Rob wants to make some fun, interesting, and maybe even useful models from Killer Queen log data.

What does equity have to do with Killer Queen? Imagine you get paid a dollar for winning, and $0 for losing. Then your equity is your win probability. Make cents?

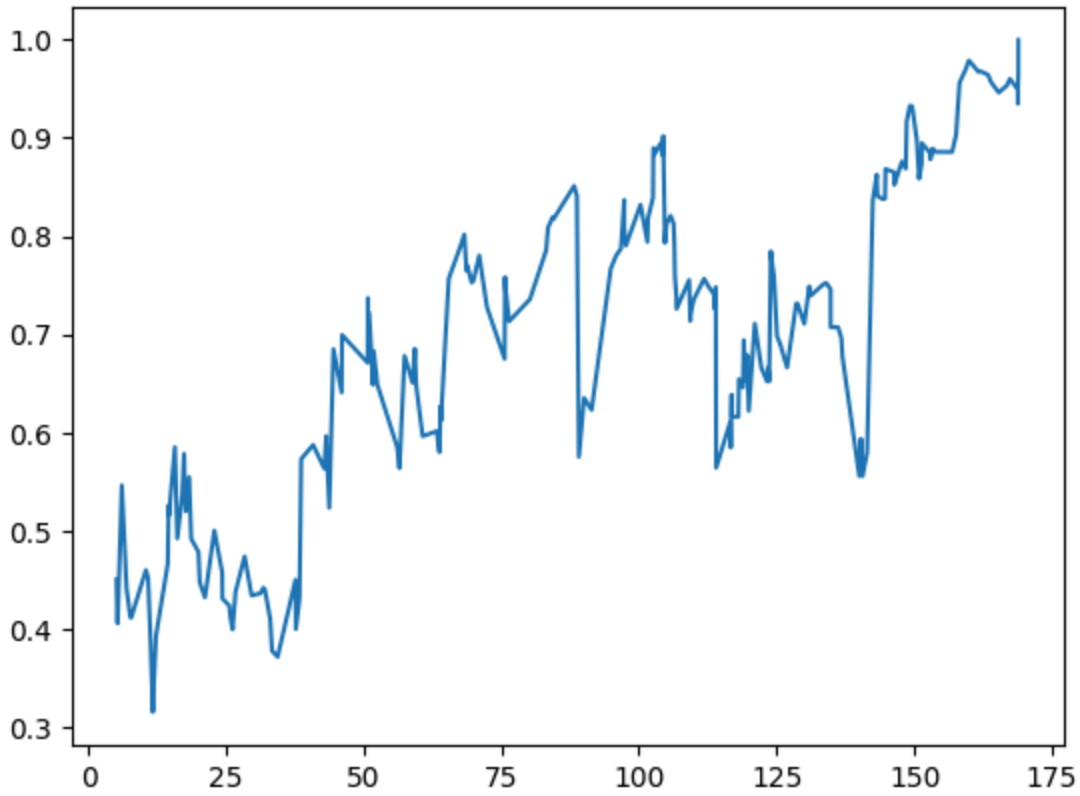
Background

Abby provided me with a hivemind data dump that includes all gameevents, hivemind logins, and game locations from 2019 until 2-27-2023. There are about 200k games logged. Data before version 17.26 is messy and annoying but there are about 90k games since Sept 2022 using 17.26 that pass validation/sanity checks. The data format is documented on the HiveMind wiki. It tracks important game events, player kills/deaths, warrior forms, snail positions when riders jump on or leave, and berry deposits. It is what is used to make the end game screens by HiveMind.

Candidate models

Equity model

This model predicts the probability of winning (henceforth equity) from a given game state. SumoLogic has built an equity model here using the same game log API, but used an inhouse data set from non-travel tournament games.



Here is an equity/live win probability (LWP) graph for the SF league night on 2/13/23, the hivemind data is here hivemind game 637156. This was built from a LightGBM classifier and was not in the training data. The big spike at ~140 corresponds to 45:51 which ends at an equity of around .88, which is a queen kill and then a 2x warrior kill which leaves blue with 2vs0 eggs advantage, a 2 speed warriors, 2/3 of the warrior gates, snail out of the tunnel, snail control and gold is wiped. Seems reasonable? Maybe .88 is even too low?

Vectorized game state models

These models would take as input a vectorized summary of the game state. All of the following are features computable from the game event logs. The model wouldn't know the position and velocity of players at any instant, but otherwise knows the important factors in the game.

Current game state features

- Map name
- Queen eggs
- Snail position. But we only have snail on/off events, use interpolation/guessing to figure out snail position if there is a rider.
- Deposited berry counts (split high vs low berries for dusk?)
- Berry availability (might be some trickiness handling famine)

- Warrior counts
- Which characters have speed
- Snail control
- Gate control
- Hivemind player IDs (sparse)

Game history features

In A Bayesian approach to in game win probability, the paper discusses contextual features for Rugby prediction, like goal scoring opportunities and successful attacking passes. They are events that might indicate a team is performing well even if it is not ahead. Rough analogs in Killer Queen might be things like warrior glances on the queen, clears on riders who are eating, kills on ledge guards, or successful pincers.

The label for the equity model is the winning team.

From these features/labels, there are many different kinds of models that could be built.

Logistic regression on a vectorized game state summary is what SumoLogic built, which seems like a reasonable place to start.

A gradient boosted tree or random forest on the vectorized game state would probably be a bit stronger and allow capturing some non-linear interaction effects.

A multilevel neural net would also allow capturing non-linear interaction effects like the tree models, and have the advantage of being able to share representations and leverage other tasks, like an next event model (discussed later).

Also try tabnet? https://github.com/dreamquark-ai/tabnet

Also try transformer language model with on game event logs with game states/command based tasks littered in? Conjoin taskspecific terms with buckets to embed a regression problem into an LM.

Applications of the Equity model

A graph of equity of time would make an interesting end of game display for hivemind.

Take the equity over time graph, and find small time spans with big changes in equity. These correspond to impactful moments in the game, and would make for good automatic highlights.

With timestamps and local recording set up, we could even extract/display these on a post game screen.

Assign credit/blame to players for the relevant subset of events (kills, berry deposits, snail progress, gate tags) and sum the change in equity per player, we could compute an impact measure for each player in the game. Of course, credit assignment here is itself a bit wonky/flawed, clearly snail progress is a function of the whole team and not just the rider. Most kills can be credited to multiple cooperating teammates, but our assignment would fail to do that. It still might be a good/interesting measure and end of game display.

Next event prediction

Build a model to predict the next event from a given game state. This model takes the same input format as the equity model, but has a much richer label set, and many more labels per game. If it was very good, it could have applications like predicting the chance that a team in lock out forms a warrior before their queen loses an egg. In practice, I expect it would simply help as a form of self supervised learning, since there are many more events per game than there are game results per game. For the Baltimore Brawl 5 data set, there is an average of 318 events for a single game. Training a dual task model on next event prediction will help the network form better representations, which could help mitigate data sparsity issues in equity prediction.

Future game state prediction

Given a game state, after N (1-5?) seconds, what is going to be the state of the game. This is like next event prediction, but perhaps it's a bit cleaner of a target, since it depends less on the exact order of events, and more on the game trend. This will be another self supervised task. It rewards a model with representations that can accurately predict things like the expected rate of change of important factors in the game like berry counts, snail progress, warrior counts, queen eggs, etc.

Combat rating

Each player will get 4 (5?) numbers. Queen, vanilla warrior, speed warrior, drone (speed drone?), which models the players ability to win fights.

Treat each kill and death as a win or loss vs opposing players. Apply a rating system like trueskill or elo to kill death interactions on a per player/position basis.

Should pincerers be rewarded? How should drones, which lack a direct ability to kill, be given some credit for being resilient or even offensive?

Give bump assist credits to reward drones and the bumper side of a pincer include them in the calculation. Perhaps consider drones bumping warriors or queens as a downweighted tie. Perhaps say, 10 bumps is treated as a kill and a death. Perhaps this should be limited to drones who have recently touched a snail, or bumps after a snail jump off should count more than bumps anywhere on the map?

Snail inertia

This model intends to measure how hard it is to move the snail from a given spot on the map. How likely is the rider to die, make progress, or jump off? Jess had built a model that estimates the changes in equity as a function of incremental snail progress. (Add her graph here?). That would be part of our equity model. This rather focuses on the difficulty of moving the snail, rather than its value.

Model validation metric

We'd like our model to work well at predicting future killer queen tournament games.

Leave one tournament out cross validation using log loss of the equity model captures that intent pretty well, though it does have the advantage of being able to train on future tournament games when evaluating tournaments in the middle. We should probably be careful to weight tournaments in proportion to their size, rather than naively averaging the loss per split.

An alternative might be to use a progressive, time aware evaluation scheme that never has training data from the future, but it seems a bit more complicated.

Data augmentation

We don't really have much data. Aside from getting more data, here are some ideas for squeezing out more juice from what we have.

Every game has a symmetric game where blue and gold are swapped, which could double the training data size, or form a free evaluation set, or be useful in a loss that penalizes differences in evaluation between a state and its team swapped symmetric state.

Imagine any game state S from the perspective of the gold team. As gold gets more speed upgrades, more gates, more berries, more queen kills, or more snail progress, we'd expect that their equity would improve. Call this improved state S'. An augmented example pair ranking loss that penalizes the model for scoring S below S' would be a way of adding this monotonicity prior into the model. A last egg is more valuable than a first egg, so a state where one queen is

put on last in exchange for a first egg usually benefits the team with more eggs more. Similarly, adding a berry to each team tends to benefit the team with a berry lead.