Estimating Difficulty in osu! with Sequential Models

Ben Wonderlin, advised by James Glenn, Yale University

Overview

- Osu! (stylized "osu!") is a competitive, single-player rhythm game for PC.
- Despite its popularity, the algorithm that osu! uses to estimate difficulty – and in turn, compute its leaderboards – is somewhat naive.
- **This project aims to address the current algorithm's weaknesses by training a sequential machine-learning model on a large amount of replay data.**

Background

- osu! is played by clicking circles in time to the beat of a song. The goal is to click the circles with a high degree of rhythmic accuracy.
- There are four possible outcomes for each note. They range from a "300" (for a perfectly timed click) to a "miss" (for no click at all).
- osu!'s current difficulty estimation algorithm is a hand-built heuristic that considers basic note attributes, such as the time and distance between notes. As such, it fails to accurately estimate the difficulty of unusual note patterns.

Fig 1: osu! gameplay (cursor in upper right)

Methodology

- I obtained a dataset of 381,000 replays from a third-party website that renders replays to video. I then parsed the replays into sequences of (note, outcome) pairs.
- I used the sequences to support a multi-class classification task. I considered three architectures: a naive, feed-forward network, an LSTM sequence classifier, and an LSTM sequence-to-sequence translator.
- For each model, the objective was to predict a note's outcome given its attributes and predecessors.

Methodology (cont.)

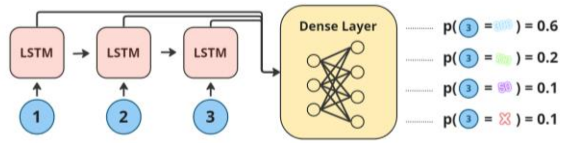

Fig 2: Model diagram of the LSTM sequence classifier

- Finally, I estimated the difficulty of a song by computing the joint probability of clicking all of its notes, according to the model's classification probabilities. In the naïve case, in which each note outcome X_i is independent, this is given by

$$-\log \mathbf{p}(\text{no misses}) \approx -\sum_{i=0}^{n} \log\left(1 - \tilde{\mathbf{p}}(X_i = \text{miss})\right)$$

Results

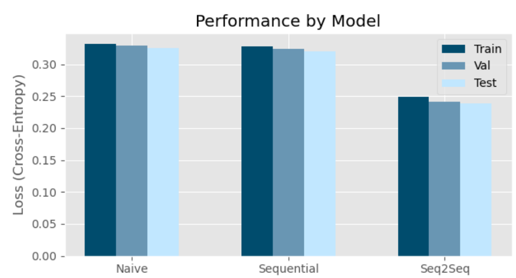

Fig 3: Classification performance by model and dataset (note-level)

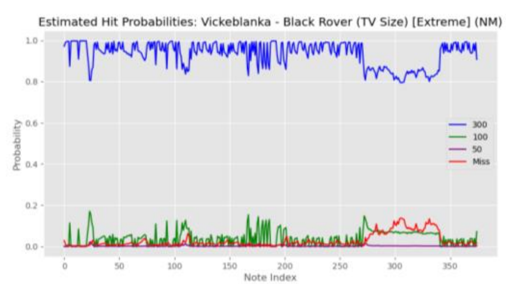

Fig 4: Predicted outcome probabilities over the course of a song with a challenging ending

Results (cont.)

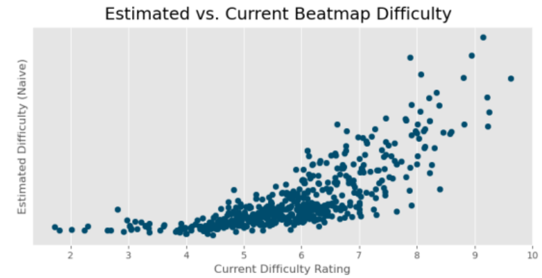

Fig 5: Comparison between my song-level difficulty estimations and those of the current algorithm

Main Findings

- **Machine-learning models accurately predict note outcomes. Sequence-to-sequence models are particularly accurate (though further sensitivity analysis is warranted).**
- **Estimating the difficulty of a song via the joint probability of clicking all of its notes results in an algorithm that closely tracks the current algorithm.**
- **A representative dataset is necessary for developing a viable data-driven algorithm. The current dataset does not contain enough poor replays, resulting in condensed difficulty estimations.**

Acknowledgments

I would like to thank James Glenn for supporting this project. I would also like to thank Liam Devoe, a graduate student at Northeastern, for supporting my efforts to parse 29 GB of replay files. I would not have accomplished much if I had to write my own parser from scratch.

References

I Gold, K., & Olivier, A. (2010). Using Machine Translation to Convert Between Difficulties in Rhythm Games. Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 6(1), 27-32. https://doi.org/10.1609/aiide.v6i1.12396