Ranking Super Smash Characters

Zach Gormley, Jeffrey Hilton, Pearce Keesling, Trayson Keli'i, Garrett Wilhelm
CS 478, Winter 2019 Department of Computer Science
Brigham Young University

Abstract

Your abstract should concisely answer the following three questions: 1) what problem are you addressing? 2) what approach are you taking to solve the problem? 3) what are your results?

1 Introduction

1.1 Problem Description

The Super Smash Brothers series maintains a strong presence in the esports community and is praised for its large, diverse roster of characters. Professional players and amateurs alike are constantly creating lists of character rankings to determine which character would perform the best in a tournament setting. A character's rank represents the relative advantage of using one character over any other in a fight of two equally matched players, and a character tier is a grouping of characters who are ranked similarly, usually listed in letter grades. We aim to run different machine learning algorithms on the rankings and tiers of the 133 characters in previous Super Smash Brothers games, then use the best learned model on the 72 characters in the sequel, Super Smash Brothers Ultimate to predict what their rankings should be. In the future, we will compare our machine-learned rankings with the official rankings when the game stops receiving updates.

1.2 Data Sources

The bulk of our data is retrieved from the Super Smash Brothers wiki (www.ssbwiki.com). This is a community driven database for all of the Super Smash Brothers games. Another source of information that we're hoping to consider comes from SmashBoard user KuroganeHammer who is the curator of a fan-made collection of more in-depth information for the characters (http://kuroganehammer.com). For our initial tests we have chosen not to include this data because it proved to be very difficult to unify across characters. There is also a lot of missing data, especially in the newest generation of the game. For some of the attributes, we wanted to incorporate information that was not purely measurable in the game, such as character archetype. For this we had to rely on our own knowledge of the games and experiments conducted by our team members. Finally, we collected tier information and individual character rankings and tier lists from the following sources to use as potential class labels:

- SmashBoards: https://smashboards.com/threads/4br-smash-for-wii-u-tier-list-v4.452109/
- SmashBackroom (via SSBwiki): (https://www.ssbwiki.com/List_of_SSB4_tier_lists_(NTSC))
- RankedBoost: (https://rankedboost.com/ssb4-tier-list/)
- HTC eSports: https://esports.htc.com/articles/esports-tier-list
- And a collection of other tier lists (including lists made by professional players) published by IGN: https://www.ign.com/wikis/super-smash-bros-ultimate/Tier_Lists

1.3 The Dataset

The data that we are using is the damage output of each character's movesets. Each of the characters has roughly the same set of moves. This allows us to compare categories of attacks across the different characters. An important step in this comparison is to preprocess the data so that they are uniformly measurable. For example, some characters perform multiple successive attacks for the same player input. It is important to combine those attacks together so that it can be compared to another character with only one attack for that player input. Our current dataset consists of 133 training instances, the rosters from games 1-4, with 42 attributes per instance, and one class label–the character's predicted tier. The test set, comprised of the roster of Super Smash Brothers Ultimate, will contain 72 instances, for a combined total of 205 instances. There is a combination of continuous and nominal data. The labels were created from standardizing the individually collected tier and ranking lists, averaging together the standardized scores for each instance, re-standardizing the averaged values to retain a standard deviation of one, and then finally discretizing the values into 6 even width bins to represent a universal standard tier list: S (Superior), A, B, C, D, and F.

The 42 features are: Weight (nominal), Archetype(nominal), Recovery(nominal), Jab(continuous), Jab Type(nominal), FTilt(continuous), FTilt Type(nominal), UTilt(continuous), UTilt Type(nominal), DTilt(continuous), DTilt Type(continuous), Dash Attack(continuous), Dash Attack Type(nominal), FSmash(continuous), FSmash Type(nominal), USmash(continuous), USmash Type(nominal), DSmash(continuous), DSmash Type(nominal), NAir(continuous), Nair Type(nominal), Fair(continuous), Fair Type(nominal), BAir(continuous), BAir Type(nominal), UAir(continuous), UAir Type(nominal), Dair(continuous), DAir type(nominal), FThrow(continuous), BThrow(continuous), UThrow(continuous), DThrow(continuous), NeutB(continuous), NeutB Type(nominal), SideB(continuous), SideB Type(nominal), UpB(continuous), UpB Type(nominal), DownB(continuous), DownB Type(nominal), Tier. Actual Instance: Mario med, hybrid, good, 8, melee, 13, melee, 10, melee,12, melee, 11, melee, 17, melee, 19, melee, 17, melee, 12.5, melee, 13, melee, 13, melee, 10.5, melee, 4, melee, 12, 26, 19, 19, 7, ranged, 7, ranged, 15, melee, 14, melee, B. Because the data is so rich in attributes and attribute values, we've attached a reference to our in-progress dataset and the tier information to give you a better idea of what we've actually collected:

- Attributes: (https://docs.google.com/spreadsheets/d/103elr0mhpr14yhiLLib4_KinrFvMXrBGhfcc1fdAJjE/edit?usp=sharing)
- Tiers: (https://docs.google.com/spreadsheets/d/18_zleVDd54rsXNylzEztfFuTKHLIQ6wrmeXgQ1AGhSc/edit?usp=sharing)

Note that, for most classification problems, having only 133 training instances with so many attributes for each instance would be problematic. However, the problem we're solving is unique in that the universal set of instances is described in full with the 205 characters across all games (including Ultimate). And with how unique each character is, the more detail we can capture in each instance, the better. Thus the need for a somewhat larger set of attributes and possible attribute values.

2 Methods

2.1 KNN

2.2 K-Means

2.3 MLP

The features used to solve the problem and details on how you gathered and represented the features, including critical decisions/choices made along the way

3 Initial Results

Your initial results with your initial model.
The iterative steps you took to get better results (improved features and/or learning models)

4 Final Results

Clear reporting and explanation of your final results including your training/testing approach

5 Conclusions

Conclusions, insights

6 Future Work

future directions you would take if time permitted

A References

Cite Smash wiki