's classical model. ANDURIL does not include yet all the features that EXCALIBUR does. In particular, ANDURIL does not support the following:

- Using more than three quantiles or different ones than the 5%, 50% and 95% quantiles. In the majority of the studies that have been conducted in the past three quantiles (5^{th}, 50^{th} and 95^{th}) have been used. However, supporting more than these three quantiles, would be a valuable addition to the tool.

- Missing values for certain items. It may occur that some items are not assessed by a particular expert(s). In this case, at the moment, this particular item should be excluded from the calculation of calibration and information score in ANDURIL while in principle the scores could be computed with the remaining items.

- Plotting the range graphs of each variable per expert (i.e. expertwise). The function `plotting_itemwise` could (relatively easily) be modified in order to support this standard EXCALIBUR feature.

- A graphical user interface (supported by EXCALIBUR) which for some practitioners may be more accessible than a command line of matlab.

- Discrepancy analysis which is supported by EXCALIBUR.

- Bayesian combination which is very seldom used but is supported by EXCALIBUR.
- ANDURIL does not support elicitation of multivariate uncertainty [6]. This feature is also not available in EXCALIBUR.

6. Final Comments

A MATLAB toolbox named ANDURIL was developed to support decision making under uncertainty, when expert judgments are combined by applying Cooke's classical model for structured expert judgment. The main purpose for developing this toolbox is to create an open source software that can be used by practitioners and researcher who are interested in applying or further developing Cooke's method. The developed tool was validated with the closed source software EXCALIBUR. For this purpose a recent study concerning green house gases emissions in Mexico was used as a test case. It was shown that ANDURIL can reproduce accurately the results of EXCALIBUR.

The advantages of having a transparent open source software for applying Cooke's method were discussed. The developed toolbox can be used to investigate different ways of calculating the intrinsic range of the aggregated opinions that may result in differences in the performance measures of the obtained DMs. Moreover, it is possible to provide the analyst with the weights of each expert per item when the item weights weighting scheme is considered. Also, it gives the opportunity to the user to calculate the integrated cumulative distribution of the DM considering in the density every percentile provided by every expert with non-zero weights, rather than just interpolating in between the 5^{th}, 50^{th} and 95^{th} percentiles of the DM. Finally, the robustness of the obtained DM can be investigated while excluding more than one seed item at a time. Surely, other possibilities than the ones discussed in this paper may be explored further by researchers interested in the method.

Concluding, the authors want to stress that the developed tool constitutes a first step towards an open source version of Cooke's classical model. Despite the limitations of the current version of ANDURIL, it is to the authors belief that the developed toolbox will be valuable to those who are interested in further investigating and applying the method. Some possible extension of the toolbox currently available in EXCALIBUR and not in ANDURIL have been discussed. It is the ambition of the authors to extend ANDURIL also with the more recent techniques of elicitation of multivariate dependence.

Acknowledgements

This research is partly supported by the EUROS research programme, which is supported by NWO domain Applied and Engineering Sciences and partly funded by the Dutch Ministry of Economic Affairs. We also acknowledge the support of COST Action IS1304 "Expert Judgment Network: Bridging the Gap Between Scientific Uncertainty and Evidence-Based Decision Making"

References

[1] R. M. Cooke, Experts in Uncertainty: Opinion and Subjective Probability in Science, Environmental ethics and science policy, Oxford University Press, 1991.

[2] R. M. Cooke, L. L. Goossens, TU Delft expert judgment data base, Reliability Engineering and System Safety 93 (5) (2008) 657 – 674.

[3] A. R. Colson, R. M. Cooke, Cross validation for the classical model of structured expert judgment, Reliability Engineering and System Safety 163 (2017) 109 – 120.

[4] J. W. Eggstaff, T. A. Mazzuchi, S. Sarkani, The effect of the number of seed variables on the performance of Cooke's classical model, Reliability Engineering and System Safety 121 (2014) 72 – 82.

[5] D. Puig, O. Morales-Nápoles, F. Bakhtiari, G. Landa, The accountability imperative for quantifying the uncertainty of emission forecasts: evidence from Mexico, Climate Policy 18 (6) (2018) 742–751.

[6] C. Werner, T. Bedford, R. M. Cooke, A. M. Hanea, O. Morales-Nápoles, Expert judgement for dependence in probabilistic modelling: A systematic literature review and future research directions, European Journal of Operational Research 258 (3) (2017) 801 – 819.