Rain(-drop) Your Bias

Zeynep Bolluk, Krishnendu Bose

Leonard Frommelt, Hyerim Hwang

Partha Pratim Kalita, Rowan Mohamed

Hardika Potey, Milad Rouygari

Introduction

What is the phenomenon you want to model? (0.5 points) Contextual information about certain criteria for making decisions plays a vital role in our decision-making process. For instance, in (Horinouchi et al., 2022) it was shown that using colors that are known from traffic lights(i.e. red and green traffic lights) leads to accordingly appropriate reaction times. This can be interpreted in a way, that the properties of the objects that constitute evidence in favor of certain decisions, influence the decision-making itself.

We hypothesize that prior knowledge about the objects that are used in a moving dot paradigm influences the reaction times in the following way: Experience with the real world has taught us, that raindrops usually fall downwards. Therefore, reaction times should be faster and/or decision-making more precise, in conditions where this real-world property is reflected in the experiment (conditions: ”plausible”, when the majority of raindrops fall, ”unplausible” otherwise, ”neutral” for neutral stimuli).

Why is that phenomenon relevant for understanding human cognition? (0.5 points) This study explores the impact of contextual knowledge on decision-making, focusing on the use of Drift Diffusion Models (DDMs) for modeling. Context includes factors and forces that make a situation distinct and understandable and misinterpretation of facts can lead to certain illogical decisions. It also includes mental pictures and risk attitudes(Pomerol, Brézillon, & Pasquier, 2002). Humans prefer to discover logical reasons for contingent events, implying a need for rationality in generating scenarios and proceduralizing contextual knowledge (Pomerol et al., 2002). Contextual knowledge impacts decision-making, but its impact depends on the cognitive processes involved with the contexts, apprehensions, and decision-making.

Methods

Why is this modeling method appropriate for understanding the phenomenon? (1 point) Drift diffusion models are a well-known modeling tool for studying binary decision-making. Numerous studies have demonstrated its effectiveness in simulating such occurrences and can be used to quantify biases. It is especially important to note their efficacy concerning quantifying bias effects in tasks that might be prone to those biases. The DDM precisely reflects two-alternative forced-choice behavior by simulating the accumulation of noisy data over time, making it appropriate for researching decision-making processes (Gupta et al., 2022).

The DDM describes decision-making as the gradual collection of data leading up to decision thresholds. The diffusion model in the current study is selected due to its capacity to fit Reaction Times(RT) and accuracy data, as well as to interpret parameters under various circumstances(Leite & Ratcliff, 2011).

Which hypothesis/hypotheses do you seek to test by contrasting two (or more) models? (1 points) With our modeling strategy, we are planning to contrast and compare the likelihood of the following hypotheses:

H1: Immediately after perceiving the identity of the stimuli, participants form a hypothesis about movement direction, that is biased by the contextual knowledge about the stimulus. This would be reflected by the high likelihood of a model, that considers the starting point (z0) as a linear function of the experiment condition (”plausible”/”unplausible”).

H2: Contextual knowledge about an object changes how participants interpret it as evidence for the sake of making a decision one way or the other. This means that each individually perceived object contributes to the decision-making process differently, e.g. a raindrop falling downwards will be interpreted as providing twice as much evidence for the ”the answer is down” decision, as compared to the perceived amount of evidence a raindrop falling up supplies for the ”the answer is down” decision. This would result in a high likelihood of a DDM that considers the drift rate (v) as a linear function of the experiment condition (”plausible”/”unplausible”).

These will be contrasted by an H0, which states that there is no effect of our conditions on the reaction times and that will be tested by using a DDM which is a naive model.

These hypotheses follow the reasoning of (Gupta et al., 2022). First, by interpreting the starting point of the DDM as bias favoring one of the two choices (usually acquired during the experiment by learning the prior distribution of stimuli, see e.g. (Simen et al., 2009)). Secondly, by interpreting the drift rate of the model as the process of accumulating evidence, including both physical properties of the experiment setup (luminance of stimuli, "true" movement direction, speed, orientation, etc) and cognitive and sensory properties of the participant (attention and eye-movement, interpretation of collected evidence, etc).

Description of computational model(s)

We implemented DDM to simulate decision-making processes in our experimental paradigm. Three variants of the DDM were constructed to investigate the impact of biased knowledge on response times.

What are the inputs, system properties, and outputs of your model(s)? (1 point) All our models output conditional distributions that are joint over reaction times and decision outcome (i.e. $p(rt, decision|inputs)$) in the form of relative frequencies.

The naive DDM does not require any inputs, it assumes that all (rt, decision) pairs are sampled from the same underlying distribution (in our case that relates to the hypothesis, that the prior knowledge does not influence the decision process).

The varying starting point/drift models take the dummy-coded conditions as input. Specifically, that means an input of ($"RainDropUp" = 0, "RainDropDown" = 1$) for the plausible condition, ($"RainDropUp" = 1, "RainDropDown" = 0$) for the unplausible condition and ($"RainDropUp" = 0, "RainDropDown" = 0$) for the neutral condition.

The variables of the model that are kept fixed during the parameter fittings, can be seen as the system properties of our model (although their impact can also be replaced by adjusting the free parameters, as discussed in the intro section of "Description of Computational Model", the exact fit of these depends on the system properties). Namely, these are the lower threshold a_{lower}, the upper threshold a_{upper}, and for simplicity the noise parameter std.

Which assumptions does each model make? (1 point) Our models assume that there exist relationships between starting point/drift and (stimulus x movement direction) in a way that is exemplified by our hypotheses. Note that this also implies the assumption that there is a relationship between z0/drift and the stimuli themselves (i.e. movement directions effect being zero) as would for example be the case if participants just generally react slower to raindrop stimuli.

Additionally, most assumptions of the default DDM are still made.

Describe the computational implementation of each model (e.g., model formulas) (1 point) The models were specified as follows:

- Naive Model: In this model, we used the parameter space without specifying any relation or variations in specific parameters. The model includes parameters for drift rate (v), threshold (a), starting point (z), and non-decision time (t). The default DDM models the amount of collected evidence in favor or against a binary decision as a value z, that changes over discrete time steps t. When the value z hits one of two thresholds a_{lower} or a_{upper}, this is interpreted as making a decision (e.g. in our case $z \geq a_{upper}$ is interpreted as making the correct decision/the decision that the distribution of evidence suggests).

  Potential free parameters of this model are:

  – starting point z_0, the initial value of z, relates to a bias in the decision making process
  – non-reaction-time nrt, the initial value of time-step t, relates to any time that is needed for reacting additionally to the decision process itself, for instance sending motor signals from the brain to the arms.
  – drift rate v, a constant that is added to z at each timestep and that reflects the average amount of evidence in favor or against a decision, as well as the process of collecting and interpreting this evidence.
  – Upper threshold a_{upper}, relates to the sensitivity of making a decision, generally a_{lower} is kept fixed since its impact can be replaced by adjusting a_{upper}, $z0$ and the signal-to-noise ratio.
  – The noise value std, is the standard deviation of zero-centered gaussian noise, that gets added to z at each time step. However, it is often kept fixed, since it constitutes the signal-to-noise ratio, which can alternatively changed by setting v and the range between a_{upper} and a_{lower} accordingly and because letting this parameter be free searches parameters in parameter fitting much harder. Additionally, a convention has to be found, regarding the mapping from the discrete time-steps t to seconds. In our model implementation, we choose a 1:1 mapping between t and milliseconds to have the same temporal resolution as our experimental data.

    In the following text, some of the above-defined parameters will be called differently, to be consistent with the formalization from the HSSM toolbox. $z0$ will be called z, nrt will be called t and a_{upper} will be called a.

- Varying Starting Point Model: This model focuses on bias, and incorporates participants' biased knowledge about the direction of raindrops. It assumes that existing knowledge will create a bias that affects the decision-making process. So, Z interacts with plausible conditions. All other parameters remained consistent with the Naive and V-model.

$$z \sim 1 + RainDropDown + RainDropUp, \tag{1}$$

will be modeled by adding the additional free parameters β_1 and β_2 such that:

$$z_0 = z + +\beta_1 * RainDropDown + \beta_2 * RainDropUp \tag{2}$$

Hence, the impact of the neutral stimuli in comparison to RainDropDown and RainDropDown will be reflected in z itself.

- Varying Drift Model: In contrast, the V Model focuses on the evidence accumulation process. It assumes the existing knowledge will affect the interpretation of evidence, so drift rate (v) will have a relation to plausible conditions. All other parameters remained consistent with the Naive and Z- models.

$$v \sim 1 + RainDropDown + RainDropUp, \quad (3)$$

will be modeled by adding the additional free parameters β_1 and β_2 such that:

$$z_t = z_{t-1} + v + \beta_1 * RDDown + \beta_2 * RDUp + N(0, std) \quad (4)$$

Again, the impact of the neutral stimuli in comparison to RainDropDown and RainDropDown will be reflected in v itself.

Description of the experiment

Provide an overview of the experiment. What are the independent variables and dependent variables of the experiment? (0.5 points) The experimental design is based on a task-switching paradigm [1] but with a slight modification. Unlike the conventional task-switching setup where participants react to either the orientation or the movement of a stimulus, in this modified version, participants are only required to respond to the movement of the object.

The stimuli used in the experiment are raindrops and circles. Participants indicate their responses by pressing the 't' key for upward movement and the 'v' key for downward movement. Each trial begins with the presentation of a fixation cross. if it turns blue, their task is to respond to the movement of the stimulus. Should the cross not appear, participants are not required to respond in that particular trial.

The dependent variables in this study are reaction time (rt) and the participants' responses, while the independent variables are the type of stimulus presented and its movement direction.

How much data was collected (number of participants and trials)? (0.5 points) Data collection for our study was a collaborative effort within our research group. Seven individuals from our team participated in the experiment on two occasions, while one member was able to participate only once. This resulted in a total of 15 participations. In the study, each participant underwent 112 trials in total, which included 16 training and 96 experimental trials; within the experimental set, 48 required responses to stimulus movement, while the other 48, acting as distractors, did not necessitate responses. The randomization of movement trials was not predetermined. A total of 720 trials were conducted across 15 participants, each completing 48 movement trials. However, 13 trials were removed, leaving 707 valid trials. Among these, 368 trials were associated with stimulus 1 and 339 trials with stimulus 4.

Model simulation

Describe the process of simulating data from the model(s). (1 point) For simulating data, we need to instantiate a model class with a given set of parameters. Then, we can sample from this model by repeatedly calling its 'generate_trajectory()'-method each time with the desired conditions (e.g. the same as in the experiment to compare output distributions) and for the desired amount of times (e.g. once for each data point in the behavioral data). We can then inspect the last generated trajectory, to get an intuition about how parameters influence the behavior of the model.

More importantly, we can also plot a histogram of the generated reaction times per each condition and response. For directly comparing simulated distributions (or for comparing them with other rt-distributions like the experimental data), there exists a ´compare()´-method that plots the condition-wise distributions of both models in the same plot. The depicted distributions are either absolute frequencies or can be normalized over the whole dataset (i.e. plotting joint distributions) or the marginal probabilities of the conditions (i.e. depicting the conditional distributions).

For usage examples, please refer to the notebook.

Model fitting

Describe the process of fitting the model(s) to the data. Remember to describe any preprocessing steps of the data. (2 points)

Parameter Search Space For the parameter fitting, we start with setting up a convenient parameter search space to encompass the parameters, drift rate (v), threshold (a), starting point (z), and non-decision time (t).
The following table shows the range of each parameter(parameter space):

v	a	z	t
-0.5 $\rightarrow$ 0.5	0.05 $\rightarrow$ 2.0	0.4 $\rightarrow$ 0.6	0.1 $\rightarrow$ 0.3

By defining the parameter space with appropriate prior distributions, we ensured that the drift-diffusion models could capture the variability in our experimental data.

Model Fitting Procedure The drift-diffusion models were fitted to the experimental data using a Bayesian approach with the Markov Chain Monte Carlo (MCMC) method. The following procedure was employed for model fitting:

- Data Preparation: Experimental data consisting of participants' responses regarding the direction of rain and a circle's movement were preprocessed and formatted for model fitting. We chose the correct column as a response variable. We coded the correct one as '1', and the incorrect one as '-1'. We also created a 'Plausibility'

[1] https://github.com/younesStrittmatter/rdkOnline

condition has three values: 1 (rain drop, falling -plausible), -1 (rain drops, going up unplausible), and 0 (all circle trial regardless of the movement). Then, we used dummy coding to implement our categorical variable to the model, which resulted in two variables: RainDrop_Up and RainDrop_Down

- Model Initialization: The Z (varying bias model), V (varying drift model), and the naive drift-diffusion models were initialized with the specified parameter search space, including prior distributions for each parameter, and the formulas for the Z and V models.
- MCMC Sampling: We utilized the No-U-Turn Sampler (NUTS) implemented in NumPyro for MCMC sampling. The sampler explores the parameter space by iteratively generating samples while ensuring efficient exploration and convergence.
- Sampling Configuration: MCMC sampling was conducted with two independent chains, each consisting of 2000 draws. An additional 2000 tuning steps were performed to adjust sampler settings and optimize convergence.
- Diagnostics of Results: We produced a summary for the Bayesian analysis, and we plotted the model trace (to follow the fitting process for each parameter) and model graph to follow the relation between parameters and plausibility in each model. By following this model fitting procedure, we obtained parameter estimates for the three drift-diffusion models, allowing us to assess decision-making processes and test our hypotheses regarding the influence of biased knowledge on response times.

Parameter recovery

Describe how you performed parameter recovery for your models. (1 points) First, we defined a parameter space, a Python dictionary containing the different ranges of the parameters. The data was simulated to generate response times and responses, with the randomly selected values from the defined parameter space. The DDM model was fitted to the response times using a log-likelihood function, employing hierarchical Bayesian inference to provide posterior distributions of model parameters.

After fitting the model to the simulated data, the posterior parameters were sampled. We printed out the true values and the posterior distribution of the fitted parameters to check if the parameter recovery is working or not and what parameters are recoverable.

Model comparison (& recovery)

Describe how you compared the models. (1 point) We performed model comparison using the ArviZ package and HSSM Toolbox, especially az.compare() function from the ArviZ library.[2] The output of this function provides a comprehensive comparison result. We focus mainly on the Expected Log Pointwise Predictive Density (elpd) value. After fitting the models using MCMC sampling, the elpd values for each model are computed using Leave-One-Out Cross-Validation (LOO-CV).

The Bayesian LOO estimate of out-of-sample predictive fit is given by:

$$elpd\ loo = \sum_{i=1}^{n} \log p(y_i|y_{-i}) \tag{5}$$

where

$$p(y_i|y_{-i}) = \int p(y_i|\theta)p(\theta|y_{-i})d\theta \tag{6}$$

is the leave-one-out predictive density given the data without the ith data point(Vehtari, Gelman, & Gabry, 2017).

This metric offers insights into the relative performance of each model. It evaluates the predictive performance of Bayesian models by measuring the goodness-of-fit to the observed data while penalizing for model complexity. According to Vehtari et al., higher ELPD indicates better predictive performance.

Optional: Describe how you performed model recovery. (0.5 bonus points) To perform model recovery, we first generated three separate datasets, each corresponding to one of three models: ”varying drift,” ”varying starting point,” and ”naive”. Following this, we fitted each simulated data to these respective models to calculate the probability P(fitted modelsimulated model). For determining these probabilities, we used the azċompare function, which includes a ”weight” column. According to the documentation, these weights can be approximately interpreted as the likelihood of each model being the correct one among those compared, given the data We then represented these probabilities in a confusion matrix, resulting in a 3x3 matrix(fig 1). It’s worth noting that we conducted the Markov Chain Monte Carlo (MCMC) sampling process four times; however, according to Wilson and Collins (2019), it should ideally be run 100 times to ensure more reliable results. We were unable to follow this recommendation because of limitations in computational resources.

Results

Simulation results

Which phenomena do the models capture and why? Make sure to support your argument with a plot. (1 point) The hypothesis-related models capture the phenomenon of a relation between the responses and (stimulus x moving direction). Note, that this also includes the phenomenon of a relationship between responses and the stimuli themselves (i.e. the influence of moving direction being zero) as could for instance be seen in generally lower reaction times/more errors for the conditions that contain raindrop stimuli.

Additionally, our models capture all the phenomena that default drift-diffusion models do, among others:

[2] python.arviz.org/en/stable/api/generated/arviz.compare.html

- Reaction Time Distributions: The predicted outcomes in general.
- Accuracy-Response Time Tradeoff: E.g. lower the threshold and starting point, for faster but less precise responses
- Neural Correlates: See (Gupta et al., 2022)

Which phenomena do the models not capture and why? (1 point) There are close to infinitely many phenomena in the real world but any model can only account for a finite (and in reality relatively small due to increased complexity) amount of phenomena. That is a general problem in Science. However, usually, Scientists are only concerned with one phenomenon at a time and the problem is rather to make sure that unobserved variables can be averaged as some form of noise.

The noise we are considering is purely Gaussian. That can be a problem when some non-gaussian noise plays a role. For instance, consider that during one trial a fly lands on the keyboard and the participant accidentally presses a button. This can lead to drastically different reactions during only a few trials. In fact, in our behavioral data there are some outliers (see fig 3 at ”wrong”) that our model will never be able to account for. With a high enough sample size, we could still hope that the mean of the non-gaussian noises is gaussian distributed due to the central limit theorem (just that a high enough sample size means high in that case). Of course, the non-gaussian noise could also come from internal brain processes and not necessarily the outside world.

Which parameters can be recovered more reliably, which less reliably? (1 Point)

The main goal was to assess how well these characteristics might be recovered from observed data, in terms of accuracy and reliability.

The table below shows the posterior parameter distribution after fitting our generated data to a naive model. When we analyzed our true parameters used at the time of simulated data generation and the posterior probability distribution of the fitted parameters, we found that parameters ’v’ and ’z’ can be recovered easily, but ’v’ is more reliable than z as r_hat of ’z’ is 2.01. In the table, we can see that ’t’ is also recoverable and very reliable but sometimes it goes out of the HDI 94% range. On the other way, ’a’ is highly unrecoverable in our case, so we feed a fixed value of ’a’ at the time of model fitting.

Optional: Model recovery

Which models can be recovered more reliably, which less reliably? (0.5 bonus points) The cells in the confusion matrix (see fig 1) represent the conditional probabilities of fitting a specific model given the simulated data. The off-diagonal cells represent the probability of selecting a particular model given the simulated data belongs to a different model. A model that fits better would have higher values along the diagonal and lower values in the off-diagonal cells. Overall, the ”naive” model has a high probability of being correctly fitted when the simulated data is actually ”naive.” The ”starting point” model has an average fit. The ”varying drift” model has the lowest probability of being correctly fitted, suggesting it is less distinct or well-defined compared to the other models in this analysis. It’s also noteworthy that there seems to be a substantial amount of confusion between the ”varying drift” and ”starting point” models, as indicated by the relatively high values off the diagonal (0.33 and 0.38, respectively).

Table 1: Posterior probability distribution of the parameters: For true values: ’v’: 0.009082133050440847, ’t’: 3.4676830527693725, ’z’: 0.5081241328370013

	mean	sd	hdi_3%	hdi_97%	r_hat
$z_{I}ntercept$	-0.568	1.476	0.004	0.041	1.42
$z_{r}ainDropUp$	1.078	1.37	0.004	0.041	1.42
$z_{R}ainDropDown$	0.644	0.009	0.004	0.041	1.42
t	1.78	1.3	3.993	4.00	1.01
v	0.014	0.024	-0.023	0.057	1
a	1.35	1.922	0	4.16	1.75

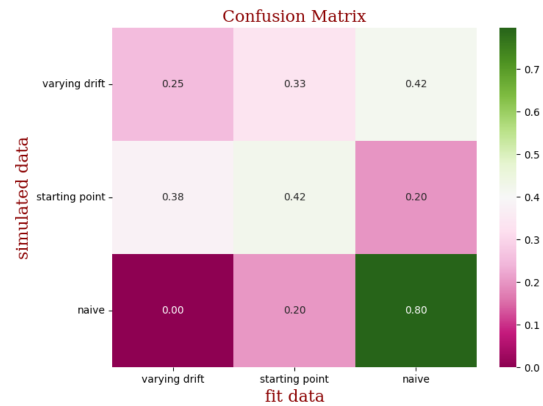

Figure 1: Model Recovery Confusion Matrix

Model comparison

Which models fit the data better and why? (1 points) The naive model is the better performing model among all as its value of elpd_loo is slightly higher than others(see Fig 2). Since all models’ value of Warning is ”False”, we can take elpd_loo as reliable (Vehtari et al., 2017). However, elpd_loo has a slight difference among the three models that might not indicate the best model robustly.

	rank	elpd_loo	p_loo	elpd_diff	weight	se	dse	warning	scale
naive model	0	-149.357171	3.696729	0.000000	1.000000e+00	33.287990	0.000000	False	log
varying starting point	1	-151.554075	7.191621	2.196904	0.000000e+00	33.938666	1.838696	False	log
varying drift	2	-151.880804	7.307885	2.523633	4.662937e-15	34.115254	1.905678	False	log

Figure 2: Model Comparison Results

Parameter fit

The results change each time we run the model, but still, the results are in a similar range with only small differences. So, we will report the results from the last run.

Which parameter values fit the data best? (1 point)

- Naive-Model:

parameters	lower	upper	mean	r_hat
v	2.154	2.765	2.467	1.00
t	0.233	0.308	0.308	1.01
a	0.964	1.178	0.476	1.00
z	0.346	0.476	1.178	1.01

  In the table, you can see the upper and lower thresholds for each parameter. R-hat is calculated for each parameter, which provides information on how well chains converge. According to the result, r-hat for v and a is 1, and for t and z, is 1.01. This indicates good convergence of the Markov chains, suggesting that the model has likely converged well.

- Varying-Drift-Model: The parameter values for V - Model that fit the data best are:

parameters	lower	upper	mean	r_hat
v-Intercept	2.201	2.853	2.530	1.0
v-RainDropDown	-0.325	0.191	-0.075	1.0
v-RainDropUp	-0.404	0.140	-0.135	1.0
t	0.229	0.307	0.268	1.0
a	0.964	1.189	1.083	1.0
z	0.336	0.468	0.407	1.0

  In the table, you can see the upper and lower thresholds for each parameter, together with mean and r-hat results. According to the result, the r-hat value for all the parameters is equal to 1. This suggests that the Markov chains have likely converged well, indicating good convergence of the model.

- Varying-Starting-Point-Model: The parameter values for Z the model that fits the data best are:

parameters	lower	upper	mean	r_hat
z-Intercept	0.354	0.531	0.417	1.0
z-RainDropDown	-0.045	0.046	-0.001	1.0
z-RainDropUp	-0.076	0.020	-0.028	1.0
t	0.001	4.613	0.272	1.0
a	0.959	1.174	1.062	1.0
v	2.120	2.730	2.443	1.0

  The table shows the upper and lower thresholds for each parameter, alongside their mean and r-hat results. The r-hat value for all parameters is 1, indicating likely good convergence of the Markov chains and the model.

Discussion

Which hypothesis does your modeling support and why? Base your answer on the model comparison (and model recovery) results. (1 point) The naive model as our winning model supports the null hypothesis which suggests the experimental results and the data are contingent on neither the pre-existing biases of the participants nor the delays in reaction time.

In the model recovery process, we found that the models with varying drift rates and starting points were not successfully recovered; only the naive model showed promises. This suggests that the model recovery phase was unable to accurately capture the critical elements of these models. Despite this setback, we proceeded to fit the models into the real data. However, despite continuing with our approach to fit the models to actual data, this necessitated of reevaluating the design of these models.

The Naive Model emerged as the winner due to limitations in our data's ability to capture the interaction between the response and stimulus x moving direction. This limitation was influenced by the size of our dataset and the dual roles of experimenters as participants (for more details, check the third question).

Which other insights does your model provide? Base your answer on the parameters fits of the winning model. (1 point) Our winning model, the Naive Model, demonstrated good convergence as indicated by the Gelman-Rubin Statistic (r_hat) values, close to 1 for all parameters. This suggests that the Markov chains have likely converged well, so it enhances our confidence in the reliability of the parameter estimates.

According to this model, we found that our model has precise estimates, based on the narrow HDI. It indicates more reliable estimates due to the higher ESS and Lower MCSE values suggesting higher precision in estimation. As we used MCMC and our dataset is small there was high variation, but our winning model achieved a low standard deviation, which indicates a low variation.

Since the naive model is a subset of our hypothesis-related models (you can always just set the beta parameters to zero, resulting in the naive model), they should be just as likely candidates for generating the data as the naive one. So we contribute the better results of the naive model to the fewer parameters it contains. This can result in an easier optimization problem when doing parameter fitting. More importantly, the elpd value takes the number of parameters into account, since according to Occam's razor models with fewer parameters should be preferred.

What are the potential weaknesses of your modeling study? (0.5 points) One potential weakness of our modeling study lies in the representation of the behavioral data. Contrary to our expectations, the behavioral data

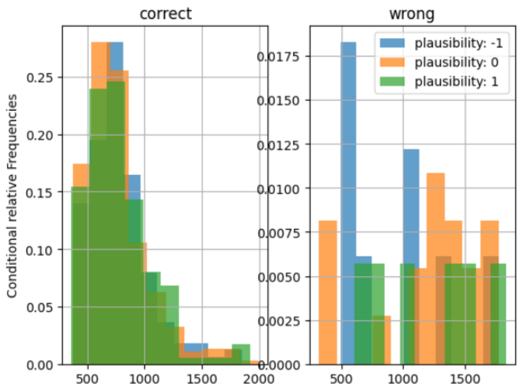

Figure 3: Behavioural data of our Experiment over eight participants. The depicted conditions are 1. Raindrop stimuli move mostly downwards (plausibility=1, 2. Raindrop stimuli are moving mostly upwards (plausibility=-1) and 3. Neutral stimuli are moving either up or down (plausibility=0).

did not corroborate the hypotheses we aimed to investigate. This discrepancy is evident in the fig (3). This could indicate limitations in either the experimental design or the data collection process, potentially affecting the validity of the model's predictions and its applicability to real-world scenarios."

Another potential limitation could come from the model implementation. The approach we took in developing and applying the model might have been flawed, possibly due to incorrect assumptions or technical errors in the coding and execution of the model. This could have led to inaccurate representations of the underlying processes we intended to model.

Additionally, in addressing the limitations of DDM models in capturing certain phenomena, it's important to acknowledge these models may face challenges in adequately representing the underlying processes when confronted with categorical data. Moreover, the simulation approach used for parameter recovery and model recovery can introduce its own biases. The simulated data using the HSSM toolbox showed a disproportionate number of responses with a value of 1 compared to -1. This imbalance inherently biases the model towards predicting a certain outcome. Consequently, the model might be less adept at predicting less frequent outcomes, and potentially overlooking subtler decision-making phenomena.

What might be another computational modeling approach for gaining a deeper understanding of the phenomenon? (0.5 points) Neural Networks (NNs) can provide deeper insights into decision-making phenomena. NNs can better encode participants' prior knowledge and response biases, allowing for a more comprehensive understanding of decision-making dynamics. They can learn intricate correlations between elements and decision outcomes, allowing for a more sophisticated understanding of small changes in decision-making behavior ((Taherdoost, 2023)).

Acknowledgements

Organizations: Hyerim Hwang, Milad Rouygari

Preparing workflow: Zeynep Bolluk, Leonard Frommelt, Hyerim Hwang, Rowan Mohamed, Milad Rouygari

Designing experiment and Hypothesis & Refining computational model: Zeynep Bolluk, Krishnendu Bose, Leonard Frommelt, Hyerim Hwang, Partha Pratim Kalita, Rowan Mohamed, Milad Rouygari

Implementing experiment & Building main computational model : Leonard Frommelt, Milad Rouygari

Collecting data & Implementing the dataset : Zeynep Bolluk, Rowan Mohamed, Milad Rouygari

Literature Review : Leonard Frommelt, Hardika Potey

2. Model implementation

Model Implementation : Leonard Frommelt

Model Simulation : Leonard Frommelt, Milad Rouygari

Parameter Fitting : Zeynep Bolluk , Rowan Mohamed

Parameter Recovery : Partha Pratim Kalita(coding & writing), Hardika Potey(writing)

Model Recovery: Milad Rouygari

Model Comparison : Krishnendu Bose, Hyerim Hwang

3. Writing the paper

Introduction & Method : Leonard Frommelt, Hardika Potey

Hypothesis & Description of Computational model : Leonard Frommelt

Description of Experiment : Milad Rouygari

Model Description and Results : Same as indicated in model implementation section

References

Gupta, A., Bansal, R., Alashwal, H., Kacar, A. S., Balci, F., & Moustafa, A. A. (2022). Neural substrates of the drift-diffusion model in brain disorders. *Frontiers in Computational Neuroscience*, *15*, 678232.

Horinouchi, T., Watanabe, T., Matsumoto, T., Yunoki, K., Kuwabara, T., Ito, K., ... Kirimoto, H. (2022). The effect of prior knowledge of color on reaction time depends on visual modality. *Heliyon*, *8*(5).

Leite, F. P., & Ratcliff, R. (2011). What cognitive processes drive response biases? a diffusion model analysis. *Judgment and Decision Making*, *6*(7), 651–687.

Pomerol, J.-C., Brézillon, P., & Pasquier, L. (2002). Operational knowledge representation for practical decision-making. *Journal of Management Information Systems*, *18*(4), 101–115.

Simen, P., Contreras, D., Buck, C., Hu, P., Holmes, P., & Cohen, J. D. (2009). Reward rate optimization in two-alternative decision making: empirical tests of theoretical predictions. *Journal of Experimental Psychology: Human Perception and Performance*, *35*(6), 1865.

Taherdoost, H. (2023). Deep learning and neural networks: Decision-making implications. *Symmetry*, *15*(9), 1723.

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and computing*, *27*, 1413–1432.