Model 1: Anchor and Adjustment

Model formulation based on:

Lieder, F., Griffiths, T. L., M. Huys, Q. J., & Goodman, N. D. (2018). The anchoring bias reflects rational use of cognitive resources. Psychonomic Bulletin & Review, 25(1), 322–349. https://doi.org/10.3758/s13423-017-1286-8

Notations used in the model:

[I am not sure if I should model x- and y-axis separately. Here, I formulate the model in $\mathbb{R}^2$.]

1. Parameters that should be given:
   - Anchor $a \in \mathbb{R}^2$: initial guess;
   - $P(X|K)$: people's probabilistic belief about X given their knowledge K (Here, we can understand it as the probability distribution of the obstacle being at different locations on the canvas if no anchor was present), should be modeled as a 2-d Gaussian Distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$, and use the empirical estimations of μ and $\mathbf{\Sigma}$.
     [This should be obtained before the simulation; currently, we don't have this information]
   - $\mathcal{H}$: the hypothesis space should contain all evenly spaced values in the range spanned by the values in the belief distribution $P(X|K)$ and the anchors $\pm$ one standard deviation. (I see this as a evenly divided grid on the canvas with each cell's coordinate representing a hypothesis).
2. Parameters that will be estimated after fitting the model to the data:
   - $P_{\text{prop}}(\delta)$: the proposal distribution to model the size of adjustment in each step, commonly modeled as $P(\delta) = \text{Poisson}(|\delta|; \mu_{\text{prop}})$. Here, $\delta \in \mathbb{R}$ could be the Euclidean distance between two hypothesis, thus we have $P(\delta = \text{distance}(h_k - h_j)) = \text{Poisson}(|k - j|; \mu_{\text{prop}})$, where h_k and h_j are the k^{th} and the j^{th} value in the hypothesis space $\mathcal{H}$, and μ_{prop} is the expected step size (which should be estimated from fitting the data).
     [How to label hypothesis arranged in a 2-d grid? The original methods worked with 1-d data so using $|k - j|$ makes sense. If denote the column, row index of the k^{th} and the j^{th} as k_m, k_n and j_m, j_n, could we do $|k_m - j_m| + |k_n - j_n|$?]
   - $\delta \in \mathbb{R}$: size of adjustment in each step, sampling from the symmetric probability distribution proposed above $P_{\text{prop}}(\delta \sim P_{\text{prop}})$.
   - t: the number of adjustment.

Model fitting process:

[This is the part that I am most unsure about.]

adjustment = relative adjustment * distance(anchor, posterior expectation)

To fit the relative adjustment for each stimulus with a specific anchor ([**or, should we model individual participant's responses?**]), we first calculate the posterior expectation. It should be the center of the 2-d Gaussian Distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$ calculated based on all participants' responses for a particular stimulus with a specific anchor.

We can then do a grid search ([**?**]) of μ_{prop} and t the number of adjustment:

for each value of μ_{prop}:
    for each value of t:
        for each iteration i in range(t):

$\hat{x}_i$ = current guess of quantity X after t adjustments ($x_0 = a$)
sample δ from $P(\delta) = \text{Poisson}(|\delta|; \mu_{\text{prop}})$
if $P(X = \hat{x}_i + \delta | k) > P(X = \hat{x}_i | k)$:
$\hat{x}_{i+1} = \hat{x}_i + \delta$
else: accept with probability $\alpha = \frac{P(X=\hat{x}_i+\delta|k)}{P(X=\hat{x}_i|k)}$
return $\hat{x}_t$

Find the $\hat{x}_t$ that returns the closest relative adjustment and record μ_{prop} and t.

Model 2: Path-Projection

Model formulation based on:

Sosa, F. A., Gershman, S. J., Ullman, T. D. (2023). Blending simulation and abstraction for physical reasoning.

$$s_t = f(s_{t-1}; D, N, E) = \begin{cases} \pi(s_{t-1}; N) & \text{if } \epsilon < E \\ A(s_{t-1}; D) & \text{if } \epsilon > E \end{cases}$$

where $\epsilon = S_c(\pi(s_{t-1}; N), A(s_{t-1}; D))$, and $A(s_{t-1}; D)$ computes s_t by projecting the position of of the ball p^B_{t-1} some distance D along the direction of the ball's velocity v^B_{t-1}.

Notations used in the model:

1. Parameters that should be given:
   - π: pure simulation of the physics engine
   - N: number of forward steps the engine performs to change the state
2. Parameters that will be estimated after fitting the model to the data:
   - D: distance skipped by abstraction A
   - E: threshold to determine the the choice of simulation/abstraction
   - k, j: time points that the model switches from abstraction $\rightarrow$ simulation $\rightarrow$ abstraction. We fix:
     - time point k such that all time points before k satisfy $\epsilon < E$ (abstraction)
     - time point j ($j \geq k$) such that all time points after j satisfy $\epsilon < E$ (abstraction)
     - so for all $k+1, k+2, ..., j-1$ time points, $\epsilon < E$
     - reasons to estimate k, j: participants are guessing when will the collision happen, so they are also testing out k, j to align with their observation

Model fitting process:

[**I would appreciate any feedback you have on it.**]

We want the model to predict the center of the 2-d Gaussian Distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$ calculated based on all participants' responses for a particular stimulus. [**But then, how could we incorporate anchor into this model?**]

Assume that, at time m, the ball falls out of the screen ($m > k, j$). Thus, the state of the ball at time m is P^B_m.

Below will be the procedure for a particular set of D, E, k, j [**I think there are too many parameters to be estimated after fitting the data, but I also find it hard to eliminate any one of them**]:

for each time step i in range(m):
    if $i < k$: $s_i = A(s_{i-1}; D)$
    else:
        sample (x, y) (the position of the obstacle) from a distribution*
        run simulation based on (x, y): $s_i = \pi(s_{i-1}; N)$
        if $S_c(P^B_{j+1}, P^B_m) < E_0$ (if at time $j+1$, P^B_{j+1} is similar enough to P^B_m; notice that $P^B_{j+1} \in A(s_j; D)$):
                accept (x, y) as a model prediction
        else: resample (x, y)
    return (x, y)

*[**maybe** $(X,Y) \sim Unif([p^B_{k\ x} \pm obs_{radius}], [p^B_{k\ y}, p^B_{my}])$]

If there is no (x, y) that satisfies the criteria, change the values of D, E, k, j. E, E_0 here could take the same value, since E_0 is also a threshold for the similarity.

[I also think that the value of D might be less important here, since we are interested in the shift from abstract to simulation and to abstract and D can vary as long as we obtain the values of k, j.]