Accurately Ranking Players in a Tournament With

Three or More Players Per Game

Kyle Shelton, Jacob Bernard, Daniel Grube  
Taylor University

Abstract

Generating rankings for Artificial Intelligence (AI) playing multiplayer games presents many challenges that must be considered and sorted out. When there are many players and many players allowed per game, it may be difficult or infeasible to play each AI against every other possible set of AI “Round Robin” style to determine the relative skill or quality of the competitors. Even having performed such a round robin style competition, if a system does not take into account variance and head to head performance, it may not rank competitors in a way that makes sense and accurately reflects the data. We propose that using a condorcet method to evaluate the results from a relatively small number of matches can generate an accurate sensical ranking much more quickly than can other styles of competition.

Accurately Ranking Players in a Tournament With

Three or More Players Per Game

The problem of ranking a set of players based on performance has two main problem areas. The first is closely analysed to the issues with selecting an effective voting/election system. Fortunately, elections and voting systems have been studied closely, and there has been done much research and mathematical modeling to compare their merits and shortcomings. From this research have emerged many qualities that are desirable for systems of ranking among multiple parties, though it has been proven by Kenneth Arrow’s impossibility theorem (1951) that no system can demonstrate each of these qualities. One desirable quality that we prioritised was Condorcet selection. Condorcet selection states that if there exists a population of candidates *a* through *n* such that candidate *x* would triumph over each other candidate *a* through *n* in head to head matchups, candidate *x* should be selected as best. We decided to use a method that preserves this quality with the hopes that it would rank players fairly and effectively.   
 Another major problem area of ranking a pool of AI in multiplayer games is how difficult it might be to rank a pool of artificial competitors so large as to make running each possible combination of competitors one at a time infeasible. There has been little to no research in the area of evaluating AI created for multiplayer games because of the relatively modern nature of the need. The research that best approximates this problem area is that of ranking live competitors in multi-competitor sports, such as golf or online multiplayer games. Such research is useful because it requires that competitors be evaluated with relatively little information. These systems fall short of our problem space however, in that AI can be tested much more vigorously in pursuit of accurate ranking because of the relatively small amount of time it takes for an set of artificial competitors to execute a single match. Artificial Intelligence competitions can generate large numbers of competitions and data in a much faster time than can any competition involving human competitors. A system to rank AI should make use of their speed to generate more accurate evaluation.

Related Research

There is no research done specifically on AI competitions but there are similar ranking systems in place that are used to rank people as opposed to AI. One ranking system is TrueSkill which is explained in an article written by Herbrich, Minka, and Graepel (2007). TrueSkill is the ranking system used in Xbox Live matchmaking services. This system seeks to match competitors with those of equal skill and also takes into consideration the fact that people typically get better the longer they play and get worse the longer they take a break from a game. While this system is accurate, AI do not get better or worse over time so it was not necessary for our method to track that variable. Another ranking method that we looked into was developed by Mark Glickman and Jonathan Hennessy (2015). Their rating system worked essentially the same as TrueSkill in that it also accounted for players getting better or worse over time. While these ranking systems are both accurate, they attempt to solve a problem that is non-existent with our competitors. Since our competitors are AI, there is no need to measure if they get better or worse over time as all of the matches happen in a relatively short amount of time.

Model

In this research we created a system that would execute a number of matches each with competitors selected randomly from the competition pool. It then evaluates the results of these competitions exclusively in a head to head manner, such that each competitor is given a single win for every competitor that it beat in that match. After some predetermined number of random games are played, the AI are ranked based on how many competitors they managed to win against more times than they lost. This is shown in Figure 1 where the ranking from best to worst was ‘B’, ‘C’, ‘A’, ‘D’.

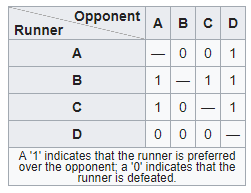


Figure 1.

This method is known as the Condorcet method and utilizing this ranking method we might be able to effectively rank the competitors accurately with limited data. This scales easily as it is just a matter of increasing a competitors relative score whenever they beat another competitor.

Experiments

In order to accurately rank players we needed to figure out an equation where we could input the total number of games and total number of players and get out the certainty that they are all ranked correctly. Step one was to setup a python function that would take the total number of players, the number of players per game, the disparity between their skill range, and the total number of games to be played. That function would write an RMSE value to a file at a predetermined interval of games simulated. This would result in a list of RMSE values and their corresponding number of games played. Step two was to graph that data and determined that the trendline that best fit the data was the power law. So the equation for that line was RMSE = C numGames B where C stands for coefficient, numGames stands for the total number of games played, and B stands for the exponent. Step three was to generate a list of power law equations as they related to the various inputs. Step four was to graph C,B vs T, G, and D where T is the total number of players, G is the number of players per game, and D is the total disparity range of the players. In these graphs only one variable would be changing at a time between T, G, and D so that we could get an accurate understanding of how each variable related to C and B. Step five was to use nonlinear regression software to create equations and residual plots to test the models.

Results

After going through all of the steps in the experiment we were finally able to land on a preliminary equation that would answer the original problem. The original equation we landed on was:

In this equation the coefficient equals -1.24548 + 0.322079 \* T ^ 1.29641 + 0.000111167 \* D / G and the exponent equals -(0.168324 \* D ^ 0.100774 / (G ^ -0.00363187) - 0.344922 \* D ^ 0.161431 / (G ^ 0.0631166 \* T ^ 0.765211)). Step six was to refine that model so that it fit the data even better. We also simplified the model significantly. The final equation we landed on is this:

In this equation the coefficient equals -1.18808 + 0.322101 \* T ^ 1.2964 and the exponent equals -(0.186474 \* D ^ 0.086981 \* (1 - 3.3364 / (T ^ 0.778216 \* G ^ 0.077255))). We were also considering the number of players per game but we found that variable had no influence over the rate at which certainty changes.

Conclusion

This method can be used to accurately rank AI as long as enough games are played. It is certainly not as fast as the other two ranking methods mentioned in this paper but it is also not as complex as those two methods. The lack of complexity allows for the system to run many games without much of a slowdown in processing time.

Future Work

One of our original limitations was the speed at which we could generate data. That has been improved; however, there is still even more room for improvement. Some parts of the process for finding the line of best fit could be streamlined or possibly entirely removed. For example, it is not really necessary to write out to a text file as often as we do. Instead, the data could just be written to a list that gets passed around within the file. Another way that time could be saved is only graphing the lines that we know fit the data the best. We currently are graphing four separate lines for each graph when we already know that each graph only needs one line. This is a step that was initially necessary but now it is not since we know the line of best fit for each data type.

References

Arrow, K. J. (2012). *Social Choice and Individual Values (Cowles Foundation for Research in*

*Economics at Yale University Monograph)*. Yale University Press.

Herbrich, R., Minka, T., & Graepel, T. (2007). TrueSkillTM: A Bayesian Skill Rating System.

Glickman, M. E., & Hennessy, J. (2015). A stochastic rank ordered logit model for rating

multi-competitor games and sports. *Journal of Quantitative Analysis in Sports,11*(3). doi:10.1515/jqas-2015-0012