Write-up

Yunpeng Wu(4362170)

June 11, 2020

1 Exposé

1.1 The Problem

The Wason selection task is a famous four-card selection task. which is first devised by Peter Cathcart Wason in 1966. In this task, There are four cards that have a letter on one side and a number on the other side. A rule is introduced to the reasoners in a form of "If p, then q". The reasoners select which cards have to be turned to prove if the rule is true or false. Logically, The cards can be regarded as p,$\overline{p}$, q,$\overline{q}$, and the rule states that p implies q. In accordance with the truth table of the implication, The rule is false if and only if p is true and q is false. Therefore, individuals have to select cards "p" and "$\overline{q}$" to test the truth of the rule. However, only a few of the participants have the correct response. The researchers have formulated many theories to account for this phenomenon by now. In this thesis, I intend to focus on two of them. The question in my thesis will be centered on how good these theories are and how accurate using a computer approach to implement these theories can predict an individual's card-selection behavior.

1.2 The approach

What I intend to implement using python is a inference model[6] and a model of the insight theory of the WST[9]. I would like to use Leave-one-out cross-validation to split the test sets and training sets, because of the minor data sets. The training part will be implemented using EM-algorithm to estimate the parameters. Then the evaluating part will calculate the mean squared error between the estimated values and the actual value to estimate how accurate are the estimates.

1.2.1 Leave-one-out cross-validation

Leave-one-out cross-validation [10] is a special case of cross-validation. The main idea is that the number of split folds equals the number of instances in the data sets. Hence, each instance can be selected as a single-item test set using all other instances as a training set. This approach is proper for small data sets in particular.

1.2.2 The bootstrap method

The bootstrap method can be also utilized to split data sets. In contrast with the leave-one-out CV, The bootstrap method[7] is sampling uniformly n instances with replacement from the data set that has a size of n. The probability of an instance not being chosen is $(1-\frac{1}{n})^n \approx e^{-1} \approx 0.368$. That is to say, Approximately 36.8 percent of the instances is not in the traning set, which can be used as test sets.

1.2.3 EM-algorithm

The EM algorithm [4] provides an iterative method of obtaining maximum likelihood estimates (MLEs) for a model where some data may be regarded as "missing". The algorithm is useful for general processing tree models.

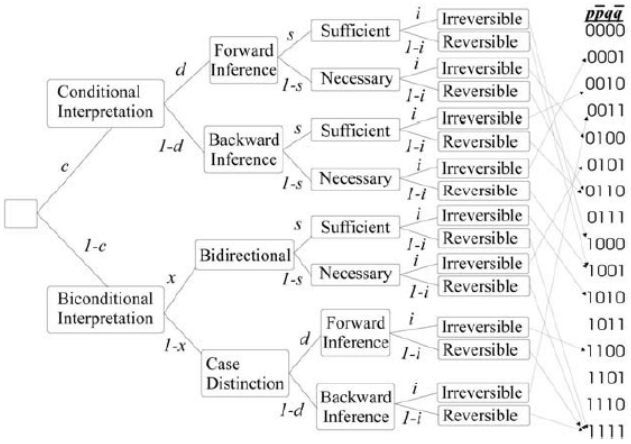

Figure 1: Processing-tree representation of the inference model[6]

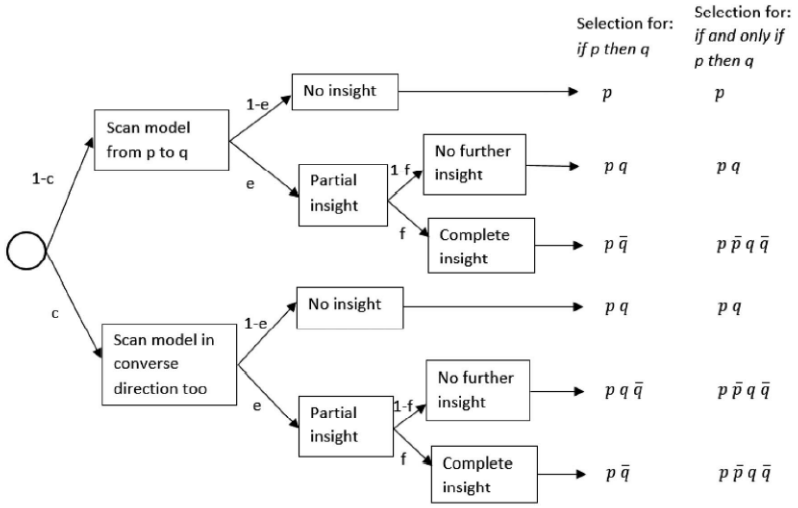

Figure 2: A multinominal process-tree for the model theory[9]

1.3 The Timeline

Week	Activity
Mai 11^{th} - Mai 24^{th}	Implementation Read literature
Mai 25^{th} - Mai 31^{th}	Implementation Analysis
Jun 1^{st} - Jun 7^{th}	Writing First draft
Jun 8^{th} - Jun 28^{th}	Read literature Writing
Jul 6^{th} - Jul 26^{th}	Writing Second draft
Jul 27^{th} - Aug 7^{th}	correction Final thesis

2 Wason Selection Task

In 1966, Peter Cathcart Wason contrived a famous reasoning tasks the Wason selection task for researching propositional reasoning, commonly abbreviated as WST. In this task, There are four cards presented on the table, given a rule in terms of "If p, then q.", such as "If an A is on one side, then there is a three on the other side.". Both sides of them have a number or letter. For example, the visible sides of cards might be A, B, 3, and 4. The reasoners have to select which cards should be turned to verify the truth of the hypothesis. Logically, the counterexample of the rule is the correct selection in the task[6]. In other words, reasoners should turn the card "A" and "4" to refute the hypothesis.

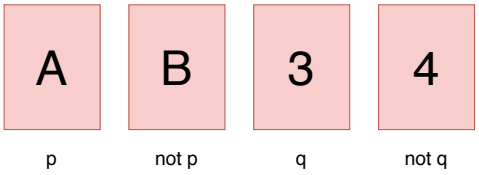

Figure 3: The cards present in the Wason selection task

By now, the WST has impelled myriads of research. One of the reasons is that the Wason selection task is tough to be solved by human reasoners[6]. In experiments, individuals rarely select the correct pairs, the counterexample of the hypothesis[9]. Most participants select the combination p and q or only the p card[9]. It seems that human can not be able to reason logically in this task. However, some theories demonstrate that the false response is also the

result of logical reasoning. In this article, two of them will be explicitly introduced in detail and implemented by python. The program also evaluates how accurate these theories can predict human behavior.

2.1 Notation

In this article, the given rule of the WST is in the form of "If p, then q.". The cards can be indicated by the p card, the $\overline{\text{p}}$ card, the q card, the $\overline{\text{q}}$ card, respectively, in terms of their associations with the proposition p and q. For example, If an A is on the letter side, then there is a three on the number side. The p card can be represented as the card with the visible side presenting an A, and the $\overline{\text{p}}$ card denotes that card presenting a B. Another two cards can be indicated analogously. Furthermore, the selection of reasoners is described by (x1, x2, x3, x4), which x1, x2, x3, and x4 are either one or zero. $x_i = 0$ means that the participant did not turn this card i. Inversely, $x_i = 1$ represents the card i is turned to verify the hypothesis. Hence, there are 16 combinations to symbolize the various selection of participants.

3 Multinomial Processing Tree Model

The multinomial processing tree, abbreviated as MPT, is wide-used to describe psychological and cognitive theories. For example, The MPT can illustrate the inference model and the insight model. The MPT model is not only easy to be created but also developed solely for processing categorical data. Each branch in MPT represents one or more processing sequences and falls into a category. In other words, the MPT model denotes that the observed categorical data results from one or more latent processing sequences[1]. Moreover, the probability of a branch is easily computable using the product of the probabilities of all corresponding links on this branch.

4 The Inference Model

The distinctive approaches of dealing with conditional statements are the main reason for individuals' differences[2]. Four conditional inferences might be invited in the reasoning process. Two of them are valid, and the other two inferences are invalid. The following table is consists of interpretations involving different conditional inferences, the origin rule "If an A is on the letter side, then there is a 3 on the number side.".

Validity	Inference	Interpretation
Valid	Modus Ponens(**MP**): Given p, it is concluded that q.	The letter side is A, therefore the number is 3.
Valid	Modus Tollens(**MT**): Given $\overline{q}$, it is concluded that $\overline{p}$.	The number side is not 3, therefore the letter is not A.
Invalid	Denial of the Antecedent(**DA**): Given $\overline{p}$, it is concluded that $\overline{q}$.	The letter side is not A, therefore the number is not 3.
Invalid	Affirmation of the Consequent(**AC**): Given q, it is concluded that p.	The number side is 3, therefore the letter is A.

The reasoners typically interpret the rule from the visible side using the available inferences. The inferences, however, can sometimes be applied to the invisible side also. There are five parameters in the inference model, which are defined as conditionality vs. biconditionality, bidirectionality vs. case distinction, direction, perceived sufficiency vs. necessity, and irreversibility[6]. They are represented by c, x, d, s, and i, respectively.

Individuals might interpret the rule in various ways and make different inferences to select cards. The rule is commonly seen as an inviting or warranting forward inference, which is from letter to number, but a backward(from number to letter side) inference may sometimes be made. For example, When the rule "If p then q" is regarded as in the form of "p only if q", a backward inference is more frequently carried out. Moreover, Which inference may be performed also depends on whether the individual perceives the antecedent as sufficient or necessary.

If the rule is understood as a warranting forward inference and the antecedent seen as sufficient, the warranted inference is MP. In this case, the individual infers that the selection p and assumes that the invisible side of card p is q. The individual understands the rule in the form of "p only if q" if the rule is comprehended as a backward inference. In this case, the antecedent is q. Hence, the individual imagines that the invisible side of the card q is p and select it. The warranted inference is AC. Analogously, if the rule is seen as a warranting forward inference and the antecedent is necessary, the individual understands the rule in the form of "q only if p" The necessarily of the antecedent implies that the invisible side of card p may be q or not q, and the invisible side of card not p is not q. Therefore, the individual infers to select card not p. The warranted inference is DA. If the direction is backward, the comprehended rule is in the form of "p if q". Therefore, the individual concludes that the invisible side of the card not q should be not p and turns it. It is MT. The following graph shows that the relationship between sufficiency and necessarily, direction, and inferences.

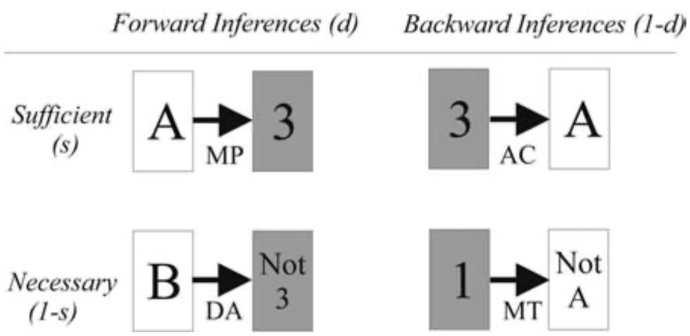

Figure 4: Relationship between sufficiency and necessarily, direction, and inferences for the rule "If there is an A on the letter side, then there is a 3 on the number side."[6]

On the other hand, the rule can be seen as a biconditional rule rather than a conditional. That is to say, the rule is understood in the form of "If and only if p then q." There are two options for representing the biconditional using conjunctions of conditional. Firstly, the biconditional can be constituted by both directional implication such as "If p then q and If q then p.". Furthermore, the case distinction is also an approach to represent the biconditional, for example, "If p then q and if not p then not q." or in the reverse direction "If q then p and if not q then not p.". For instances of bidirectional implications, if the individual invites an inference, this inference should be applied to both implications, such as for MP, the selection should be card p and card q. For the case distinction, if the rule is seen as "If p then q" and "If not p then not q", the inference is carried out to both conditional. For example, if the invited inference is MP, then the individual select card p and not p. The same selection results from the inference AC. Analogously, if the rule is understood in the reverse direction, such as "If q then p" and "If not q then not p", the selection is q and not q invited inference DA or MT.
So far, we assume that the inference is applied to the visible side of the card, but it can also be drawn on the invisible side. For example, if the inference MP is available, the individual may assume that p is on the invisible side of the card not q. According to the inference MP, the visible side should be q rather than not q. It is contradicting with the visible result. The individual infers that p must not be on the invisible side and therefore turns this card.

4.1 The Specific Algorithm

Firstly, the algorithm generates a dictionary of rules in terms of figure 4, which maps the assumptions to the outcomes. After that, it generates a list containing all results of assumptions. If conditional(biconditional) is seen as a forward inference or backward inference(bidirectional or case distinction), the algorithm deletes the result of conditional or biconditional from the list. Subsequently, the algorithm calculates the intersection of elements in the list to determine which cards should be turned. At last, if assumptions contain reversible, the available inference is applied to the invisible side, and the card which contrasts with the visible side is selected.

5 The Insight theory

The insight theory postulates that various levels of insight on the importance of the counterexamples and the individual's scanning direction may result in different card selections[9]. Three parameters govern the MPT of this theory: Scanning from p to q vs. Scanning in both directions, No insight vs. Partial Insight, and No further insight vs. Complete Insight.

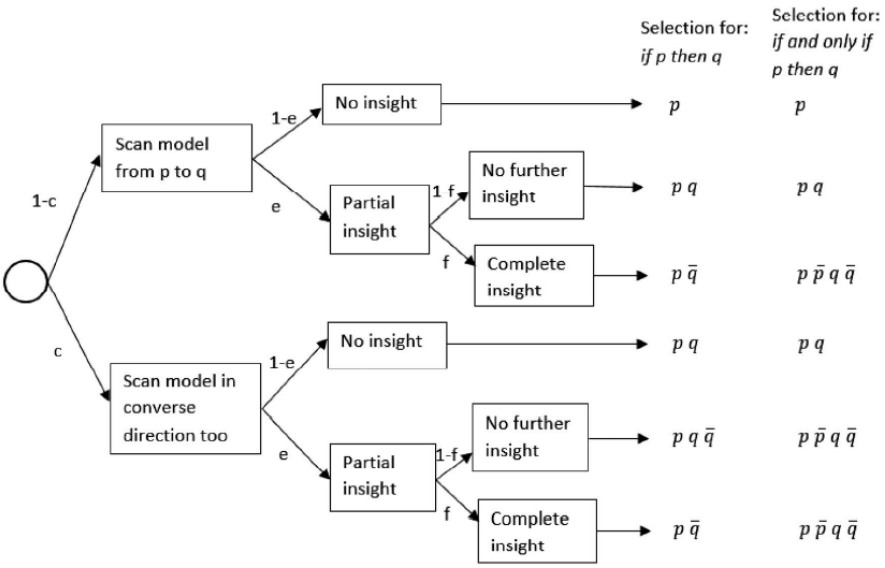

Figure 5: The multinomial processing tree of the model theory[9]

The model theory demonstrates the possibility and impossibility of the hypothesis. It can be used to achieve the same functionality as the insight theory. According to the dual-process theory, there are two systems applied for human reasoning in which system 1 generates immediate intuitions, and system 2 makes slow deliberations. Firstly, the model theory generates a mental model of the conditional underlying system 1, which depicts those clauses that are true in a possibility.[5]. The mental models of the hypothesis "If an A is on one side, then there is a 3 on the other side." are

A 3

...

It indicates that the possibility in which A and 3 are true. The ellipsis represents possibilities that the antecedent of conditional is false. That means that there are possibilities for the hypothesis other than the conjunction of A and 3[3]. If the individual continues to think explicitly possibilities that the antecedent is false, the mental model can flesh out to the full explicit model[5],which results from deliberation(system 2). The fully explicit model generates all conjunctions of the possibilities:

$$\begin{array}{l} \mathrm{A}\,3 \\ \overline{A}\,3 \\ \overline{A}\,\overline{3} \end{array}$$

Individuals can simply infer that the counterexample(A and $\overline{3}$) of the hypothesis is impossible from the fully explicit model. Furthermore, the two cases containing $\overline{A}$ are possible whether the hypothesis is true or not. For example, given that the hypothesis is false, we can infer that the conjunction of A and 3 is impossible, and the two cases containing $\overline{A}$ is possible from the fully explicit model. That is, the truth of the hypothesis does not influence the possibility of those two cases. Therefore, there are only one possibility and one impossibility implied by the truth of the hypothesis.
The algorithm of the selection task is based on the model theory and the insight theory. Firstly, individuals generate the mental model of the hypothesis "If p then q.", and the program initials an empty list used to contain the selected card. If the individual scans the mental model forward, the program adds a p in the list. Moreover, if the individual scans in both directions, it adds p and q to the list. In this case, the individual has no insight on the counterexample of the hypothesis. Nevertheless, It is possible to develop the mental model to the fully explicit model with a partial insight using system 2. According to the fully explicit model, the program adds $\overline{q}$ to the list for falsifying the hypothesis if q is on the list. If not, the program adds q to the list in order to verify the hypothesis. In addition, when the individual has full insight into the counterexample from the outset, the program selects the counterexample of the hypothesis as the selection.

5.1 The Specific Algorithm

In the beginning, the algorithm processes the given list of conditional to obtain propositions. Then it generates the mental model of the conditional and selects cards in terms of the scanning direction. If the assumptions have no insight on the counterexample of the hypothesis, it returns the selected cards. However, if the assumption is partial insight, the algorithm produces a fully explicit model from the mental model and select the card that can verify the hypothesis or falsify it. When the assumption is fully insight, the algorithm will provide all conjunctions of the hypothesis and find the impossibility conjunction that is not in the fully explicit model, the counterexample, then return it as the selection.

6 EM Algorithm

6.1 The specific implementation process of EM algorithm

The EM algorithm contains two steps. An expectation followed by a maximization step.
Firstly, The program can accept three(for the model theory) or five(for the inference theory) numbers as the initial parameters. In the expectation step, I calculate the expected value with the parameters,which is obtained from the last iteration or the initial parameters, using this formula [4]: $E_{ij} = \frac{n_j p_{ij}}{p_j}$, where n_j is the observed category data from a sample in the train

set and p_{ij} is the branch probability with the i-th branch of the j-th category and p_j is the category probability.
In the maximization step, the program computes the new parameters using the expected values from the expectation step. The main idea is to calculate the sum **a** of probabilities of the branches, which has the estimated parameter, and the sum **b** of probabilities of all branches. The new value of the parameter for the next iteration is $\frac{\mathbf{a}}{\mathbf{b}}$.

7 The leave-one-out and The bootstrap method

7.1 The Leave-one-out Method

Leave-one-out cross-validation is a particular case of the k-fold cross-validation, which k = 1. In other words, the number of folds is equal to the number of instances in the data set[10]. In the training procedure, Each instance can be chosen as a test case and all others in the train set. It can obtain more valuable information arising from the more training times. Hence, This approach is commonly applied to the smaller data set.
This article distinguishes two different Leave-one-out methods by leaving an experiment or a participant. There are three types of experiments in the given data set, respectively, abstract hypotheses, everyday hypotheses, or deontic principles. Firstly, the method selects all experiments with the same sort of generalization, as a primary data set. Then it chooses a participant from one experiment of the primary data set as the test case. In the subsequent step, all other experiments in the primary data set are used to obtain the distribution information using the EM algorithm. In terms of the obtained information, the program can predict which cards the participant selects. Then the program picks the next participant to predict. This procedure may be operated hundreds of times, depending on how many participants in the test case. The accuracy of the model can be calculated using the number of the correct predictions divided by the number of participants in the test case.
On the other hand, the method can select experiments with close numbers of participants from the data set. For example, The number of participants in each chosen experiment has a difference smaller than 20 with each other. Then one of the chosen experiments can be seen as the test case, and all other selected experiments are used to train the models. The training part returns the estimated parameters using the EM algorithm. After that, the program calculates the Euclidean distance using the predicting result and the test case. Subsequently, the procedure will be repeated until all folds have been seen as the test case. In contrast, the method can also select experiments with more considerable differences in the number of participants. In this case, the program evaluates the parameters using the cosine similarity, which is not influenced by the number of participants. The following shows the formulas and codes to compute the corresponding similarity.

7.2 The Bootstrap Method

The bootstrap method[7] is sampling uniformly n instances with replacement from the data set that has a size of n. The probability of an instance not being chosen $(1-\frac{1}{n})^n \approx e^{-1} \approx 0.368$.

Algorithm 1: Cosine Similarity

Input: v1: the vector of the distribution obtained by the estimated parameters
Input: v2: the vector of the distribution of the test case
Output: cos_sim: the cosine similarity of two given vectors
the_dot_product = v1 $*$ v2;
abs_norm = norm(v1) * norm(v2);
cos_sim = the_dot / abs_norm;

Algorithm 2: Euclidean Distance

Input: v1: the vector of the distribution obtained by the estimated parameters
Input: v2: the vector of the distribution of the test case
Output: euc_dis: the euclidean distance of two given vectors
euc_dis = 0;
for $i = 0$ **to** *length(v1) - 1* **do** euc_dis = euc_dis + square(v1[i] - v2[i]);
euc_dis = sqrt(euc_dis);

That is to say, Approximately 36.8 percent of the instances is not in the training set, which can be used as test sets.

7.2.1 The computing approach of the goodness-of-fit in the bootstrap method

The program uses the log-likelihood ratio test(G-test) [8] for computing the goodness-of-fit. This test is calculated by taking the observed number, dividing it by the expected number, then taking the natural log of this ratio. The associated formula [4] is $\mathbf{G}^2 = 2\sum_{j=1}^{J} n_j \log(\frac{n_j}{n\tilde{p}_j})$, where n_j is the observed number and $(n\tilde{p}_j)$ is the expected number.
There are many test samples to evaluate the trained parameters in the bootstrap method. After using the G-test, the program receives the goodness-of-fit for each test sample. I want to discuss four approaches to calculate the whole goodness-of-fit for the parameters.
Addition: Just calculate the sum of all goodness-of-fit in the test set.
Mean: In contrast to the addition, the arithmetic mean can report the central tendencies.
More specifically, if two test sets have the same size, both test sets have the same goodness-of-fit by addition or mean. That is to say, the addition and the mean of that have not any differences in this case. Moreover, Both approaches can be influenced by the values that very larger or smaller than most of the values. For example, Values that have small fluctuating differences obtain smaller goodness-of-fit than those with a very large number. As a result, I think that the **standard deviation** can clearly represent the goodness-of-fit. It measures the spread around the mean. The test set that has a smaller standard deviation has a narrower spread of measurements around the mean. Therefore, the test set has fewer small or large values than the data set with a higher standard deviation. In comparison with the **variance**, the standard deviation is represented in original scale and more easily to read. From my views, The mean and the standard deviation are the best choice for computing the whole goodness-of-fit in the bootstrap method.

8 Results

8.1 Type: "-m", Initial Theta: 0.5 0.5 0.5

theta_c	theta_e	theta_f	goodness_of_fit
0.571773	0.173800	0.471774	10.449637
0.571773	0.173800	0.471774	5.117153
0.571773	0.173800	0.471774	171.444078
0.571773	0.173800	0.471774	21.089637
0.571773	0.173800	0.471774	33.987683
0.571773	0.173800	0.471774	126.652243
0.571773	0.173800	0.471774	54.314158
0.571773	0.173800	0.471774	5.995635
0.571773	0.173800	0.471774	13.629760
0.571773	0.173800	0.471774	17.250342
0.571773	0.173800	0.471774	14.818855
0.571773	0.173800	0.471774	48.091411
0.571773	0.173800	0.471774	42.941456
0.571773	0.173800	0.471774	32.904662
0.571773	0.173800	0.471774	33.526656
0.571773	0.173800	0.471774	24.552571
0.571773	0.173800	0.471774	101.987144
0.571773	0.173800	0.471774	142.797317
0.571773	0.173800	0.471774	85.978234
0.571773	0.173800	0.471774	83.555282
0.571773	0.173800	0.471774	67.659709
0.571773	0.173800	0.471774	48.267784
0.571773	0.173800	0.471774	72.647545
0.571773	0.173800	0.471774	69.178354
0.571773	0.173800	0.471774	60.568561
0.571773	0.173800	0.471774	37.330574
0.571773	0.173800	0.471774	236.727879
0.571773	0.173800	0.471774	31.900005
0.571773	0.173800	0.471774	5.442196
0.571773	0.173800	0.471774	24.798712
0.571773	0.173800	0.471774	19.907850
0.571773	0.173800	0.471774	22.176920
0.571773	0.173800	0.471774	35.621284
0.571773	0.173800	0.471774	3.247540
0.571773	0.173800	0.471774	8.423348
0.571772	0.173800	0.471774	22.030011
0.571773	0.173800	0.471774	749.040176
0.571773	0.173800	0.471774	33.195134
0.571778	0.173799	0.471774	268.971435

0.571802	0.173794	0.471774	31.901201
0.571804	0.173793	0.471774	5.442249
0.571767	0.173801	0.471774	11.668751
0.571726	0.173810	0.471774	6.438177
0.571785	0.173797	0.471774	21.326789
0.571618	0.173834	0.471774	74.394478
0.572049	0.173740	0.471774	6.963029
0.571306	0.173902	0.471774	80.019560
0.573487	0.173431	0.471774	10.793143
0.569443	0.174313	0.471774	63.895972
0.577945	0.172503	0.471774	25.463794
0.563432	0.175692	0.471774	38.545952
0.568076	0.174670	0.471774	7.897496
0.556154	0.175628	0.471774	6.348391
0.591424	0.172063	0.471774	355.680964

theta_c	theta_e	theta_f	goodness_of_fit(mean, SD)
0.619177	0.256679	0.382353	(37.08, 63.73)
0.530013	0.048190	0.739130	(91.53, 183.71)
0.719454	0.163343	0.631579	(43.38, 108.09)
0.580890	0.086678	0.625000	(51.82, 104.29)
0.866543	0.155864	0.125000	(97.5, 186.13)
0.514935	0.077134	0.043478	(136.74, 285.48)
0.543522	0.045516	0.807692	(76.23, 141.34)
0.563234	0.102858	0.416667	(52.67, 68.0)
0.694752	0.078608	0.071429	(111.84, 171.12)
0.577594	0.130766	0.576923	(46.74, 80.8)

8.2 Type: "-i", Initial Theta: 0.5 0.5 0.5 0.5 0.5

theta_c	theta_d	theta_x	theta_s	theta_i	goodness_of_fit
0.552347	0.255221	0.337358	0.443864	0.855596	34.973288
0.552347	0.255221	0.337358	0.443864	0.855596	46.914167
0.552347	0.255221	0.337358	0.443864	0.855596	27.770913
0.552347	0.255221	0.337358	0.443864	0.855596	917.740650
0.552347	0.255221	0.337358	0.443864	0.855596	576.655172
0.552347	0.255221	0.337358	0.443864	0.855596	569.871171
0.552347	0.255221	0.337358	0.443864	0.855596	468.540610
0.552347	0.255221	0.337358	0.443864	0.855596	600.804264
0.552347	0.255221	0.337358	0.443864	0.855596	439.474196
0.552347	0.255221	0.337358	0.443864	0.855596	524.856397
0.552347	0.255221	0.337358	0.443864	0.855596	500.527152
0.552347	0.255221	0.337358	0.443864	0.855596	567.863263

0.552347	0.255224	0.337358	0.443866	0.855596	513.333406
0.552347	0.255187	0.337359	0.443840	0.855596	715.900113
0.552347	0.255605	0.337330	0.444099	0.855596	678.193373
0.552347	0.253744	0.337556	0.442974	0.855596	366.394208
0.552347	0.278966	0.337847	0.456143	0.855596	160.882741

theta_c	theta_d	theta_x	theta_s	theta_i	goodness_of_fit(mean, SD)
0.592593	0.305239	0.239728	0.354131	0.777778	(168.51, 247.74)
0.555556	0.213262	0.303583	0.162930	0.629630	(135.46, 274.35)
0.347826	0.318236	0.419764	0.227595	0.782609	(207.76, 280.09)
0.515924	0.365577	0.391131	0.400117	0.812102	(114.33, 163.26)
0.760000	0.362351	0.085838	0.368460	0.560000	(237.57, 382.57)
0.471545	0.307381	0.282734	0.230705	0.520325	(195.49, 355.07)
0.810811	0.235752	0.117627	0.205001	0.351351	(301.3, 479.2)
0.774194	0.196070	0.142760	0.179424	0.387097	(270.48, 484.44)
0.750000	0.535346	0.195956	0.602205	0.926056	(178.4, 222.29)
0.300000	0.208443	0.563095	0.131802	0.800000	(163.9, 256.35)

References

[1] William H Batchelder and David M Riefer. Theoretical and empirical review of multinomial process tree modeling. *Psychonomic Bulletin & Review*, 6(1):57–86, 1999.

[2] Jonathan St BT Evans, Simon J Handley, Helen Neilens, and David E Over. Thinking about conditionals: A study of individual differences. *Memory & cognition*, 35(7):1772–1784, 2007.

[3] Keith James Holyoak and Robert G Morrison. *The Cambridge handbook of thinking and reasoning*, volume 137. Cambridge University Press Cambridge, 2005.

[4] Xiangen Hu and William H Batchelder. The statistical analysis of general processing tree models with the em algorithm. *Psychometrika*, 59(1):21–47, 1994.

[5] Philip N Johnson-Laird, Sangeet S Khemlani, and Geoffrey P Goodwin. Logic, probability, and human reasoning. *Trends in cognitive sciences*, 19(4):201–214, 2015.

[6] Karl Christoph Klauer, Christoph Stahl, and Edgar Erdfelder. The abstract selection task: New data and an almost comprehensive model. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 33(4):680, 2007.

[7] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[8] John H McDonald. *Handbook of biological statistics*, volume 2. sparky house publishing Baltimore, MD, 2009.

[9] Marco Ragni, Ilir Kola, and Philip N Johnson-Laird. On selecting evidence to test hypotheses: A theory of selection tasks. *Psychological bulletin*, 144(8):779, 2018.

[10] Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.