

Decoding Team Composition in MOBA

A Learning Approach

Jiachen Li (jiachenl)
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA

Yilin He (yilinhe)
Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA

Chengliang Lian (clian)
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA

Abstract—Team composition has been a interesting as well as challenging problem especially in the setting of competitive online games. A good team composition will strongly improve your chance for wining, while a bad team composition may probably ruin your game. In this report, we studied the team composition in the massive online battle arena (MOBA) games, and tried to decode the impact of team composition. We collected data from two most popular MOBA games, and then proposed two feature models to represent these data. Machine learning algorithms with different characteristics such as logistic regression and neural networks are used to evaluate our works, and our current results shows that the team composition does have connection with game results and the relationship is encodable.

I. INTRODUCTION

Multiplayer online battle arena (MOBA) is a rising force in online games that has features of fast paced, single round and team play. In the past few months, more than 27 million players fight with each other per day in one of such kind of game called League of Legends (LoL) [1] while millions are attracted by another game called Defense of the Ancient 2 (DotA2). International MOBA tournaments are held over the world with a prize pool of over \$10 million [2]. Analysis on MOBA games will not only gain us experience of data analysis but will also be beneficial to players and game companies all over the world.

In MOBA games like LoL or DotA 2, two groups of five players are pitted against each other based on their level of game proficiency. In the beginning of a game, each player first picks an unique character out of over 100 options, and these 5 characters picked in a group finally forms a team. Given that the game server usually matches the players with similar levels within a game, then the result of a game should not be affected too much by the skill of each individual player, but depends on the characters combination of different teams, which is known as team composition.

An interesting questions to the online communities as well as professional teams is how to form a good team composition to achieve highest winning probability. The challenging part for this problem is the large number of possible team composition. Since each game contains more than 100 characters, there are more than $\binom{100}{10} \cdot \binom{10}{5}$ kinds of team composition in LoL or DotA 2.

A. Goal and Progress

To answer this question, we can translate it into a classic machine learning problem: given a team composition, predict the outcome for the game. Then we can interpret the parameters in our machine learning algorithm to understand how to form a team composition for MOBA games.

In the midpoint, we have successfully fulfilled the goal defined in the proposal. We collected and cleaned up data for both LoL and DotA 2. We built a model to predict game outcome and evaluated it using logistic regression and support vector machine. Using the simple model, we achieved an accuracy of 60.62% for LoL and 63.48% for DotA 2.

II. DATA SET PREPARATION

We have obtained two data sets for this study, one from the LoL and the other from DotA 2. The whole data set contains more than 80,000 independent game history records in total, which are adequate for our analysis in the first few steps.

A. Data Collection

The data set was collected through public APIs provided by the video game producers Riot and Steam respectively. The APIs can provide us with a given number of most recent completed games as well as the corresponding whole statistics. To make full use of the APIs, we ran our feature collection program on AWS Cloud Computing resources to collect these data in a 7×24 manner.

B. Data Contents

The contents of the original data collected from APIs are shown as follow:

- *Game Id*: An Unique identifier for the game, used to avoid data redundancy;
- *Game duration*: how long the each game last, used for data cleaning;
- *Winner*: The winning team, used as labels;
- *Team composition*: the characters selected by each team, used as main features;
- *Skill levels*: the player's performance in his/her past history, used to evaluate the potential performance of this player;

- *KDA Statistics*: the character's performance in this game, i.e., how many other characters it killed (K), how many times it died (D) and how many times it assisted its teammate to kill the other characters (A).

C. Data Cleaning

As we all known that the performance of machine learning algorithms is heavily dependent on the quality of features. So before extracting features from raw data, we need to do some data cleaning.

Since the public APIs always return the most recent game records, we should first to check whether there is any redundancy in our data set. Besides, we need to make sure the data is useful, which can be verified from the game duration. For example, if the game duration is too short, then it is quite possible that some player leaves the game early, so that this record may not be able to support our learning goal and should be dropped. Finally, we will have a look at the KDA statistics, if some character's KDA is below a normal level, then this character may not fully participate into the game. Therefore, the corresponding record should also be dropped.

After finishing these procedures, we can then extract features from the data and train our models.

III. MODELS AND ASSUMPTIONS

In this section, we proposed two models for decoding team composition. The first model is based on ranking the performance of each character while the second model takes the group effect into consideration. We take the first model as the baseline, and mainly apply our ideas in the second model.

To start discussing the models, let us first define the two teams in the game as \mathcal{A} and \mathcal{B} , where each team contains five unique characters (i.e., $c_1, c_2 \dots, c_5$). Moreover, suppose that the game has N characters, then all the characters can be encoded with an integer between 1 and N .

A. Model-I: Character Ranking Model

In the character ranking model, what we really want to learn is the "weight" of different characters, which measures how important the characters to a team. To introduce the that, we may first need to make some assumption:

Assumption 1: We assume that players in both team all have high skills. Moreover, they are proficient with the characters they pick.

Remark 1: This assumption is reasonable because, during the data collection part, we only collected the records with a higher skill levels. Moreover, in order to win the game, players are not likely to pick the characters that they are not familiar with, especially for the high level player.

With this assumption, we can further assume that the game result is independent of the players, then we can use the vector $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$ to represent the training feature. The element x_i indicates the presence of i th character in the game, and the detailed value for it is given by

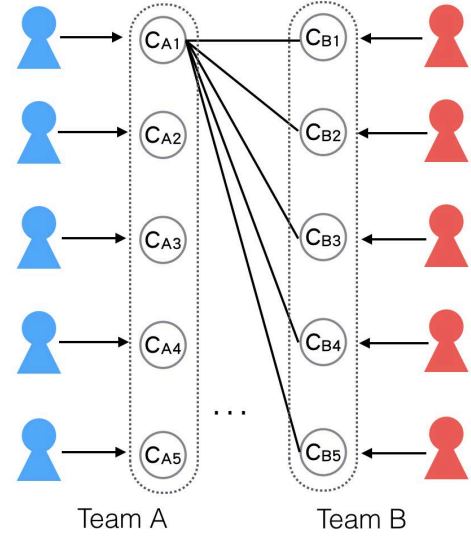


Fig. 1: An illustration of team composition and the aspects that may influences the game result.

$$x_i = \begin{cases} 1 & \text{if } i \in \mathcal{A}, \\ -1 & \text{if } i \in \mathcal{B}, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Of course, the winner team will work as the label of the feature.

B. Model-II: Group Ranking Model

Unlike the character ranking model where we make the independent assumption between player skills and game results. The group ranking model not only considers the player skills, but also take the relationship of different characters into account. For example, Fig. 1 has shown the factors that may have potential influence on the game result. It can be seen from the figure that the performance of one specific character should affect all the other characters in the different team. This is not difficult to understand because there are many times in the game that all the characters in a team will need to work together to achieve their goals. Therefore, to learn the team composition, we need also use such information.

To utilize these additional information, we first introduce the character matrix.

Definition 1: A matrix $A \in \mathbb{R}^{N \times N}$ is called the character matrix if the element $a_{i,j}$ indicates the probability that the team with character i beats the team with character j , i.e. $a_{i,j} = \Pr\{\text{team with } c_i \text{ beats team with } c_j\}$.

The character matrix gives us an evaluation between different character pair that can be used for learning. However, to obtain a more complete feature we may also need to consider the impact of the players (i.e., use a function $g(P_{A,i}, P_{B,j})$ to evaluate the winning ratio between two players), so the final feature could be a matrix $F \in \mathbb{R}^{N \times N}$ based on this information. The element of feature matrix is defined as

$$f_{i,j} = a_{i,j} \cdot g(P_{A,i}, P_{B,j}), \quad (2)$$

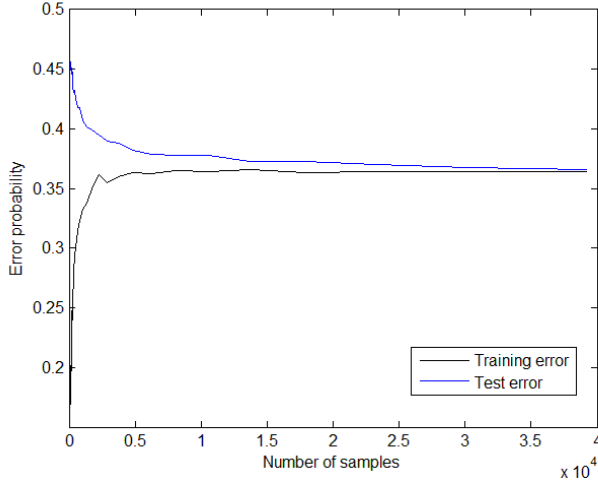


Fig. 2: The error probability of logistic regression on the Dota 2 dataset.

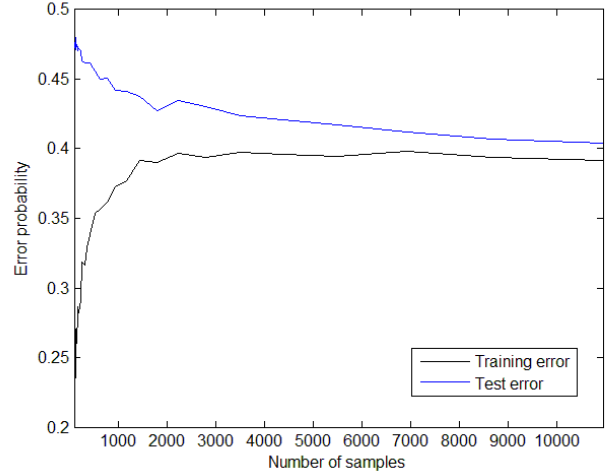


Fig. 3: The error probability of logistic regression on the LoL dataset.

where the $a_{i,j}$ is the element from character matrix, $g(P_{A,i}, P_{B,j})$ is the function that measure the winning ratio between the player in group \mathcal{A} using character i and the player in group \mathcal{B} using character j . Notice that we can multiply these two elements because they are independent.

Remark 2: There is no doubt that the feature matrix contains more information than the plain character feature in the character ranking model. However, one of the biggest challenge when applying the feature matrix is its high dimension (i.e. $N \times N$, which will also lead to the sparsity of the parameters). To solve this problem, we will use the clustering methods to cluster the N characters into K groups, so that the comparison between two individual characters will become the comparison between two clusters. Thus the dimension problem will be reduced. (This will be the one of the work during the second part of this semester.)

IV. PERFORMANCE EVALUATION

A. Evaluation of logistic regression

Fig. 2 and 3 show the error probability versus training sample on two dataset with Logistic Regression, We select a learning rate of $\alpha = 0.001$, the proportion of test set is 30%, which is randomly picked from the dataset. For each data set, the iteration time for gradient decent is 100 during the training.

This figure represents the learning curve for logistic regression algorithm. With small number of training examples, we obtained a small training error and a large testing error. As we increase the number of training examples, the training error increase and testing error decreases. For DotA 2, both test error and training error converges to 36.5% after around 10,000 training points.

The weight vector we obtained in logistic regression represents each character's influence on game result. For character i , a positive weight w_i implies that picking this character will

increase the chance of winning. Therefore, the prediction is based on individual characters and does not consider synergy between characters.

B. Evaluation of neural network

In neural networks model, we choose to use 3 layers: 1 hidden layer with 100 nodes. The input layer has p nodes and the output layer has 2 nodes. We implemented feedforward back propagation neural network. Sigmoid transform was applied and lambda in cost function was set to 0.5. (Since the evaluation of error probability versus the number of sample is too expensive for neural networks, we didn't do that in this midway report, but we will add it in the final report)

C. Comparison of algorithms

TABLE I: performance of different algorithms

Accuracy (%)	LoL		Dota 2	
	Training	Test	Training	Test
Logistic Regression	62.55	60.62	63.06	63.48
Neural Networks	72.63	70.77	65.12	64.81

While we can not directly interpret the parameters we obtained from neural networks model, the high accuracy shows the importance of considering the relationship between selected characters. Since logistic regression has a linear decision boundary, it can only make prediction based on a linear combination of individual characters. Therefore, logistic regression can not capture the relationship between selected characters. On the other side, the learning process of Neural Network combines input feature in the hidden layer to take consideration of the relationship between selected characters.

D. Comparison of data sets

Logistic regression yield similar accuracy for LoL and Dota 2 while neural networks has better result on LoL than Dota 2. This result from logistic regression implies that in LoL and

DotA 2, selection of individual characters have similar impact on game result. One possible explanation for higher accuracy in LoL data set when using neural networks is that the team composition in LoL focus more on the synergy between characters, thus considering more complicated relationship between characters gives better result.

V. CONCLUSION AND FUTURE WORK

Team composition has been a interesting as well as challenging problem especially in the setting of competitive online games. A good team composition will strongly improve your chance for wining, while a bad team composition may probably ruin your game.

In this report, we studied the team composition in the massive online battle arena (MOBA) games, and tried to decode the impact of team composition. During first half of the semester, we finished the goal we set in the proposal including:

- collect data set from LoL and DotA 2
- build feature model
- run logistic regression on each data set
- run neural network on each data set

Our current results shows that the team composition does have connection with game results and the relationship is encodable. However, due to the limitation of original model, we proposed a second model to encode the relationship between selected characters and the players.

For the remaining semester, our plan is shown as follows p Nov 1 - Nov 15: Set up and run logistic regression and deep learning algorithm on our dataset using new model.

Nov 16 - Nov 30: Analysis our result and tune parameters for different algorithm.

To make the project successful, the minimum requirement is not longer the data collection, pre-processing and algorithms implementation, but to validate our model and make a better analysis and learn well.

REFERENCES

- [1] Ian Sherr, "Player Tally for League of Legends Surges," *Digits RSS*. N.p., n.d. Web. 28 Sept, 2014.
- [2] Valve, Dota 2 – 2014 Compendium – The International. Dota 2 Official Website.