Predicting the Outcome of Dota 2 Matches Using Machine Learning and Algorithms

Meg Coleman, Michael Demone, Kenneth Huang, Jordan Patterson, Alexander Riga
University of Victoria
Course: Software Engineering 474, Data Mining
Date: April 9, 2018

***Abstract*—The purpose of this project was to predict the winning team of a Defense Of The Ancients 2 (Dota 2) match. This was done using features based on players' skill and hero statistics. For each player, there are eight possible features for hero statistics, and two possible features for player skill, leading to 100 total possible feature inputs. This indicated whether team hero composition or player skill influences the outcome of a match, and the correlation for both. It was found that hero composition had no effect on making a reliable prediction, and that player skill was very effective at reliably predicting the outcome of a match. Furthermore, it was found that although decent results can be had with simple neural networks, logistic regression, and stochastic gradient descent, the best results were attained with random forest classification, surpassing its peers by 10% or more at times.**

I. Introduction

Dota 2 is a MOBA (Mulitplayer Online Battle Arena) game by Valve, in which players are divided into two teams of five. Each player chooses one character, known as a "hero", out of a pool of 114, with only unique heroes allowed. Each hero has different skills, strengths, and weaknesses. The objective of the game is to destroy the base of the enemy team. It is heavily skill-based, and professional Dota tournaments have prize pools upwards of $20 million dollars. [1]

Dota 2 is a very difficult game to predict an outcome for beforehand, due to the many unknown variables that occur during play. Skill levels are always improving, parameters are changed often to ensure the game remains balanced, and heroes are slowly added to the game. The goal was to predict the outcome of a game based on hero information, player skill, and a combination of both. This was done with neural networks, logistic regression, random forest classification, and stochastic gradient descent in a linear Support Vector Machine.

II. Related Work

Multiplayer Online Battle Arenas (MOBAs) are very popular in the online game community, especially Dota 2. Researchers F. Rioult, JP. Métivier, B. Helleu, N. Scelles and C. Durand analyzed matches for 200,000 players, to determine outcomes based simply on player location and speed. These topological clues were used in various mathematical functions to determine area, gathering ability, inertia, diameter and remoteness of a polygon corresponding to teams every second. According to their results, area was irrelevant of the final outcome. However, gathering ability and inertia were very relevant to the outcome of a match, with the winning team having better gathering ability and weaker inertia. Diameter and remoteness also had an effect, but to a lesser degree that was not explained. The conclusion of the report was that little to no difference was found between level of play and outcome; the most collected and grouped team wins whenever skill level is alike. This determined that team play in Dota 2 is more important than individual player skill. However, the study acknowledged that a lot of bias existed in how they used the data, so their high prediction accuracy may be inflated. [2]

A. Apostoae predicted the outcome of matches in a completely different way, taking input exclusively from before the matches began. [3] The training input was match outcome, team compositions with synergy, average public MMR, number of players with MMR, game mode, and lobby type. Logistic regression was used with cross validation for training, with two possible output queries. One was a prediction based on the ten heroes in the game, the other predicted the best tenth hero to pick based on nine heroes already selected. Aspostoae achieved 60% accuracy with his model, and admitted that because of how much noise there is in ranked games, his accuracy is not much better than predicting one team will always win.

K. Conley & D. Perry used logistic regression with k-nearest-neighbors and cross validation to achieve an optimal accuracy of 69%. [4] Their input data ignored matches where players left the match early, used only players ranked in the top 8% of MMR, and chose specific game modes to target. The logistic regressor had 104*2 features, one for each unique hero on each team. This means that the logistic regressor had no information on the players' skill, but still achieved an accuracy of 69%. We are highly skeptical of this result, and believe that it would not be replicable in modern Dota 2 matches. Based on Aostoaes results, and later ours, we show hero choice has no effect on the outcome of typical ranked games.

III. Data sources

A relevant project was found on Github, which was created by Apostoae. It contains 1.7 million matches, each with columns for match id, winner, team composition (heroes on each team), average MMR, number of MMR ranked players, game mode, and lobby type (ranked or unranked). This data set was used to build our initial models. https://github.com/andreiapostoae/dota2-predictor

OpenDota was used for retrieving relevant hero data, such as win rates and average gold earned for heroes, as well as player skill information. This data was compiled into three files, one for player skill information, one for hero information, and one for hero win rate. This API was also used for gathering data on recent matches for further testing of several models. https://docs.opendota.com/#

IV. Project Details

All investigated features rely on either the players' ability or their hero; each feature is given in more detail below.

Features based on the players' skill: players matchmaking rank (MMR) and players score. The players MMR gives a numeric representation of their skill calculated by the game in order to place people of similar MMR in games together. A player's score illustrates how good a player is with their current hero and is also calculated by the game, but for players to track themselves to see what heroes they play best with.

Statistics recorded for each players' hero: global win rate, gold per minute, experience (XP) per minute, kills per minute, last hits per minute, damage per minute, healing per minute, and tower damage. The goal of using these features was to have the algorithms determine the teams composition, and whether they would work well together or perform well independently. A team might have many aggressive heroes that get a lot of kills, but none that give support to each other, so they could have very high kills per minute, but very low healing per minute. Each hero would look strong, but without support, the team would die often and potentially fail because of poor team composition. Finally, the win rate for the hero currently being played shows its overall effectiveness compared to other heroes in the meta.

Initially, the algorithms used only the win rates for the hero that each player is currently playing to predict the match outcome. Using a readily available dataset that has 1.7 million matches with what heroes are on each team, we took the hero win rates from the OpenDota API and combined them will this dataset [3]. Following this, the algorithms were trained on all the hero features except the hero win rate. These two datasets were used to see how well the algorithms can predict an outcome entirely based on team composition. The final dataset used incorporated only player skill, and was trained on all algorithms to see the effect player skill alone would have.

V. Methodology

We use only ranked matches for all data, since that is what we wanted to train on.

It was necessary to preprocess the data so that it could effectively be used for our various algorithms. First, a dataset created from hero win rate was used, with a feature set of shape (N, 10). We later focused on the Apostoae dataset to test how our algorithms would compare to this previous study. This study had data for N matches with columns for match id, radiant win, hero ids for each player on each team, average game mmr, individual mmr, the game mode, and the lobby type. It used a feature set of shape (N, 70).

The second dataset was created from a python service which retrieved game data in real time from the Dota 2 API and fed it to the algorithms after preprocessing. The script queries the Dota 2 API directly for 40,000 matches at a time; of these 40,000 matches, as low as 6,500 are complete and valid for use. It places the match id for those N matches into a dictionary, containing the score and MMR for all players in the game, creating a feature set with shape (N, 20).

All datasets were normalized. Once we had the data, we used a variety of algorithms for prediction such that results could be compared to maximize our accuracy. We implemented logistic regression, a simple neural network, stochastic gradient descent, and random forest classification. The hero winrate dataset was trained and tested on logistic regression and the neural network. The hero parameters dataset was trained and tested on logistic regression, the neural network, and stochastic gradient descent. The player dataset was trained and tested on logistic regression, the neural network, and random forest classification.

VI. Results and Discussion

The PyTorch logistic regression and PyTorch neural network initially used the hero winreate dataset to determine if global win rates of heroes could be used to predict matches. This network was trained on 200,000 randomized matches from a training dataset, and then tested on 1,000 matches from a testing dataset, which were sourced from Apostoae [3]. The network has 10 inputs, one for each hero's global win rate. It's optimizer is stochastic gradient descent, with several architectures and activation functions experimented with. Initially, a network with two hidden layers of 100 and 40 nodes was made, with ReLu activation functions and a sigmoid activation function from the last layer to the output. This simple network converged to a 50% accuracy rate, along with the logistic regression. Following this, a network was tested with three smaller layers containing 20, 10, and five nodes, respectively. There were three tanh activation functions, with sigmoid once again as the final layer's activation function. This network also converged to 50% accuracy each of the five times it was run. These results lead to the assumption that hero win rates mean nothing on a game to game basis, and cannot be used to predict outcomes. Based on these results, the final method should not use hero win rates as a feature.

Although win rates of individual heroes had no significant effect on the outcome of the game, it was thought that other statistics of the heroes would have a significant part to play in the outcome of a given match. Using the Apostoae dataset again, which supplied hero ids from about two million matches, and the most current statistics of in game heroes, parsed from OpenDota, a new dataset was created that contained information on each hero in the game. The dataset was laid out so that each player had seven attributes: Gold per minute, experience per minute, kills per minute, last hits per minute, damage per minute, healing per minute, and tower damage per minute. This dataset was used on three different machines, the Stochastic Gradient Descent Classifier provided by Scikit-learn which is a linear Support Vector Machine, as well as the same logistic regression and neural networks from before. All three algorithms obtained ∼50% accuracy with SGD Classifier obtaining the best accuracy of 50.5%. The reason for these poor results are discussed in the Issues section.

Since all hero statistics appeared to have no affect on the outcome of the match, the only other variables before a match are player skill related. The last dataset used was comprised of each player in the match's matchmatching rating (MMR) and score. The player score is calculated by the Dota 2 API, and is a quantitative description of how well this player plays their hero. This dataset ended up having 40,000 matches, but only less than 6,500 matches had all of the player data. The algorithms used were tested on both complete matches and matches with missing values, which were filled with average statistics from the rest of the team. For this dataset several algorithms were tried including scikit-learns random forest classification and linear regression as well as the same logistic regression and neural networks from earlier.

For comparison, a hardcoded algorithm was tested that takes the average MMR and score of each team and compares them. When tested on the matches with incomplete player data, this algorithm scored 60.8% on the average of 10 trials. When tested only on matches with complete data, this algorithm yielded a 68.4% accuracy.

Pytorch was set up with the intent of having multiple networks trained on it. The first was logistic regression (0 hidden layers and a sigmoid activation function) and it obtained an average of 61.4% accuracy over 10 different runs with a shuffled dataset and incomplete matches filled with averaged data. On complete matches only, this network achieved a 63.9% accuracy over 10 trials. This low accuracy may have been from the lack of trials, so the learning rate of the machine was increased from 0.01 to 0.1 and the machine was run through more trials. This resulted in more random accuracies, but obtained a max accuracy of 73% over 10 trials and an average of 68.3%, just under the hardcoded algorithm accuracy. Since logistic regression yielded poor results due to a small dataset, no other algorithms were tested on this data as they would all need larger datasets than logistic regression.

The final machine tested was a random forest classifier provided by Scikit-learn. This yielded 61.2% accuracy with the incomplete matches, but for complete matches it out performed everything else with an average accuracy of 79.2% after 10 trials [5]. This model also reported a max accuracy of 87% and a minimum accuracy of 74%, which is still higher than the best logistic regression model. However, we were skeptical of this and decided to retest on new player data, being parsed from the OpenDota server on request. This showed random forest classification yielding 69% accuracy on average, which was the same as hard coded, and only slightly better than the logistic classification. However, it is apparant that random forest classification has the tighest spread on this data, with a high min and low max. This gives evidence that predicting based solely on player skill is typical to be around 69% accurate, and it is the same as hardcoded because they function similarly. The 80% average accuracy generated by the original training data must therefore be from one of two reasons. It could be invalid output, and some error was taking place in how the data was handled. Otherwise it is valid, and training on old data then testing on new data is unreliable due to the different time periods the matches took place in. Training and testing on seperate new datasets would be the way to determine this. All of the results can be seen in table I, table II, and table III.

VII. Issues

Initially, the intention was to have statistics about each individual player in the match being input into the various algorithms. If given a player id number, various statistics about them can be queried, such as their personal win rate, their win rate on specific heroes, and much more that could give us context about the type of player they are

Model	Accuracy	Standard Dev	Min	Max
Hardcoded	68.3%	4.1%	61.5%	75.0%
Log Reg.	66.8%	4.6%	58.0%	73.0%
Random Forest	79.3%	3.7%	74.0%	87.0%

TABLE I
AVERAGE MACHINE STATISTICS ON 2931 COMPLETE MATCHES ONLY OVER 10 TRIALS

Model	Accuracy	Standard Dev	Min	Max
Hardcoded	60.9%	1.2%	59.3%	64.0%
Log Reg.	61.3%	1.9%	57.9%	64.1%
Random Forest	60.9%	0.9%	59.2%	62.3%

TABLE II
AVERAGE MACHINE STATISTICS ON 12777 COMPLETE AND INCOMPLETE MATCHES OVER 10 TRIALS

Model	Accuracy	Standard Dev	Min	Max
Hardcoded	69.9%	0%	69.9%	69.9%
Log Reg.	66.8%	2.4%	60.7%	69.2%
Random Forest	69.0%	0.6%	68.2%	67.0%

TABLE III
AVERAGE MACHINE STATISTICS WHEN TRAINED ON OLD COMPLETE MATCHES AND TESTED ON NEW COMPLETE MATCHES OVER 10 TRIALS

and how good they are. It became apparent after spending time working with the Dota 2 API and analyzing all possible data sources, that this information is simply not available in an attainable way. When players play Dota 2, they have the option to not share their statistics. If a player has selected this and is in the game we retrieve from the API, then the game data will not have their user ID or anything about them specifically. This did not initially appear to be an issue until, after scraping hundreds of games, we realized that very little of the games queried had all player's information. This was the same issue with all of the other datasets we used.

Since hero selection is such an important part of the Dota 2, the developers are constantly changing and adding new heroes to the game. This keeps the team compositions fresh and forces players to learn more heroes. In the Dota community, this is referred to as the ”meta” changing. We believe our 50% accuracy on the win rates and team composition algorithms were due to our hero information having a different ”meta” than the matches that we had. The dataset with hero ids was retrieved three years ago, but the ”meta” information was gathered in the middle of March 2018, so the heroes have most likely significantly changed since then.

VIII. FUTURE WORK

After having difficulty with the earlier dataset not showing good results, a newer dataset was created and used to achieved better results, but it was used on different algorithms. It could be interested to go back to the algorithms that were used for our first dataset and test those with the new dataset to see if the dataset was truly the issue.

With enough time, it could be very interesting to grab all games with all player data to train and test. For instance, find only games where all 10 player ids are available to see. Then it would be possible to query the API based on those player ids to see more in depth stats about each person playing, such as their win rate on certain heroes, their preferred roles, and other relevant statistics for predicting a game outcome. We believe this would be an extremely effective method for prediction, and it would include nearly all of the possible data that could be retrieved about a game before it starts.

Finally, getting training and testing on new data would give more accurate output and allow us to draw better conclusions from our algorithms.

IX. ETHICAL IMPLICATIONS

Although this project is about predicting video game outcomes, there are still some relevant ethical implications. In video games with ranking systems such as Dota 2, there are often issues with players trying to gain ranks unfairly or abuse the rank system. Two of the most common methods for this are ”smurfing” and ”boosting”. Smurfing is the act of purposefully losing rank, or buying a new account at a lower rank to play against lower ranked opponents to rank up quicker or to simply play easier games. This can ruin the experience for lower ranked players, as they won't be able to compete with the smurfing player. Boosting is when players use smurf accounts to rank up their friends or other players. This gives them a significant unfair advantage and allows them to rank up much faster. Using our methodology to determine the most effective hero composition and what gives them the most likely chance of winning, this could be used for smurfs and boosters to game the system even more and rank up faster than they deserve. This is important, because if enough people understand that they can win by using tricks instead of playing more and improving, then the entire game ecosystem can be ruined and it can become not fun to play any more for the community as a whole.

Considering our low accuracy with hero data, however, this is unlikely to be an issue. Rather, with the player skill data and the random forest classification, which is capable of 70%+ accuracy easily on the most recent matches, there is potential ethical complications. If one were to create a system that gathered the skill data from all opposing players in the match, and kept their own team skill data scored, they could reliably predict if they were likely to win a game or not before it began. This becomes an issue when used in lobby and before the start of the match, where it can be used to filter matches for one with a high probability of success. This essentially becomes a legal form of boosting that is very difficult to catch, and will help the players doing it rank up far more than their counterpart.

Finally, if similar results are capable of being found on pro matches where betting can take place, it will become an issue until the 'house' taking the bets accounts for this and creates an odds vs rewards gambling system, such as what is present in most sport betting. Until caught however, it could pose a serious problem for easy low risk betting for those using data such as what is present in this project.

X. Conclusion

We were able to successfully train several models and achieved a maximum accuracy of 87% for predicting the outcome of a Dota 2 match. We also discovered that a hero's statistics has minimal impact to a Dota 2 match outcome, and it is the player's skill that matters. We came to this conclusion by gathering and training on different datasets and parameters. During the training process, many models were attempted, including multi-layer neural networks, logistic regression and random forest classification. Among all the attempted models, random forest classification yields the best results, giving the previously mentioned 87% accuracy. Using simple statistics, we were able to get 69% accuracy on nearly any player data, which matched random forest classification with some datasets. If more time was allowed, we would train on more in depth player specific data to see what attributes for a player change outcomes the most. However, it is safe to say that player skill has a noteable impact on the outcome of any given Dota 2 match, and needs to be taken into account when predicting.
Temp save…delete soon

XI. Task Distribution

Distribution of tasks amongst team members
No p Preprocessing: (completed by March 14)

- 1.1:
  Gather hero data from Dota API - Alexander
- 1.2:
  Gather player data from Dota API - Kenneth
- 1.3:
  Preprocessing data - Mike, Jordan

Algorithm and Analysis: (completed by March 20)

- 2.1:
  Create, train and test neural network - Mike
- 2.2:
  Train network with different features - Alexander
- 2.3:
  Train with SGD - Meg
- 2.4:
  Train and test with random forest regression - Jordan
- 2.5:
  Compare the classification functions - Jordan

Evaluation and Accuracy Testing: (completed by March 28)

- 3.1:
  Test algorithm on new data and examine the results - Jordan, Mike
- 3.2:
  Test on different architectures and hyperparameters - Jordan, Mike

References

[1] [Online]. Available: http://dota2.prizetrac.kr/international2017. [Accessed: 04-Mar-2018].

[2] F. C. A. Rioult, J.-P. Métivier, B. Helleu, N. Scelles, and C. Durand, "Mining Tracks of Competitive Video Games," AASRI Procedia, vol. 8, pp. 82–87, 2014. [Online]. Available: https://ac.els-cdn.com/S221267161400081X/1-s2.0-S221267161400081X-main.pdf?_tid=d50fde13-c318-4612-8de2-ab740d74a99f&acdnat=1520198102_12004b2c2efe1e0b17b9b531fae9853c. [Accessed: 04-Mar-2018].

[3] Andreiapostoae, "andreiapostoae/dota2-predictor," GitHub, 17-Oct-2017. [Online]. Available: https://github.com/andreiapostoae/dota2-predictor. [Accessed: 04-Mar-2018].

[4] K. Conley, D. Perry, "How Does He Saw Me? A Recommendation Engine for Picking Heroes in Dota 2", Stanford. [Online]. Available: http://cs229.stanford.edu/proj2013/PerryConley-HowDoesHeSawMeARecommendationEngineForPickingHeroesInDota2.pdf. [Accessed: 04-Mar-2018].

[5] Jordan Patterson, "Dota 2 Random Forest Classification", Github. [Online]. Available: https://github.com/jpatts/DOTA2-predictor. [Accessed: 09-Apr-2018].