A Bayesian Analysis on Hit Error in *osu!*

Sage Li

Georgia Institute of Technology

Janurary 16, 2025

Abstract

In this paper, we construct a hierarchical Bayesian model of hit error in *osu!*, a popular game that incorporates aspects of rhythm and aim in a two-dimensional plane. In particular, we wish to investigate the relationship between a few parameters, such as player skill, and object distance, on the distribution of attempted hits on an object. A Bayesian hierarchical model is constructed for its flexibility in modeling complex nested random effects and strength even with limited data.

Keywords: Bayesian, Hierarchical, *osu!*

1 Introduction

1.1 Background and Terminology

Created in 2007 by Dean "peppy" Herbert, *osu!* is a free-to-play rhythm game inspired by the *osu!* Tatakae! Ouendan series of games published by Nintendo. The game incorporates aspects of a traditional rhythm-based title and places importance on aim within a two-dimensional playfield.

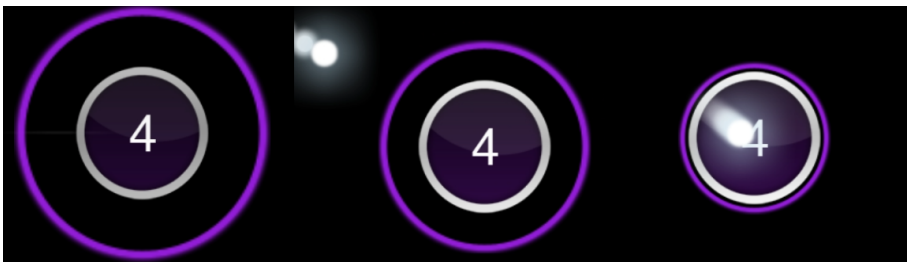

Figure 1: A sample of *osu!* gameplay.

The gameplay of *osu!* consists of a series of consecutive "hitcircles" appearing, typically mapped by a "mapper" in sync to a backing track, on a 512 tall by 384 "game-pixel" wide playfield, where the pixel size of a game pixel is adjusted accordingly to a user's display resolution. A moment before an intended hit on a hitcircle, an outer "approach circle"

appears and converges, then subsequently overlaps with the border of the hitcircle, at which time a player is intended to move their cursor over the hitcircle, and "tap" with an input, typically a computer keyboard or mouse click. There are other objects such as "sliders" and "spinners," but they are generally irrelevant to this study. A player's accuracy in timing a tap is rated in a few windows- with the maximum unweighted score achievable as a 300, with subsequent 100 and 50 timing windows. Any tap outside of a 50 timing window is rated as a miss. A player's accuracy on a play is calculated by the ratio of achieved unweighted score to total possible unweighted score on a specific beatmap.

Figure 2: Objects are placed in patterns by mappers to represent the music

Mappers have creative freedom to put objects in various patterns to best represent a backing track. Naturally, some patterns are more difficult than others, with some significant considerations being object spacing, the map's beats per minute (BPM), and the shape of the surrounding hitcircles. A common style of pattern referenced in this study is called a "jump"— a series of consecutive, highly spaced hitcircles which require a high degree of aiming dexterity. The difficulty of the map is computed with a built-in algorithm as a "star rating." Although the star-rating system is controversial in the community for being inaccurate in many cases, it is generally consistent within the same style of map.

As players continue to play the *osu!*, their own skill naturally increases- and a leaderboard system based on "performance points," commonly abbreviated as "pp", is implemented to attempt to quantify individual skill. Players gain these points through a built-in algorithm, considering a player's performance on a map and the duration and difficulty of the map. Note that while performance point reward and map star rating are generally correlated, the exact values do not necessarily correspond one-to-one. A player's total performance point rating is computed by aggregating the player's top performance scores, and weighing them with their higher scores more heavily.

For reference, a chart of global rankings (as of November 24, 2024), and their corresponding performance points is provided below.

Global Ranking	Performance Points
1	$30,252$
10	$21,513$
100	$17,485$
$1,000$	$12,953$
$10,000$	$8,747$
$100,000$	$\approx 4,730$
$1,000,000$	≈ 620

For more in-depth information on anything related to gameplay, please refer to the well-maintained community wiki [3].

1.2 Significance and Objectives

In this study, we focus on the aiming aspect of *osu!*, particularly the distribution of error between the center of a hitcircle and a player's attempted hit. Previous attempts to model hit distributions have relied on frequentist techniques, but here we aim to produce a more robust and flexible model using a Bayesian hierarchical approach. Bayesian methods have a few notable advantages: given properly informed priors, they often yield more stable estimates with smaller datasets and allow for the direct probabilistic interpretation of model parameters [2].

2 Method

For some map $\mathbf{M}$ consisting of n circles, we construct the time series

$$\{m_1, m_2, \cdots, m_n\} \in \mathbf{M},$$

where

$$m_i \equiv (x_i, y_i)$$

where (x_i, y_i) is the game-pixel location of the i^{th} hitcircle to appear in a map. Let jump distance $\mathbf{D}$ be the series

$$\{0, d_1, d_2, \cdots, d_n\} \in \mathbf{D}$$

$$d_i = \begin{cases} ||m_i - m_{i-1}||_2, & \text{if } x \neq 1 \\ 0, & \text{if } x = 1 \end{cases}$$

where

$$||m_i - m_{i-1}||_2 = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

is the 2-norm, or euclidean distance between successive objects.

We choose maps with relatively consistent timings between circles to minimize the effect of the cursor speed required for successive jumps.

For a given player replay, we can extract attempted hits from the replay file using a modified version of a tablet optimization program. Each attempted hit is associated with its corresponding hitcircle position, forming a series of hit attempts h_i

$$\{h_1, h_2, \cdots, h_n\} \in \mathbf{H}.$$

Taking the 2-norm between $h_n \in \mathbf{H}$ and $m_n \in \mathbf{M}$ gives us observations

$$\{e_1, e_2, \cdots, e_n\} \in \mathbf{E},$$

where

$$e_i = ||h_i - j_i||_2,$$

which we call **hit errors**.

We now begin construction of our hierarchical Bayesian model. From [1], the distribution of the 2-norm of a two-dimensional Gaussian is gamma distributed, so we assume a gamma prior on hit error e

$$e_{ij} \sim \text{Gamma}(\mu_{ij}, \beta)$$

player i, and hit number j, where rate parameter β has a prior

$$\beta \sim \text{Gamma}(2, 0.1)$$

and shape parameter μ_{ij}

$$\mu_{ij} = \alpha_0 + \alpha_1 pp_i + \alpha_2 d_{ij} + \epsilon_i$$

with the j^{th} player's total performance points pp_j, jump distance d_{ijk} corresponding to the hit, star rating of the map sr_i,

where all coefficients α_n have prior distributions,

$$\begin{aligned}\alpha_0 &= \text{Normal}(0, 10)\\ \alpha_1 &= \text{Normal}(200, 10)\\ \alpha_2 &= \text{Normal}(0, .5)\end{aligned}$$

player-level intercept parameter ϵ has prior,

$$\epsilon_i \sim \text{Normal}(0, \tau)$$
$$\tau \sim \text{Half-Normal}(\sigma = 1)$$

The model is created in PyMC and can be run in the attached Jupyter notebook. Data are collected from various volunteers playing the maps "squartatrice vs. disperagioia", and "my love life needs a lobotomy" of various difficulties. 4000 hits sampled randomly throughout each map are used as a training set. Player performance points and jump distances are normalized to increase the performance of the model.

3 Results

1000 tuning samples and 2000 draws were computed on 16 cores to produce a total of 36000 total draws. The results of the model are plotted in figures 1 and 2 for α_n.

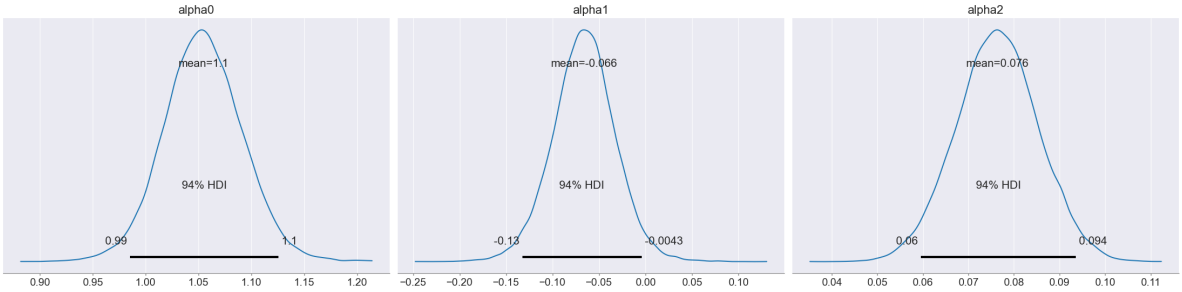

Figure 3: Posteriors of coefficients α

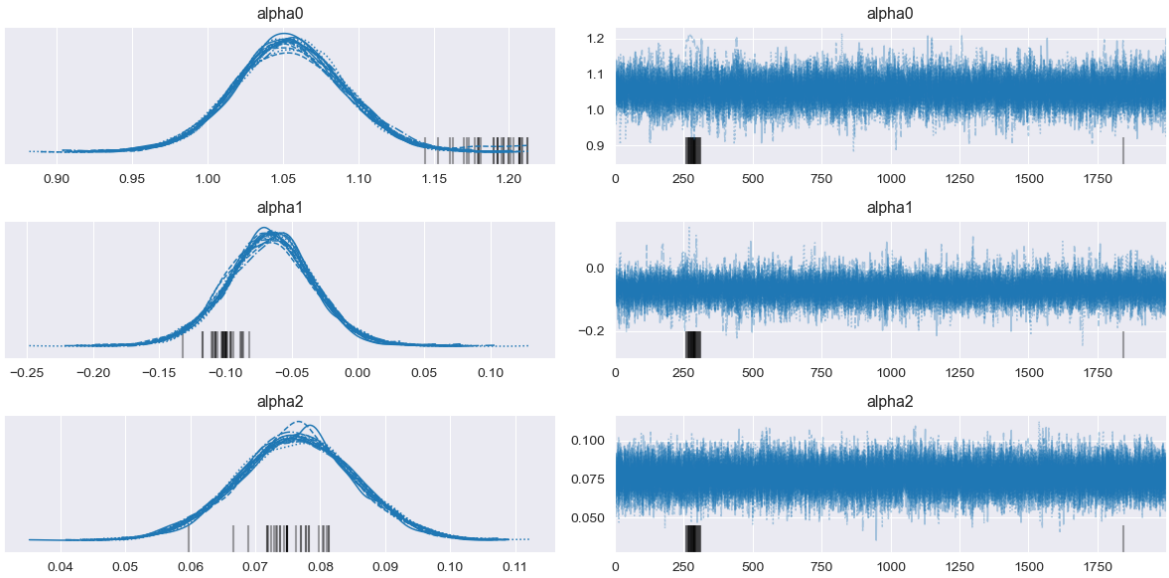

Figure 4: Trace plots of coefficients α

Here are some statistics on the distributions of the various parameters and hyperparameters. Full statistics can be viewed in the accompanying Jupyter notebook.

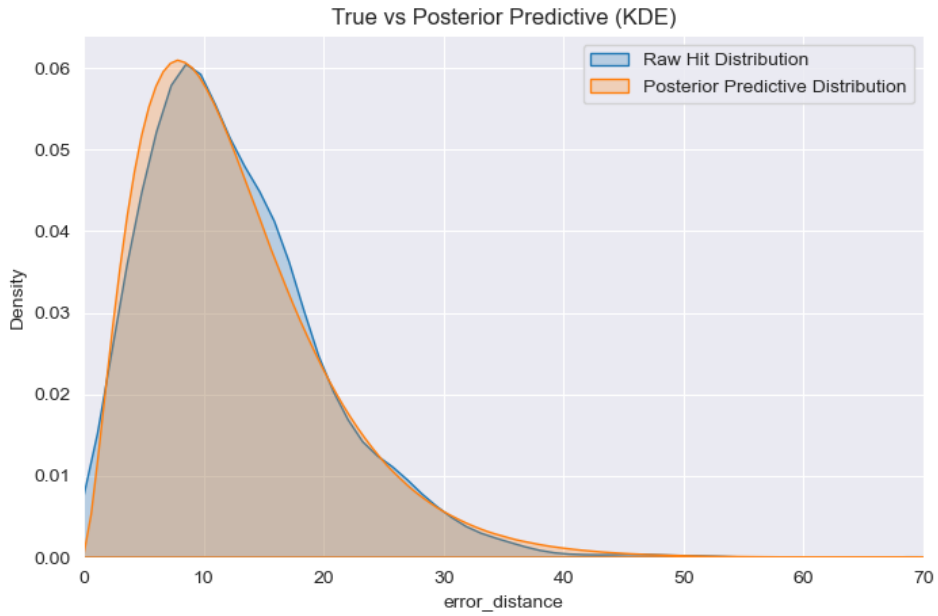

Figure 5: True hit error distribution compared to predictive posterior distribution given by the model.

Parameter	Mean	Std. Dev.	2.5% HDI	97.5% HDI
α_0	1.055	0.038	0.985	1.126
α_1	-0.066	0.034	-0.133	-0.004
α_2	0.076	0.009	0.060	0.094
τ	0.102	0.029	0.055	0.154
β	0.214	0.005	0.205	0.223

Samples were drawn from the model to form a posterior predictive distribution, and the result is plotted against the raw hit error distribution of our data shown in figure 5.

4 Discussion

The mean of a gamma distribution Gamma(μ, β) is

$$E(x) = \frac{\mu}{\beta} \Rightarrow E(x) \propto \mu$$

We can see that α_0, which corresponds to a global intercept value of the distribution of the mean, is quite confidently close to 1. Seeing additionally that β is approximated by 0.2, we can interpret the model as the following: a player with 0 performance points on a jump of 0 distance will have a mean hit error of around 5 playfield pixels. We expect that this parameter will be much more refined with more data.

The parameter corresponding to the impact of standardized performance points, α_1, is negative at 95% confidence— a negative α_1 can be interpreted as increasing performance points of a player decreasing the shape parameter of the distribution. Recall that $E(x) \propto$

μ, so the mean of the the player's hit error distribution is decreased accordingly. This is expected; we would expect that as players play the game more and more and their performance point value increases, their coordination between their input device and the game improves, and the player's hit mean hit error decreases.

The parameter corresponding to the impact of jump distance on error, α_2, is positive at a 95% confidence. The shape parameter of our hit error distribution is increased with larger jumps, and subsequently the mean is increased. We expect that larger jumps require more dexterity to accurately aim, which leads to increased mean hit errors.

5 Conclusion and Future Work

A major limitation of this report was the computational power we had available. The size of the data were relatively small, with only 14 unique players and 4000 total hits— even then, sampling 48,000 total draws with a 16-core CPU took the order of 10 minutes or so per attempt. With more powerful computing resources, it would be possible to gather data on many more players and maps to provide more precision. Although Bayesian models perform well even with limited data, consideration of more map types would lead to a more holistic model.

Future work could include individual map attributes: even within the same set, maps have attributes specified by the mapper such as approach circle rate, circle size. Including these in the model could produce a more holistic approach, though at the cost of some possible co-linearity.

Distance between objects is accounted for as a model parameter, but not the timing between each of the jumps. With this, two objects separated by a long break are interpreted identically to a faster "jump". We attempted to mitigate this by picking maps with relatively consistent 1/4 beat patterns throughout the map. Integrating a parameter proportional to the ratio of jump distance and jump time, equivalent to something like "jump speed," could serve as a potential solution. Additionally, jump distance between two axes could be separated- due to the mechanics of most aiming methods used in *osu!*, the vertical axes is often considered to be more easily traversed.

We have produced a more flexible model for hit distributions in *osu!*; however, much future work is needed to further refine the model. We hope that this paper shows the potential advantages of Bayesian statistics over classical statistics and inspires future work on the topic.

Acknowledgments

We would like to thank Eathan "Shauns" Xu for his guidance in creating the hit analysis script.

References

[1] Benjamin Thirey, Randal Hickman. Distribution of euclidean distances between randomly distributed gaussian points in n-space. https://arxiv.org/abs/1508.02238. Accessed: November 30, 2024.

[2] Isabella Fornacon-Wood et. al. Understanding the differences between bayesian and frequentist statistics. *International Journal of Radiation Oncology, Biology, Physics*, 2021.

[3] osu!wiki contributors. osu!wiki. https://osu.ppy.sh/wiki/en/Main_page. Accessed: November 30, 2024.