A multi-fidelity analysis of CS:GO skill rating systems

Mikel Bober, Naunidh Dua & Max McGuinness
Machine Learning and the Physical World • Michaelmas Term 2022

Background: CS:GO (Counter-Strike: Global Offensive) is a tactical first-person shooter game, with a large competitive scene based on playing other teams of a similar skill. Matches are played 5 versus 5, and best of 30 rounds (draws can happen depending on the tournament organiser). Crucial to competitive games like CS:GO is the skill rating system which determines who to pit against each other.

In this project, we will analyse different ranking systems as emulators, along with acquisition functions (choosing which teams to pit against each other) which allows for estimates of players/teams skill to be accurately ascertained in as few matches as possible. The underlying dataset will be taken from professional CS:GO matches, of which thousands have the results posted on HLTV.org[1]. Using this data as a simulator will allow us to determine the validity of each approach without needing to play thousands of real matches.

Emulator: We will build an *emulator* of the outcome of matches between two teams. Existing systems such as TrueSkill[2] and Glicko[3] do exactly this - by assigning each team a Gaussian skill rating, one can *emulate* a match by calculating an estimated win probability between the two teams. We can model the collection of skill ratings of all teams with Gaussian Processes.

In Trueskill, whilst there is no covariance between players, there are factor graphs where the skill of players depend on other players. One can imagine rating systems which incorporate covariance into the GP (i.e. if two teams play against each other a lot, the covariance between them will be low, but may be high between two teams that have never played together).

Simulation: Of course, we cannot sample new matches to fit the emulator, and producing a higher-fidelity simulator of matches between players is difficult. In place of our simulator, we will use an oracle which returns the outcome of a recent real match between the selected teams.

An acquisition function is employed to select matches for sampling (albeit over a domain limited by our choice of simulator). This function is necessitated by the high cost of real-world sampling, and is designed to maximise the amount of information gained from each match, performing a type of surrogate optimization. The acquisition function thus reflects the real-world selection process, while the simulator represents the matches themselves.

Evaluation: We will compare different emulators, acquisition functions and number of matches (sampled points) used for learning the skill ratings. When the underlying function is known, it is easy to evaluate the fit of a Gaussian Process; however, in our case there is no underlying skill value which we can compare against.

Instead, we will evaluate the learnt skill ratings using each approach on “future” matches i.e. the final subset of the dataset, and whether the skill model can predict these. By doing this, we can answer questions such as how many matches does it take to accurately fit the ratings, and how much randomness is inherent in a match up between two teams (versus which team has the higher skill).

---

[1] https://www.hltv.org/results
[2] https://trueskill.org/
[3] https://en.wikipedia.org/wiki/Glicko_rating_system