Bachelor thesis
Computer Science

Radboud University

Improving matchmaking with game data

Author:
Wietse Kuipers
4317904

First supervisor/assessor:
Prof. dr. T.M. (Tom) Heskes
tomh@cs.ru.nl

Second supervisor:
MSc I.G. (Gabriel) Bucur
G.Bucur@cs.ru.nl

August 17, 2016

Abstract

In this work several weaknesses of matchmaking for team games with skill rating systems are highlighted, and an alternative data based approach to matchmaking is presented. This new approach is then tested using matches from the five versus five online game *Dota 2*.

Contents

Chapter 1

Introduction

A competitive game is not fun when there are players with no chance to win. For all players to enjoy the game it is important to make sure they all have a somewhat equal chance to enjoy the game. Various systems have been designed to quantify skill and find balanced matches, the most famous one arguably being the Elo rating system, created by the American chess player Arpad Elo. Such rating systems assign a number to players, and the closer the ratings of two players, the more fair a game between them should be.

With the rising popularity of multiplayer games played over the internet, a demand for matchmaking was created. Matchmaking is the process of selecting players to play with and against each other from a pool of players. The metric used to select the players is often a rating as shown by some some of the most popular online games right now. The games are played in a team-versus-team format, further complicating matchmaking. This research presents an approach to matchmaking for team games that uses data from previous games to suggest fair games. The data was gathered from the online team game Dota 2.

Chapter 2

Related work

2.1 The Elo rating system

The Elo rating system[3] is frequently used to give players a skill rating, and that rating is then used to perform matchmaking. This is the case in Dota 2, and as such the new system proposed in this research will be compared to Elo.

The Elo rating system is used by many different competitions due to its simplicity and how it can be applied to almost any game the only data required to update an Elo rating after a match is the rating of the winner and the rating of the loser. The system also takes into account large rating differences in matches, which makes winning against much better players much more valuable.

Adapting the Elo rating system to team games is also easy, as you can take the average rating of every player on a team, and then use the average rating of the teams to calculate the updates to each rating. Doing this means you assume that the skill rating of a team is the average of all ratings of the team's players, which might not be true for all games.

2.2 Other research

Research by Olivier Delalleau[2] proposes a new method for matchmaking which uses a neural network trained on matches and player feedback on those matches. This method maximizes the fun players have rather then the balance of a match and uses a neural network to achieve this rather then simpler models.

Research by Hao Wang[6] shows another way of matchmaking that yields more fun matches by identifying the playstyles that make a game more fun. This research also links player enjoyment to match length with a small survey, indicating that players consider short games less enjoyable.

Research by M. Claypool[1] further investigates the link between match duration, balance and enjoyment. It analyzes some *League in Legends* games with objective and subjective data, and concludes that games which appear balanced from the player ranks are often found unbalanced by the losing team. It also concludes that players enjoy unbalanced matches as long as they are on the winning team.

The method proposed in this research differs from the ones described above in the algorithms used and the focus on match balance rather than on player enjoyment. Another difference is that our method does not rely on player feedback on matches.

Chapter 3

Method

3.1 Determining balance

For a competitive game, balance is whether every player in the game had an equal chance to win the game. If the game is symmetric, then a game is perfectly balanced when every player is equally skilled at the game. Balance is important in many competitive games, because playing the game is more interesting for the players when they have a fair chance to lose or win.

Many games and sports, both physical and digital, offer ways to find balanced matches. A popular way to do this is to use Elo-rating[3] or a variant of this, where the skill of the player is reflected by a number, the rating. The problem here is that Elo-rating and others only calculate this rating by looking at whether you lost or won a match, and against who you did this. These ratings do not take into account how you actually play the game, which could lead to a match being unfair due to how players play even though their ratings are close.

In many games, it can be said that a game is short in duration when the players differ significantly in skill, and long when the players are equally skilled. A good example of this is tennis, where a game between close players goes on until there is a two point difference, whereas a much better player can win in just four servings. If a game has a correlation between duration and balance, then you can create balanced matches by selecting the matches that are the longest.

Assuming this positive correlation between match duration and balance exists, you can improve the balance of matches by making them longer.

3.2 The dataset

The dataset used for this research are match results from public Dota 2 matches. After playing a match of Dota 2 through the matchmaking system offered by the game, the results of the match are exposed through an API.

The data contained in a match results consists of various statistics from the played game.

The collected data consists of 8266 Dota 2 match results, played between 9PM and 10Pm on the 2nd of January. Some filtering is done on the data, as there are some issues with public matches, namely:

- It is possible for players to stop playing a match before it ends, which inherently causes unbalanced matches.
- Some public matches are played against computer players, who almost always lose.
- The matchmaking system is sometimes abused to have a team of automated players lose as quickly as possible.

In order to filter out these matches, only matches that lasted more than 10 minutes, had ten human players and were played to completion are considered. The minimum duration of 10 minutes was chosen by looking at matches containing bots, computer programs that pretend to be human players, which all ended before 10 minutes. Furthermore, ending a game of Dota 2 before 10 minutes almost always requires coordination between both teams, which invalidates the data for this research as the players are not interested in a balanced game in this situation.

A dataset containing players, their lifetime average statistics and their Elo-rating was retrieved from the `yasp.co` API.

3.2.1 Relevant features

From the data retrieved from the API, only a few features are relevant when trying to predict whether a match is balanced. The main factors that indicate the performance of a player are:

Feature	Explanation
Gold per minute	The total amount of gold earned divided by the duration of the match in minutes. Gold is earned by taking objectives, and defeating other players.
Experience per minute	Similar to gold per minute, but experience is also gained by simply being present when gold is awarded.
Hero damage per second	The amount of damage done to enemy heroes divided by the duration of the match in seconds. Although it is not necessary to damage heroes to win the game, it is often hard to take objectives when enemy heroes are alive.
Tower damage per second	The amount of damage done to enemy buildings divided by the duration of the match in seconds. Destroying towers is necessary to win the game.
Kills, deaths and assists per seconds	When a hero is defeated, the defeated player is given one death, and the player who defeated the hero is awarded a kill. Anyone who helped in defeating a hero is awarded an assist.

These features are summed for every team, and the difference between the two sums is the actual feature used in the model.

These features are useful in particular because they likely correlate with the skill of player and they are always present for every player. Because of this the averages of these values for a player can be used when trying to predict the duration of a match that has not been played yet.

3.3 Predictive models

In order to predict the duration of a match given certain features, a model needs to be trained on the dataset.

3.3.1 Ordinary Least Squares regression

Ordinary Least Squares is a regression method that works by by assigning a weight to every independent variable, such that the sum of squared residuals is minimal, where residuals are the differences between the observed and the predicted value of the dependent variable.

Linear regression assumes that the relationships between the explanatory variables and the dependent variable is linear, which could be the case for this dataset after the transformations described in the previous section.

3.3.2 K Nearest Neighbors regression

K Nearest Neighbors performs regression by finding the K closest neighbors with some distance measure, Minkowski distance here, and then taking the mean predicted value of those neighbors.

If matches with similar feature have similar durations, then this algorithm will perform well. Bad performance from this algorithm would suggest that matches with similar features have very different durations.

3.3.3 Random forest regression

Random forest regression works by using multiple, different decision trees and averaging their prediction to get the actual predicted value. This is an improvement over normal decision trees, which tend to over-fit.

All these models are implemented in the scikit-learn[5] package for the Python programming language.

3.4 Testing the model

The proposed matchmaking process is tested by comparing it to simple Elo-based matchmaking. Given ten players and their Elo-ratings, the process for Elo-based matchmaking is simple: Find two teams of five players and for each team, take the sum of the Elo-ratings of the players. The best match is the match where difference between the sums is the smallest.

Our proposed model based matchmaking is tested by comparing the teams suggested by the model to the teams suggested by Elo-rating based matchmaking. The model finds the teams that will result in the longest match by comparing every possible team and taking the teams that would result in the longest match.

Chapter 4

Results

4.1 The matchmaking process

The features discussed earlier have a problem if you were to use them as you would use a rating as given by a rating system. The features reflect the overall performance of a team or individual performance within that specific game, but they do not take into account how skilled your opponents were. Because of this, metrics indicating high performance can be deceiving, as you may have gotten those by playing against players who are less skilled than you.

One way to solve this problem is to use a rating system such as Elo, as described in chapter 2. A more suitable procedure for matchmaking would thus be to first find a set of players with close ratings that somewhat reflect their skill, then shuffle that set in such a way the predicted duration is the highest. If the assumption that longer matches are more balanced holds, the shuffling will result in the most balanced game you could have created.

4.2 The models

The models are all evaluated by their R^2 coefficient. This coefficient can be calculated for every model chosen, and indicates whether the model is producing good predictions, and if the model is actually using the input features to make these predictions. This is calculated by letting a trained model perform regression on a test data set, and then calculating $R^2 = (1-u/v)$ where u is the sum of the squared differences between the predicted values and the actual value, and v s the sum of squared differences between the actual values and the average actual values. The best possible score is 1, and lower scores are worse. The score is calculated by cross validation over ten folds, meaning the prediction for one fold is calculated by a model trained on the other nine folds.

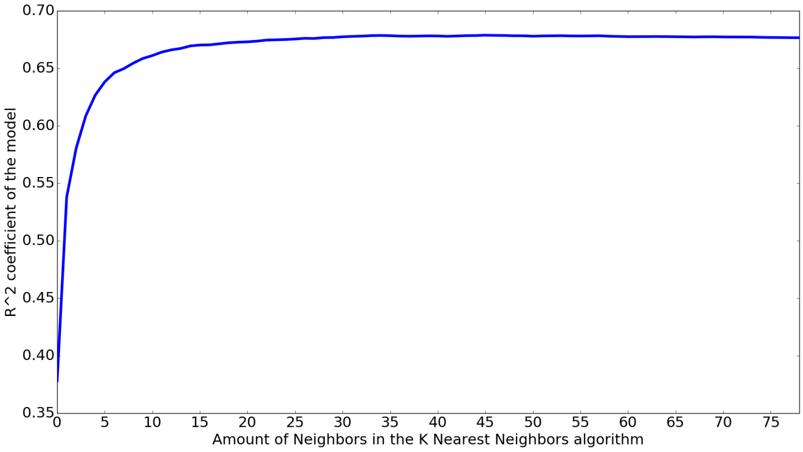

Figure 4.1: The R^2 coefficient of the trained model as a function of the amount of neighbors for K-Nearest Neighbors regression when trained on the match dataset.

4.2.1 Ordinary Least Squares regression

The Ordinary Least Squares regression model has a R^2 coefficient of 0.639 when trained on the entire dataset.

4.2.2 K-Nearest Neighbors regression

This model uses K-Nearest Neighbors regression with uniform weights and Minkowski distance. The amount of neighbors is an important parameter in this model so this had to be optimized first. The R^2 coefficient was found to be the largest when using 45 neighbors, with the coefficient being 0.679.

4.2.3 Random forests

Important parameters of random forest regression are the amount of trees used, the maximum depth of these trees and the amount of features used in every tree. The coefficient always improves or stays the same when increasing the amount of trees used, so the amount of trees is kept at 31 to reduce computation times and because the coefficient does not improve a lot beyond this number. The optimal values for the other parameters were found with a grid search, which found that four features per tree with each tree having a maximum depth of seven was optimal. With these parameters the model has a R^2 coefficient of 0.679.

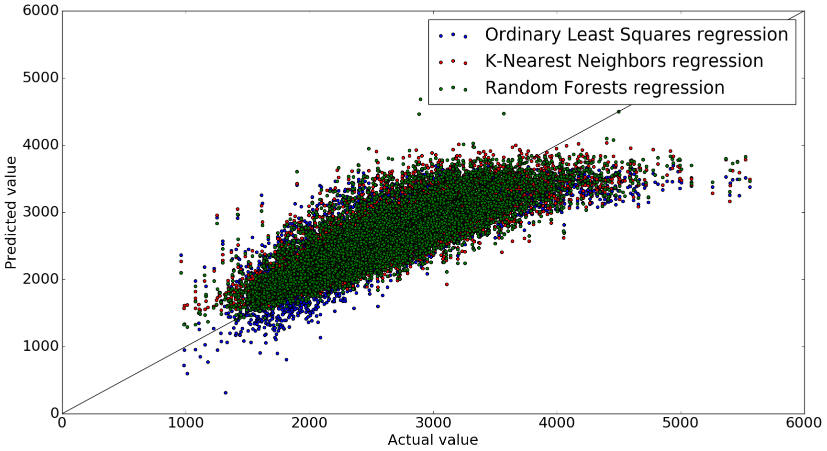

Figure 4.2: A scatter plot where every point is a match from the dataset. The X-value is the actual match duration for that match, and the Y-value is the prediction the model did. The colours indicate which model made the prediction

4.2.4 Model comparison

Random forests and K-Nearest Neighbors both produce the best models as shown by their R^2 coefficients, and it can be seen on figure 4.2 that the predictions by these models are closer to a perfect prediction than Ordinary Least Squares regression.

The random forests model is thus the most suitable for matchmaking, as the performance the same as K-Nearest Neighbors but does not require normalization of the input data, so less computation is required.

4.3 Testing the matchmaking

The model based matchmaking was tested against Elo-based matchmaking in two scenarios: Finding the best match for ten players with a close Elo-rating and finding the best match for random players.

There are some practical problems with verifying whether the matches suggested by the model would actually be longer, as it was not practical to actually have the matches played. Because the results can not be verified by a test data set, it is important to adjust the experiment in a way that would punish a model that assigns random match lengths to team compositions, as such a model would assign some random matches a long duration , which would give plausible results even though the underlying model is random.

To penalize random models, a random number is sampled from the error distribution of the model, which is assumed Gaussian. For our trained model, this is a Gaussian distribution with a σ of 360 and a mean around zero.

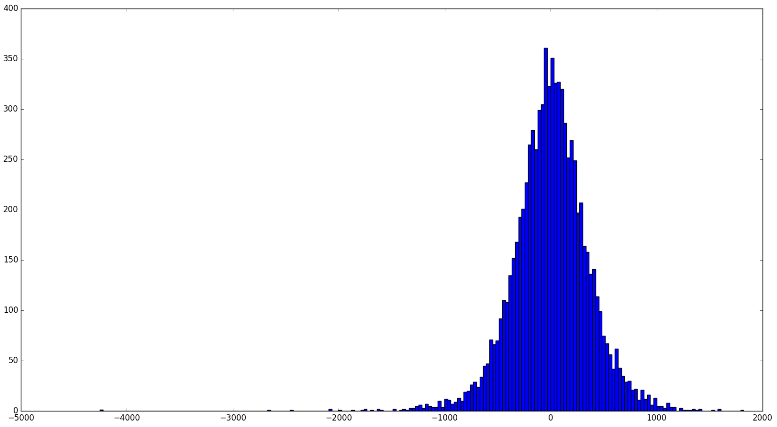

Figure 4.3: The error distribution for the trained model

The results of the experiment are as following:

Scenario	**Average match duration**	**Average Elo gain**
Elo based matchmaking with similarly rated players	3633 seconds	25
Model based matchmaking with similarly rated players	4173 seconds	22
Elo based matchmaking with random players	3607 seconds	25
Model based matchmaking with random players	4236 seconds	4

The Elo difference translates into the gain or loss of Elo rating after winning or losing. The amount lost or gained is based on the difference in average rating, and indicates how fair a game was according to the Elo rating system. In a game where the average rating of two teams was very close, the amount of rating lost or gained is 25. [4] The further this amount of rating lost or won is away from 25, the more unbalanced the game was according to the Elo rating system. Both Elo-based matchmaking experiments yielded matches that would result in a rating gain of 25 on average, as the average Elo difference is quite low. With model based matchmaking for teams with similar ratings, the average match would have an Elo gain of 22, which is quite bad. For random teams, the average Elo gain would be 4, which would be unacceptable in practice.

Chapter 5

Discussion

5.1 Dataset consistency

The match data collected from Valve's own API should be a good sample of all Dota 2 matches, as doubling the amount of samples did not cause a significant increase in the accuracy of the predictive model. The data of the players, their averages and their Elo-rating were retrieved from a third party website which introduces the following problems:

- You must opt-in to make your data public, so the data could be skewed.
- The website only exposes the most recent data and this data is collected over all available matches of a player, so this data might not match the time frame of the match data.

The player dataset is large enough(1069672 players) to still provide a reasonable sample, and the average value of the statistics should not fluctuate significantly between different time periods if enough matches are played.

5.2 Model accuracy

All models seem to be somewhat good at predicting match length using the input features, as indicated by their R^2 coefficients. What is not clear is whether the lifetime averages of players are useful inputs to the model. As long as the players achieve results that are close to their averages, the model will produce good predictions, but there is no guarantee that this is the case. Furthermore averages can be slow to update as a player improves or gets worse, but this can be prevented by only taking the average over a set amount of recent matches.

Perhaps better predictions could be obtained by using deep learning models, but due to computation time constraints they were not considered for this research.

5.3 Experiment results

The model based matchmaking appears to have its intended effect of finding matches that are predicted to be longer than matches found by Elo-based matchmaking.

When the players have Elo ratings that are close, there seems to be a clear trade-off between match length and Elo rating. The increase in Elo difference when only considering the model is too high to be used in practice, but this could be compensated by setting a maximum Elo difference for matches, and using the model within that bound.

The bad performance of the model in the experiment with random players is explained by the problem described in section 4.1. Because players with very different ratings can have similar feature averages, the model might create matches that will absolutely be unbalanced, but it should be noted that matchmaking with completely random players is not a very good idea to begin with.

Chapter 6

Conclusion

The large amount of data generated by Dota 2 makes it possible to take a data driven approach to many problems, including matchmaking. The proposed algorithm shows one way to do matchmaking using public match data, but still relies on some form of skill rating to keep the results sensible.

The experiment done to verify the working of the algorithm does not prove its effectiveness over Elo rating as the matches were never played, and thus no feedback was gather, but it serves as a proof of concept that the data based matchmaking produces sensible matches.

The main problem in this research was the lack of one consistent data set containing matches and their players, including the averages and the ratings of the players. With such a dataset it would be easier to verify any new matchmaking algorithms, but the results presented here show that alternative approaches including ours have a lot of potential.

Bibliography

[1] M. Claypool et al. "Surrender at 20? Matchmaking in league of legends". In: *Games Entertainment Media Conference (GEM), 2015 IEEE*. Oct. 2015, pp. 1–4. DOI: `10.1109/GEM.2015.7377234`.

[2] O. Delalleau et al. "Beyond Skill Rating: Advanced Matchmaking in Ghost Recon Online". In: *IEEE Transactions on Computational Intelligence and AI in Games* 4.3 (Sept. 2012), pp. 167–177. ISSN: 1943-068X. DOI: `10.1109/TCIAIG.2012.2188833`.

[3] Arpad E Elo. *The rating of chessplayers, past and present*. Arco Pub., 1978.

[4] *How much MMR am I risking? (Approximately)*. URL: `https://www.reddit.com/r/DotA2/comments/3zf7pv/how_much_mmr_am_i_risking_approximately/` (visited on 05/25/2016).

[5] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[6] H. Wang, H. T. Yang, and C. T. Sun. "Thinking Style and Team Competition Game Performance and Enjoyment". In: *IEEE Transactions on Computational Intelligence and AI in Games* 7.3 (Sept. 2015), pp. 243–254. ISSN: 1943-068X. DOI: `10.1109/TCIAIG.2015.2466240`.

Appendix A

Dota 2 in a nutshell

Dota 2 is a video game played between two teams of five players, where each player controls a character called a hero. The goal of the game is to destroy the stronghold of the opposing team. In order to do so, players must gather resources(gold and experience) to strengthen their heroes by defeating computer controlled units or the opposing players.

Although the statistics used in this research are limited to some averages per player, they offer a good perspective on the performance and role of a player within the game.

For more information, Valve has released a short video, but the best way to learn about the game is to watch or play it.

Appendix B

Match data description

The match data used for this research was retrieved through the official API that valve exposes. It allows you to download match data in JSON format by making GET requests to their server. One such downloaded match looks like this, but in a real match there would be ten players with more entries in the `ability_upgrades` field.

```
{
    'game_mode':3,
    'engine':1,
    'positive_votes':0,
    'negative_votes':0,
    'lobby_type':7,
    'tower_status_radiant':1830,
    'start_time':1451682491,
    'human_players':10,
    'leagueid':0,
    'barracks_status_radiant':63,
    'players':[
        {
            'last_hits':62,
            'item_5':102,
            'hero_damage':13674,
            'gold_spent':15515,
            'gold_per_min':347,
            'denies':0,
            'level':23,
            'item_0':180,
            'item_3':30,
            'gold':4127,
            'deaths':5,
            'assists':24,
            'xp_per_min':492,
            'leaver_status':0,
            'account_id':86716949,
            'ability_upgrades':[
                {
                    'time':397,
                    'level':1,
                    'ability':5023
                }
            ],
            'player_slot':0,
            'hero_healing':108,
            'item_4':116,
            'item_1':1,
            'hero_id':7,
            'tower_damage':67,
            'kills':7,
            'item_2':81
        }    ],
    'match_seq_num':1802988952,
    'cluster':133,
    'match_id':2047382110,
    'duration':3503,
    'radiant_win':True,
    'first_blood_time':145,
    'tower_status_dire':0,
    'barracks_status_dire':0
}
```

The data from the player fields is the most interesting here, together with the duration of the match.